

Parametric Techniques

Lecture 3

Jason Corso

SUNY at Buffalo

22 January 2009

Introduction

- In Lecture 2, we learned how to form optimal decision boundaries when the full probabilistic structure of the problem is known.
- However, this is rarely the case in practice.
- Instead, we have some knowledge of the problem and some example data and we must estimate the probabilities.
- **Focus of this lecture** is to study a pair of techniques for estimating the parameters of the likelihood models (given a particular form of likelihood function, such as a Gaussian).

Introduction

- In Lecture 2, we learned how to form optimal decision boundaries when the full probabilistic structure of the problem is known.
- However, this is rarely the case in practice.
- Instead, we have some knowledge of the problem and some example data and we must estimate the probabilities.
- **Focus of this lecture** is to study a pair of techniques for estimating the parameters of the likelihood models (given a particular form of likelihood function, such as a Gaussian).
- **Parametric Models** – For a particular class ω_i , we consider a set of parameters θ_i to fully define the likelihood model.
 - For the Gaussian, $\theta_i = (\mu_i, \Sigma_i)$.

Introduction

- In Lecture 2, we learned how to form optimal decision boundaries when the full probabilistic structure of the problem is known.
- However, this is rarely the case in practice.
- Instead, we have some knowledge of the problem and some example data and we must estimate the probabilities.
- **Focus of this lecture** is to study a pair of techniques for estimating the parameters of the likelihood models (given a particular form of likelihood function, such as a Gaussian).
- **Parametric Models** – For a particular class ω_i , we consider a set of parameters θ_i to fully define the likelihood model.
 - For the Gaussian, $\theta_i = (\mu_i, \Sigma_i)$.
- **Supervised Learning** – we are working in a supervised situation where we have an set of training data:

$$\mathcal{D} = \{(\mathbf{x}, \omega)_1, (\mathbf{x}, \omega)_2, \dots, (\mathbf{x}, \omega)_N\} \quad (1)$$

Overview of the Methods

- **Intuitive Problem:** Given a set of training data, \mathcal{D} , containing labels for c classes, train the likelihood models $p(\mathbf{x}|\omega_i, \theta_i)$ by estimating the parameters θ_i for $i = 1, \dots, c$.

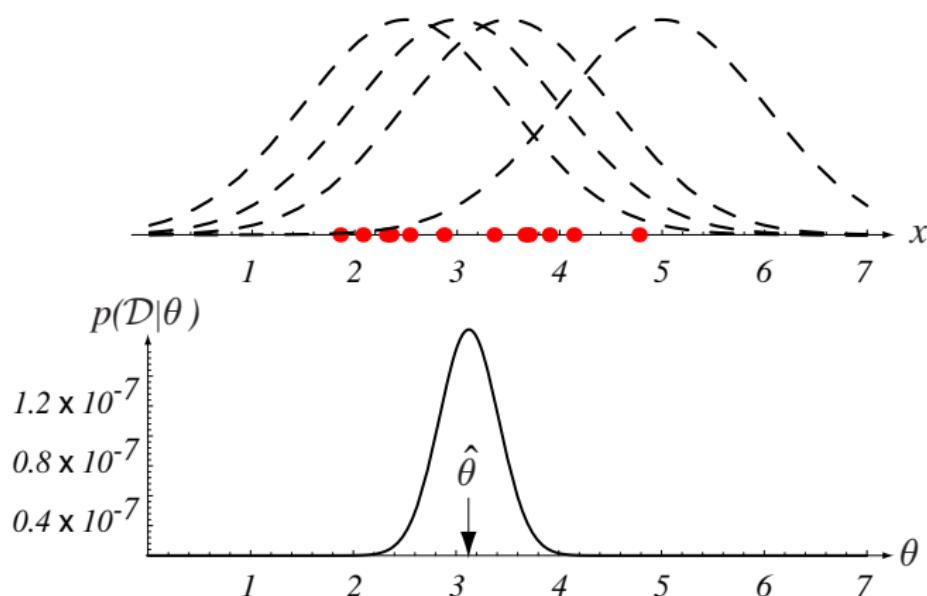
Overview of the Methods

- **Intuitive Problem:** Given a set of training data, \mathcal{D} , containing labels for c classes, train the likelihood models $p(\mathbf{x}|\omega_i, \theta_i)$ by estimating the parameters θ_i for $i = 1, \dots, c$.
- **Maximum Likelihood Parameter Estimation**
 - Views the parameters as quantities that are fixed by unknown.
 - The best estimate of their value is the one that maximizes the probability of obtaining the samples in \mathcal{D} .

Overview of the Methods

- **Intuitive Problem:** Given a set of training data, \mathcal{D} , containing labels for c classes, train the likelihood models $p(\mathbf{x}|\omega_i, \theta_i)$ by estimating the parameters θ_i for $i = 1, \dots, c$.
- **Maximum Likelihood Parameter Estimation**
 - Views the parameters as quantities that are fixed by unknown.
 - The best estimate of their value is the one that maximizes the probability of obtaining the samples in \mathcal{D} .
- **Bayesian Parameter Estimation**
 - Views the parameters as random variables having some known prior distribution.
 - The samples converts this prior into a posterior and revises our estimate of the distribution over the parameters.
 - We shall typically see that the posterior is increasingly peaked for larger \mathcal{D} .

Maximum Likelihood Intuition



- Underlying model is assumed to be a Gaussian of particular variance but unknown mean.

Preliminaries

- Separate our training data according to class; i.e., we have c data sets $\mathcal{D}_1, \dots, \mathcal{D}_c$.
- Assume that samples in \mathcal{D}_i give no information for θ_j for all $i \neq j$.

Preliminaries

- Separate our training data according to class; i.e., we have c data sets $\mathcal{D}_1, \dots, \mathcal{D}_c$.
- Assume that samples in \mathcal{D}_i give no information for θ_j for all $i \neq j$.
- Assume the samples in \mathcal{D}_j have been drawn independently according to the (unknown but) fixed density $p(\mathbf{x}|\omega_j)$.
 - We say these samples are **i.i.d.** — independent and identically distributed.

Preliminaries

- Separate our training data according to class; i.e., we have c data sets $\mathcal{D}_1, \dots, \mathcal{D}_c$.
- Assume that samples in \mathcal{D}_i give no information for θ_j for all $i \neq j$.
- Assume the samples in \mathcal{D}_j have been drawn independently according to the (unknown but) fixed density $p(\mathbf{x}|\omega_j)$.
 - We say these samples are **i.i.d.** — independent and identically distributed.
- Assume $p(\mathbf{x}|\omega_j)$ has some fixed parametric form and is fully described by θ_j ; hence we write $p(\mathbf{x}|\omega_j, \theta_j)$.

Preliminaries

- Separate our training data according to class; i.e., we have c data sets $\mathcal{D}_1, \dots, \mathcal{D}_c$.
- Assume that samples in \mathcal{D}_i give no information for θ_j for all $i \neq j$.
- Assume the samples in \mathcal{D}_j have been drawn independently according to the (unknown but) fixed density $p(\mathbf{x}|\omega_j)$.
 - We say these samples are **i.i.d.** — independent and identically distributed.
- Assume $p(\mathbf{x}|\omega_j)$ has some fixed parametric form and is fully described by θ_j ; hence we write $p(\mathbf{x}|\omega_j, \theta_j)$.
- We thus have c separate problems of the form:

Definition

Use a set $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of training samples drawn independently from the density $p(\mathbf{x}|\theta)$ to estimate the unknown parameter vector θ .

(Log-)Likelihood

- Because we assume i.i.d. we have

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^n p(\mathbf{x}_k|\boldsymbol{\theta}) . \quad (2)$$

- The log-likelihood is typically easier to work with both analytically and numerically.

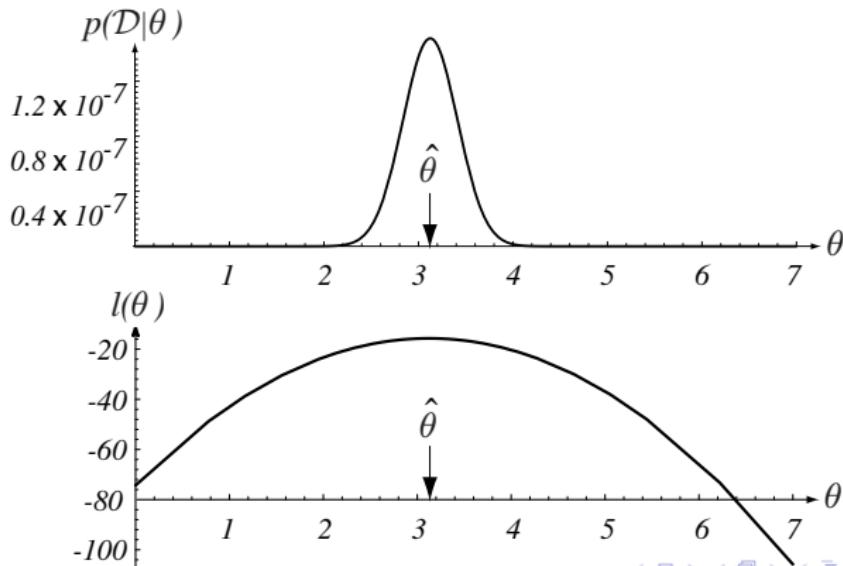
$$l_{\mathcal{D}}(\boldsymbol{\theta}) \equiv l(\boldsymbol{\theta}) \doteq \ln p(\mathcal{D}|\boldsymbol{\theta}) \quad (3)$$

$$= \sum_{k=1}^n \ln p(\mathbf{x}_k|\boldsymbol{\theta}) \quad (4)$$

Maximum (Log-)Likelihood

- The **maximum likelihood estimate** of θ if the value $\hat{\theta}$ that maximizes $p(\mathcal{D}|\theta)$ or equivalently maximizes $l_{\mathcal{D}}(\theta)$.

$$\hat{\theta} = \arg \max_{\theta} l_{\mathcal{D}}(\theta) \quad (5)$$



Necessary Conditions for MLE

- For p parameters, $\boldsymbol{\theta} \doteq [\theta_1 \quad \theta_2 \quad \dots \quad \theta_p]^T$.
- Let $\nabla_{\boldsymbol{\theta}}$ be the gradient operator, then $\nabla_{\boldsymbol{\theta}} \doteq \left[\frac{\partial}{\partial \theta_1} \quad \dots \quad \frac{\partial}{\partial \theta_p} \right]^T$.
- The set of **necessary conditions** for the maximum likelihood estimate of $\boldsymbol{\theta}$ are obtained from the following system of p equations:

$$\nabla_{\boldsymbol{\theta}} l = \sum_{k=1}^n \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{x}_k | \boldsymbol{\theta}) = 0 \quad (6)$$

Necessary Conditions for MLE

- For p parameters, $\boldsymbol{\theta} \doteq [\theta_1 \quad \theta_2 \quad \dots \quad \theta_p]^T$.
- Let $\nabla_{\boldsymbol{\theta}}$ be the gradient operator, then $\nabla_{\boldsymbol{\theta}} \doteq \left[\frac{\partial}{\partial \theta_1} \quad \dots \quad \frac{\partial}{\partial \theta_p} \right]^T$.
- The set of **necessary conditions** for the maximum likelihood estimate of $\boldsymbol{\theta}$ are obtained from the following system of p equations:

$$\nabla_{\boldsymbol{\theta}} l = \sum_{k=1}^n \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{x}_k | \boldsymbol{\theta}) = 0 \quad (6)$$

- A solution $\hat{\boldsymbol{\theta}}$ to (6) can be a true global maximum, a local maximum or minimum or an inflection point of $l(\boldsymbol{\theta})$.

Necessary Conditions for MLE

- For p parameters, $\boldsymbol{\theta} \doteq [\theta_1 \quad \theta_2 \quad \dots \quad \theta_p]^T$.
- Let $\nabla_{\boldsymbol{\theta}}$ be the gradient operator, then $\nabla_{\boldsymbol{\theta}} \doteq \left[\frac{\partial}{\partial \theta_1} \quad \dots \quad \frac{\partial}{\partial \theta_p} \right]^T$.
- The set of **necessary conditions** for the maximum likelihood estimate of $\boldsymbol{\theta}$ are obtained from the following system of p equations:

$$\nabla_{\boldsymbol{\theta}} l = \sum_{k=1}^n \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{x}_k | \boldsymbol{\theta}) = 0 \quad (6)$$

- A solution $\hat{\boldsymbol{\theta}}$ to (6) can be a true global maximum, a local maximum or minimum or an inflection point of $l(\boldsymbol{\theta})$.
- Keep in mind that $\hat{\boldsymbol{\theta}}$ is only an estimate. Only in the limit of an infinitely large number of training samples can we expect it to be the true parameters of the underlying density.

Gaussian Case with Known Σ and Unknown μ

- For a single sample point \mathbf{x}_k :

$$\ln p(\mathbf{x}_k|\boldsymbol{\mu}) = -\frac{1}{2} \ln \left[(2\pi)^d |\boldsymbol{\Sigma}| \right] - \frac{1}{2} (\mathbf{x}_k - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) \quad (7)$$

$$\nabla_{\boldsymbol{\mu}} \ln p(\mathbf{x}_k|\boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) \quad (8)$$

Gaussian Case with Known Σ and Unknown μ

- For a single sample point \mathbf{x}_k :

$$\ln p(\mathbf{x}_k | \boldsymbol{\mu}) = -\frac{1}{2} \ln \left[(2\pi)^d |\boldsymbol{\Sigma}| \right] - \frac{1}{2} (\mathbf{x}_k - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) \quad (7)$$

$$\nabla_{\boldsymbol{\mu}} \ln p(\mathbf{x}_k | \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) \quad (8)$$

- We see that the ML-estimate must satisfy

$$\sum_{k=1}^n \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0 \quad (9)$$

Gaussian Case with Known Σ and Unknown μ

- For a single sample point \mathbf{x}_k :

$$\ln p(\mathbf{x}_k | \boldsymbol{\mu}) = -\frac{1}{2} \ln \left[(2\pi)^d |\boldsymbol{\Sigma}| \right] - \frac{1}{2} (\mathbf{x}_k - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) \quad (7)$$

$$\nabla_{\boldsymbol{\mu}} \ln p(\mathbf{x}_k | \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) \quad (8)$$

- We see that the ML-estimate must satisfy

$$\sum_{k=1}^n \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0 \quad (9)$$

- And we get the sample mean!

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k \quad (10)$$

Univariate Gaussian Case with Unknown μ and σ^2

The Log-Likelihood

- Let $\theta = (\mu, \sigma^2)$. The log-likelihood of x_k is

$$\ln p(x_k|\theta) = -\frac{1}{2} \ln [2\pi\sigma^2] - \frac{1}{2\sigma^2}(x_k - \mu)^2 \quad (11)$$

$$\nabla_{\theta} \ln p(x_k|\theta) = \left[\begin{array}{c} \frac{1}{\sigma^2}(x_k - \mu) \\ -\frac{1}{2\sigma^2} + \frac{(x_k - \mu)^2}{2\sigma^4} \end{array} \right] \quad (12)$$

Univariate Gaussian Case with Unknown μ and σ^2

Necessary Conditions

- The following conditions are defined:

$$\sum_{k=1}^n \frac{1}{\hat{\sigma}^2} (x_k - \hat{\mu}) = 0 \quad (13)$$

$$-\sum_{k=1}^n \frac{1}{\hat{\sigma}^2} + \sum_{k=1}^n \frac{(x_k - \hat{\mu})^2}{\hat{\sigma}^4} = 0 \quad (14)$$

Univariate Gaussian Case with Unknown μ and σ^2

ML-Estimates

- After some manipulation we have the following:

$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^n x_k \quad (15)$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{k=1}^n (x_k - \hat{\mu})^2 \quad (16)$$

- These are encouraging results – even in the case of unknown μ and σ^2 the ML-estimate of μ corresponds to the sample mean.

Bias

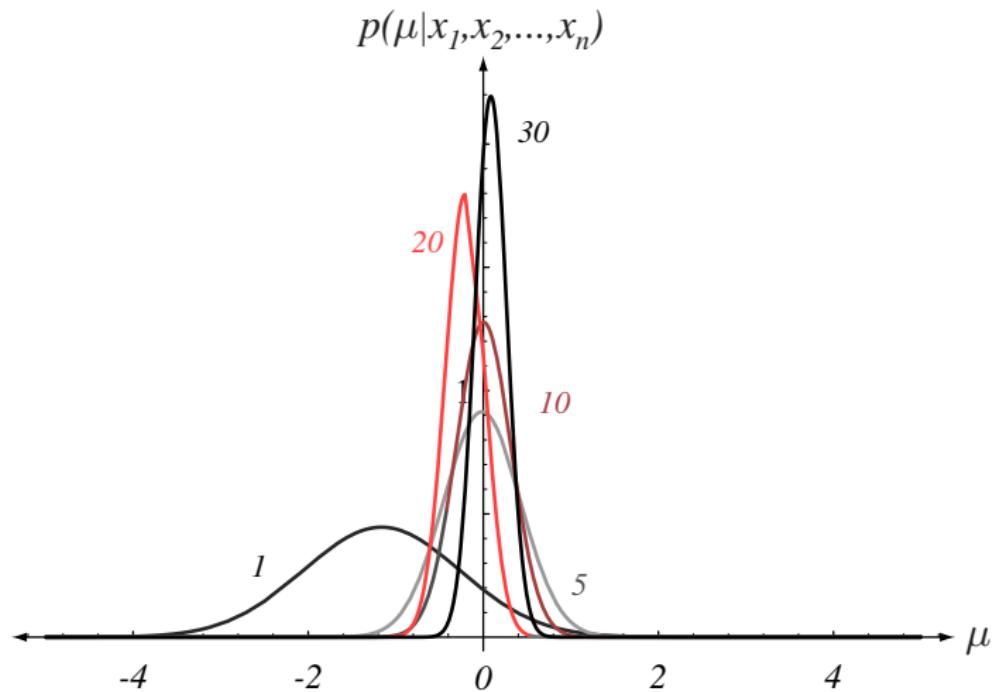
- The maximum likelihood estimate for the variance σ^2 is **biased**.
- The expected value over datasets of size n of the sample variance is not equal to the true variance

$$\mathcal{E} \left[\frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2 \right] = \frac{n-1}{n} \sigma^2 \neq \sigma^2 \quad (17)$$

- In other words, the ML-estimate of the variance systematically underestimates the variance of the distribution.
- As $n \rightarrow \infty$ the problem of bias is reduced or removed, but bias remains a problem of the ML-estimator.
- An unbiased ML-estimator of the variance is

$$\hat{\sigma}_{\text{unbiased}}^2 = \frac{1}{n-1} \sum_{k=1}^n (x_k - \hat{\mu})^2 \quad (18)$$

Bayesian Parameter Estimation Intuition



General Assumptions

Bayesian Parameter Estimation

- The form of the density $p(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be known (e.g., it is a Gaussian).

General Assumptions

Bayesian Parameter Estimation

- The form of the density $p(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be known (e.g., it is a Gaussian).
- The values of the parameter vector $\boldsymbol{\theta}$ are not exactly known.

General Assumptions

Bayesian Parameter Estimation

- The form of the density $p(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be known (e.g., it is a Gaussian).
- The values of the parameter vector $\boldsymbol{\theta}$ are not exactly known.
- Our initial knowledge about the parameters is summarized in a prior distribution $p(\boldsymbol{\theta})$.

General Assumptions

Bayesian Parameter Estimation

- The form of the density $p(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be known (e.g., it is a Gaussian).
- The values of the parameter vector $\boldsymbol{\theta}$ are not exactly known.
- Our initial knowledge about the parameters is summarized in a prior distribution $p(\boldsymbol{\theta})$.
- The rest of our knowledge about $\boldsymbol{\theta}$ is contained in a set \mathcal{D} of n i.i.d. samples $\mathbf{x}_1, \dots, \mathbf{x}_n$ drawn according to fixed $p(\mathbf{x})$.

General Assumptions

Bayesian Parameter Estimation

- The form of the density $p(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be known (e.g., it is a Gaussian).
- The values of the parameter vector $\boldsymbol{\theta}$ are not exactly known.
- Our initial knowledge about the parameters is summarized in a prior distribution $p(\boldsymbol{\theta})$.
- The rest of our knowledge about $\boldsymbol{\theta}$ is contained in a set \mathcal{D} of n i.i.d. samples $\mathbf{x}_1, \dots, \mathbf{x}_n$ drawn according to fixed $p(\mathbf{x})$.

Goal

Our ultimate goal is to estimate $p(\mathbf{x}|\mathcal{D})$, which is as close as we can come to estimating the unknown $p(\mathbf{x})$.

Linking Likelihood and the Parameter Distribution

- How do we relate the prior distribution on the parameters to the samples?

Linking Likelihood and the Parameter Distribution

- How do we relate the prior distribution on the parameters to the samples?
- **Missing Data!** The samples will convert our prior $p(\theta)$ to a posterior $p(\theta|\mathcal{D})$, by integrating the joint density over θ :

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}, \boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \quad (19)$$

$$= \int p(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \quad (20)$$

Linking Likelihood and the Parameter Distribution

- How do we relate the prior distribution on the parameters to the samples?
- **Missing Data!** The samples will convert our prior $p(\theta)$ to a posterior $p(\theta|\mathcal{D})$, by integrating the joint density over θ :

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}, \boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \quad (19)$$

$$= \int p(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \quad (20)$$

- And, because the distribution of \mathbf{x} is known given the parameters $\boldsymbol{\theta}$, we simplify to

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \quad (21)$$

Linking Likelihood and the Parameter Distribution

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

- We can see the link between the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ and the posterior for the unknown parameters $p(\boldsymbol{\theta}|\mathcal{D})$.

Linking Likelihood and the Parameter Distribution

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

- We can see the link between the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ and the posterior for the unknown parameters $p(\boldsymbol{\theta}|\mathcal{D})$.
- If the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ peaks very sharply for sample point $\hat{\boldsymbol{\theta}}$, then we obtain

$$p(\mathbf{x}|\mathcal{D}) \simeq p(\mathbf{x}|\hat{\boldsymbol{\theta}}) . \quad (22)$$

Linking Likelihood and the Parameter Distribution

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

- We can see the link between the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ and the posterior for the unknown parameters $p(\boldsymbol{\theta}|\mathcal{D})$.
- If the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ peaks very sharply for sample point $\hat{\boldsymbol{\theta}}$, then we obtain

$$p(\mathbf{x}|\mathcal{D}) \simeq p(\mathbf{x}|\hat{\boldsymbol{\theta}}) . \quad (22)$$

- And, we will see that during Bayesian parameter estimation, the distribution over the parameters will get increasingly “peaky” as the number of samples increases.

Linking Likelihood and the Parameter Distribution

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

- We can see the link between the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ and the posterior for the unknown parameters $p(\boldsymbol{\theta}|\mathcal{D})$.
- If the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ peaks very sharply for sample point $\hat{\boldsymbol{\theta}}$, then we obtain

$$p(\mathbf{x}|\mathcal{D}) \simeq p(\mathbf{x}|\hat{\boldsymbol{\theta}}) . \quad (22)$$

- And, we will see that during Bayesian parameter estimation, the distribution over the parameters will get increasingly “peaky” as the number of samples increases.
- What if the integral is not readily analytically computed?

The Posterior Density on the Parameters

- The primary task in Bayesian Parameter Estimation is the computation of the posterior density $p(\theta|\mathcal{D})$.
- By Bayes formula

$$p(\theta|\mathcal{D}) = \frac{1}{Z} p(\mathcal{D}|\theta) p(\theta) \quad (23)$$

- Z is a normalizing constant:

$$Z = \int p(\mathcal{D}|\theta) p(\theta) d\theta \quad (24)$$

The Posterior Density on the Parameters

- The primary task in Bayesian Parameter Estimation is the computation of the posterior density $p(\theta|\mathcal{D})$.
- By Bayes formula

$$p(\theta|\mathcal{D}) = \frac{1}{Z} p(\mathcal{D}|\theta) p(\theta) \quad (23)$$

- Z is a normalizing constant:

$$Z = \int p(\mathcal{D}|\theta) p(\theta) d\theta \quad (24)$$

- And, by the independence assumption on \mathcal{D} :

$$p(\mathcal{D}|\theta) = \prod_{k=1}^n p(\mathbf{x}_k|\theta) \quad (25)$$

- Let's see some examples before we return to this general formulation.

Univariate Gaussian Case with Known σ^2

- Assume $p(x|\mu) \sim N(\mu, \sigma^2)$ with known σ^2 .
- Whatever prior knowledge we know about μ is expressed in $p(\mu)$, which is known.

Univariate Gaussian Case with Known σ^2

- Assume $p(x|\mu) \sim N(\mu, \sigma^2)$ with known σ^2 .
- Whatever prior knowledge we know about μ is expressed in $p(\mu)$, which is known.
- Indeed, we assume it took is a Gaussian

$$p(\mu) \sim N(\mu_0, \sigma_0^2) . \quad (26)$$

μ_0 represents our best guess of the value of the mean and σ_0^2 represents our uncertainty about this guess.

Univariate Gaussian Case with Known σ^2

- Assume $p(x|\mu) \sim N(\mu, \sigma^2)$ with known σ^2 .
- Whatever prior knowledge we know about μ is expressed in $p(\mu)$, which is known.
- Indeed, we assume it took is a Gaussian

$$p(\mu) \sim N(\mu_0, \sigma_0^2) . \quad (26)$$

μ_0 represents our best guess of the value of the mean and σ_0^2 represents our uncertainty about this guess.

- Note: the choice of the prior as a Gaussian is not so crucial. Rather, the assumption that **we know the prior is crucial**.

Univariate Gaussian Case with Known σ^2

Training samples

- We assume that we are given samples $\mathcal{D} = \{x_1, \dots, x_n\}$ from $p(x, \mu)$.
- Take some time to think through this point—unlike in MLE, we cannot assume that we have a single value of the parameter in the underlying distribution.

Univariate Gaussian Case with Known σ^2

Bayes Rule



$$p(\mu|\mathcal{D}) = \frac{1}{Z} p(\mathcal{D}|\mu)p(\mu) \quad (27)$$

$$= \frac{1}{Z} \prod_k p(x_k|\mu)p(\mu) \quad (28)$$

- See how the training samples modulate our prior knowledge of the parameters in the posterior?

Univariate Gaussian Case with Known σ^2

Expanding...



$$p(\mu|\mathcal{D}) = \frac{1}{Z} \prod_k \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{1}{2} \left(\frac{x_k - \mu}{\sigma} \right)^2 \right] \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp \left[-\frac{1}{2} \left(\frac{\mu - \mu_0}{\sigma_0} \right)^2 \right] \quad (29)$$

- After some manipulation, we can see that $p(\mu|\mathcal{D})$ is an exponential function of a quadratic of μ , which is another way of saying a normal density.

$$p(\mu|\mathcal{D}) = \frac{1}{Z'} \exp \left[-\frac{1}{2} \left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right) \mu^2 - 2 \left(\frac{1}{\sigma^2} \sum_k x_k + \frac{\mu_0}{\sigma_0^2} \right) \mu \right] \right] \quad (30)$$

Univariate Gaussian Case with Known σ^2

Names of these convenient distributions...

- And, this will be true regardless of the number of training samples.
- In other words, $p(\mu|\mathcal{D})$ remains a normal as the number of samples increases.
- Hence, $p(\mu|\mathcal{D})$ is said to be a **reproducing density**.
- $p(\mu)$ is said to be a **conjugate prior**.

Univariate Gaussian Case with Known σ^2

Rewriting...

- We can write $p(\mu|\mathcal{D}) \sim N(\mu_n, \sigma_n^2)$. Then, we have

$$p(\mu|\mathcal{D}) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{1}{2} \left(\frac{\mu - \mu_n}{\sigma_n}\right)^2\right] \quad (31)$$

- The new coefficients are

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \quad (32)$$

$$\frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2} \bar{\mu}_n + \frac{\mu_0}{\sigma_0^2} \quad (33)$$

- $\bar{\mu}_n$ is the sample mean over the n samples.

Univariate Gaussian Case with Known σ^2

Rewriting...

- Solving explicitly for μ_n and σ_n^2

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \bar{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0 \quad (34)$$

$$\sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2} \quad (35)$$

shows explicitly how the prior information is combined with the training samples **to estimate the parameters of the posterior distribution.**

- After n samples, μ_n is our best guess for the mean of the posterior and σ_n^2 is our uncertainty about it.

Univariate Gaussian Case with Known σ^2

Uncertainty...

- What can we say about this uncertainty as n increases?

$$\sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

Univariate Gaussian Case with Known σ^2

Uncertainty...

- What can we say about this uncertainty as n increases?

$$\sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

- That each observation **monotonically decreases our uncertainty** about the distribution.

$$\lim_{n \rightarrow \infty} \sigma_n^2 = 0 \tag{36}$$

- In other terms, as n increases, $p(\mu|\mathcal{D})$ becomes more and more sharply peaked approaching a Dirac delta function.

Univariate Gaussian Case with Known σ^2

- What can we say about the parameter μ_n as n increases?

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \bar{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

Univariate Gaussian Case with Known σ^2

- What can we say about the parameter μ_n as n increases?

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \bar{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

- It is a convex combination between the sample mean $\bar{\mu}_n$ (from the observed data) and the prior μ_0 .
- Thus, it always lives somewhere between $\bar{\mu}_n$ and μ_0 .
- And, it approaches the sample mean as n approaches ∞ :

$$\lim_{n \rightarrow \infty} \mu_n = \bar{\mu}_n \equiv \frac{1}{n} \sum_{k=1}^n x_k \quad (37)$$

Univariate Gaussian Case with Known σ^2

Putting it all together to obtain $p(x|\mathcal{D})$.

- Our goal has been to obtain an estimate of how likely a novel sample x is given the entire training set \mathcal{D} : $p(x|\mathcal{D})$.

$$p(x|\mathcal{D}) = \int p(x|\mu)p(\mu|\mathcal{D})d\mu \quad (38)$$

$$= \int \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right] \quad (39)$$

$$\int \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp \left[-\frac{1}{2} \left(\frac{\mu-\mu_n}{\sigma_n} \right)^2 \right] \quad (39)$$

$$= \frac{1}{2\pi\sigma\sigma_n} \exp \left[\frac{1}{2} \frac{(x-\mu_n)^2}{\sigma^2 + \sigma_n^2} \right] f(\sigma, \sigma_n) \quad (40)$$

- Essentially, $p(x|\mathcal{D}) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$.

Some Comparisons

Maximum Likelihood

- Point Estimator

$$p(x|\mathcal{D}) = p(x|\hat{\theta})$$

- Parameter Estimate

$$\hat{\theta} = \arg \max_{\theta} \ln p(\mathcal{D}|\theta)$$

Some Comparisons

Maximum Likelihood

- Point Estimator

$$p(x|\mathcal{D}) = p(x|\hat{\theta})$$

- Parameter Estimate

$$\hat{\theta} = \arg \max_{\theta} \ln p(\mathcal{D}|\theta)$$

Bayesian

- Distribution Estimator

$$p(x|\mathcal{D}) = \int p(x|\theta)p(\theta|\mathcal{D})d\theta$$

- Distribution Estimate

$$p(\theta|\mathcal{D}) = \frac{1}{Z}p(\mathcal{D}|\theta)p(\theta)$$

Some Comparisons

- So, is the Bayesian approach like Maximum Likelihood with a prior?

Some Comparisons

- So, is the Bayesian approach like Maximum Likelihood with a prior?
- **NO!**

Maximum Posterior

- Point Estimator

$$p(x|\mathcal{D}) = p(x|\hat{\theta})$$

- Parameter Estimate

$$\hat{\theta} = \arg \max_{\theta} \ln p(\mathcal{D}|\theta)p(\theta)$$

Bayesian

- Distribution Estimator

$$p(x|\mathcal{D}) = \int p(x|\theta)p(\theta|\mathcal{D})d\theta$$

- Distribution Estimate

$$p(\theta|\mathcal{D}) = \frac{1}{Z}p(\mathcal{D}|\theta)p(\theta)$$

Some Comparisons

Comments on the two methods

- For reasonable priors, MLE and BPE are equivalent in the asymptotic limit of infinite training data.
- **Computationally** – MLE methods are preferred for computational reasons because they are comparatively simpler (differential calculus versus multidimensional integration).

Some Comparisons

Comments on the two methods

- For reasonable priors, MLE and BPE are equivalent in the asymptotic limit of infinite training data.
- **Computationally** – MLE methods are preferred for computational reasons because they are comparatively simpler (differential calculus versus multidimensional integration).
- **Interpretability** – MLE methods are often more readily interpreted because they give a single point answer whereas BPE methods give a distribution over answers which can be more complicated.

Some Comparisons

Comments on the two methods

- For reasonable priors, MLE and BPE are equivalent in the asymptotic limit of infinite training data.
- **Computationally** – MLE methods are preferred for computational reasons because they are comparatively simpler (differential calculus versus multidimensional integration).
- **Interpretability** – MLE methods are often more readily interpreted because they give a single point answer whereas BPE methods give a distribution over answers which can be more complicated.
- **Confidence In Priors** – But, the Bayesian methods bring more information to the table. If the underlying distribution is of a different parametric form than originally assumed, Bayesian methods will do better.

Some Comparisons

Comments on the two methods

- For reasonable priors, MLE and BPE are equivalent in the asymptotic limit of infinite training data.
- **Computationally** – MLE methods are preferred for computational reasons because they are comparatively simpler (differential calculus versus multidimensional integration).
- **Interpretability** – MLE methods are often more readily interpreted because they give a single point answer whereas BPE methods give a distribution over answers which can be more complicated.
- **Confidence In Priors** – But, the Bayesian methods bring more information to the table. If the underlying distribution is of a different parametric form than originally assumed, Bayesian methods will do better.
- **Bias-Variance** – Bayesian methods make the bias-variance tradeoff more explicit by directly incorporating the uncertainty in the estimates.

Some Comparisons

Comments on the two methods

Take Home Message

There are strong theoretical and methodological arguments supporting Bayesian estimation, though in practice maximum-likelihood estimation is simpler, and when used for designing classifiers, can lead to classifiers that are nearly as accurate.

Recursive Bayesian Estimation

- Another reason to prefer Bayesian estimation is that it provides a natural way to incorporate additional training data as it becomes available.
- Let a training set with n samples be denoted \mathcal{D}^n .

Recursive Bayesian Estimation

- Another reason to prefer Bayesian estimation is that it provides a natural way to incorporate additional training data as it becomes available.
- Let a training set with n samples be denoted \mathcal{D}^n .
- Then, due to our independence assumption:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^n p(\mathbf{x}_k|\boldsymbol{\theta}) \quad (41)$$

we have

$$p(\mathcal{D}^n|\boldsymbol{\theta}) = p(\mathbf{x}_n|\boldsymbol{\theta})p(\mathcal{D}^{n-1}|\boldsymbol{\theta}) \quad (42)$$

Recursive Bayesian Estimation

- And, with Bayes Formula, we see that the posterior satisfies the recursion

$$p(\boldsymbol{\theta}|\mathcal{D}^n) = \frac{1}{Z} p(\mathbf{x}_n|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}^{n-1}) . \quad (43)$$

- This is an instance of **on-line learning**.

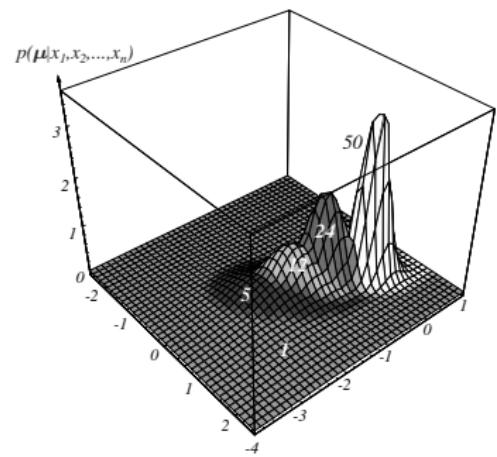
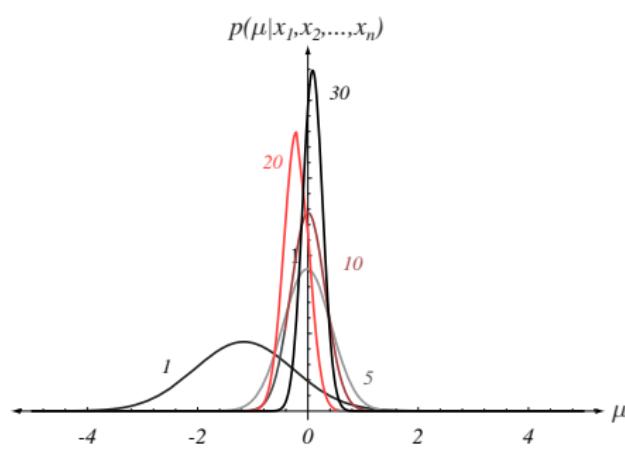
Recursive Bayesian Estimation

- And, with Bayes Formula, we see that the posterior satisfies the recursion

$$p(\boldsymbol{\theta}|\mathcal{D}^n) = \frac{1}{Z} p(\mathbf{x}_n|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}^{n-1}) . \quad (43)$$

- This is an instance of **on-line learning**.
- In principle, this derivation requires that we retain the entire training set in \mathcal{D}^{n-1} to calculate $p(\boldsymbol{\theta}|\mathcal{D}^n)$. But, for some distributions, we can simply retain the sufficient statistics, which contain all the information needed.

Recursive Bayesian Estimation



Example of Recursive Bayes

- Suppose we believe our samples come from a uniform distribution:

$$p(x|\theta) \sim U(0, \theta) = \begin{cases} 1/\theta & 0 \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

- Initially, we know only that our parameter θ is bounded by 10, i.e., $0 \leq \theta \leq 10$.

Example of Recursive Bayes

- Suppose we believe our samples come from a uniform distribution:

$$p(x|\theta) \sim U(0, \theta) = \begin{cases} 1/\theta & 0 \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

- Initially, we know only that our parameter θ is bounded by 10, i.e., $0 \leq \theta \leq 10$.
- Before any data arrive, we have

$$p(\theta|\mathcal{D}^0) = p(\theta) = U(0, 10) . \quad (45)$$

Example of Recursive Bayes

- Suppose we believe our samples come from a uniform distribution:

$$p(x|\theta) \sim U(0, \theta) = \begin{cases} 1/\theta & 0 \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

- Initially, we know only that our parameter θ is bounded by 10, i.e., $0 \leq \theta \leq 10$.
- Before any data arrive, we have

$$p(\theta|\mathcal{D}^0) = p(\theta) = U(0, 10) . \quad (45)$$

- We get a training data set $\mathcal{D} = \{4, 7, 2, 8\}$.

Example of Recursive Bayes

- When the first data point arrives, $x_1 = 4$, we get an improved estimate of θ :

$$p(\theta|\mathcal{D}^1) \propto p(x|\theta)p(\theta|\mathcal{D}^0) = \begin{cases} 1/\theta & 4 \leq \theta \leq 10 \\ 0 & \text{otherwise} \end{cases} \quad (46)$$

Example of Recursive Bayes

- When the first data point arrives, $x_1 = 4$, we get an improved estimate of θ :

$$p(\theta|\mathcal{D}^1) \propto p(x|\theta)p(\theta|\mathcal{D}^0) = \begin{cases} 1/\theta & 4 \leq \theta \leq 10 \\ 0 & \text{otherwise} \end{cases} \quad (46)$$

- When the next data point arrives, $x_2 = 7$, we have

$$p(\theta|\mathcal{D}^2) \propto p(x|\theta)p(\theta|\mathcal{D}^1) = \begin{cases} 1/\theta^2 & 7 \leq \theta \leq 10 \\ 0 & \text{otherwise} \end{cases} \quad (47)$$

Example of Recursive Bayes

- When the first data point arrives, $x_1 = 4$, we get an improved estimate of θ :

$$p(\theta|\mathcal{D}^1) \propto p(x|\theta)p(\theta|\mathcal{D}^0) = \begin{cases} 1/\theta & 4 \leq \theta \leq 10 \\ 0 & \text{otherwise} \end{cases} \quad (46)$$

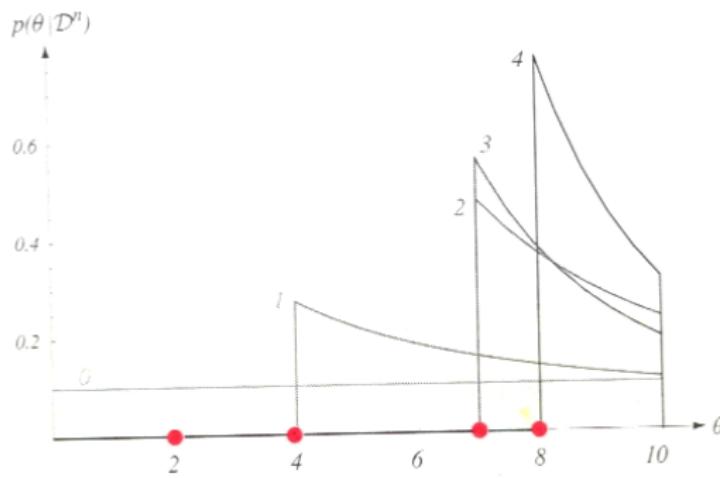
- When the next data point arrives, $x_2 = 7$, we have

$$p(\theta|\mathcal{D}^2) \propto p(x|\theta)p(\theta|\mathcal{D}^1) = \begin{cases} 1/\theta^2 & 7 \leq \theta \leq 10 \\ 0 & \text{otherwise} \end{cases} \quad (47)$$

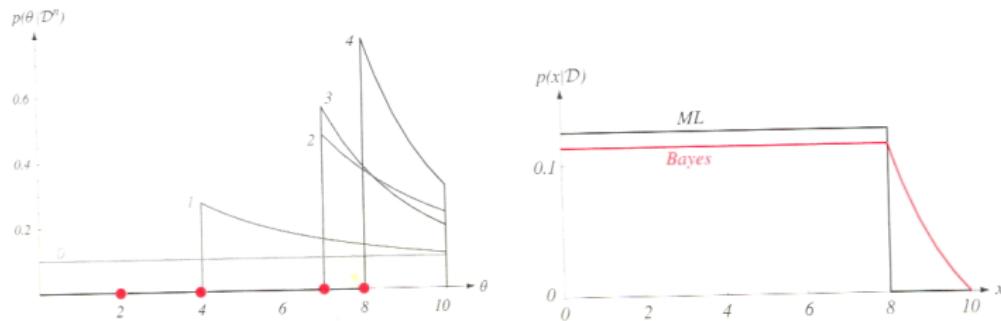
- And so on....

Example of Recursive Bayes

- Notice that each successive data sample introduces a factor of $1/\theta$ into $p(x|\theta)$.
- The distribution of samples is nonzero only for x values above the max, $p(\theta|\mathcal{D}^n) \propto 1/\theta^n$ for $\max_x[\mathcal{D}^n] \leq \theta \leq 10$.
- Our distribution is



Example of Recursive Bayes



- The maximum likelihood solution is $\hat{\theta} = 8$, implying $p(x|\mathcal{D}) \sim U(0, 8)$.
- But, the Bayesian solution shows a different character:
 - Starts out flat.
 - As more points are added, it becomes increasingly peaked at the value of the highest data point.
 - And, the Bayesian estimate has a tail for points above 8 reflecting our prior distribution.

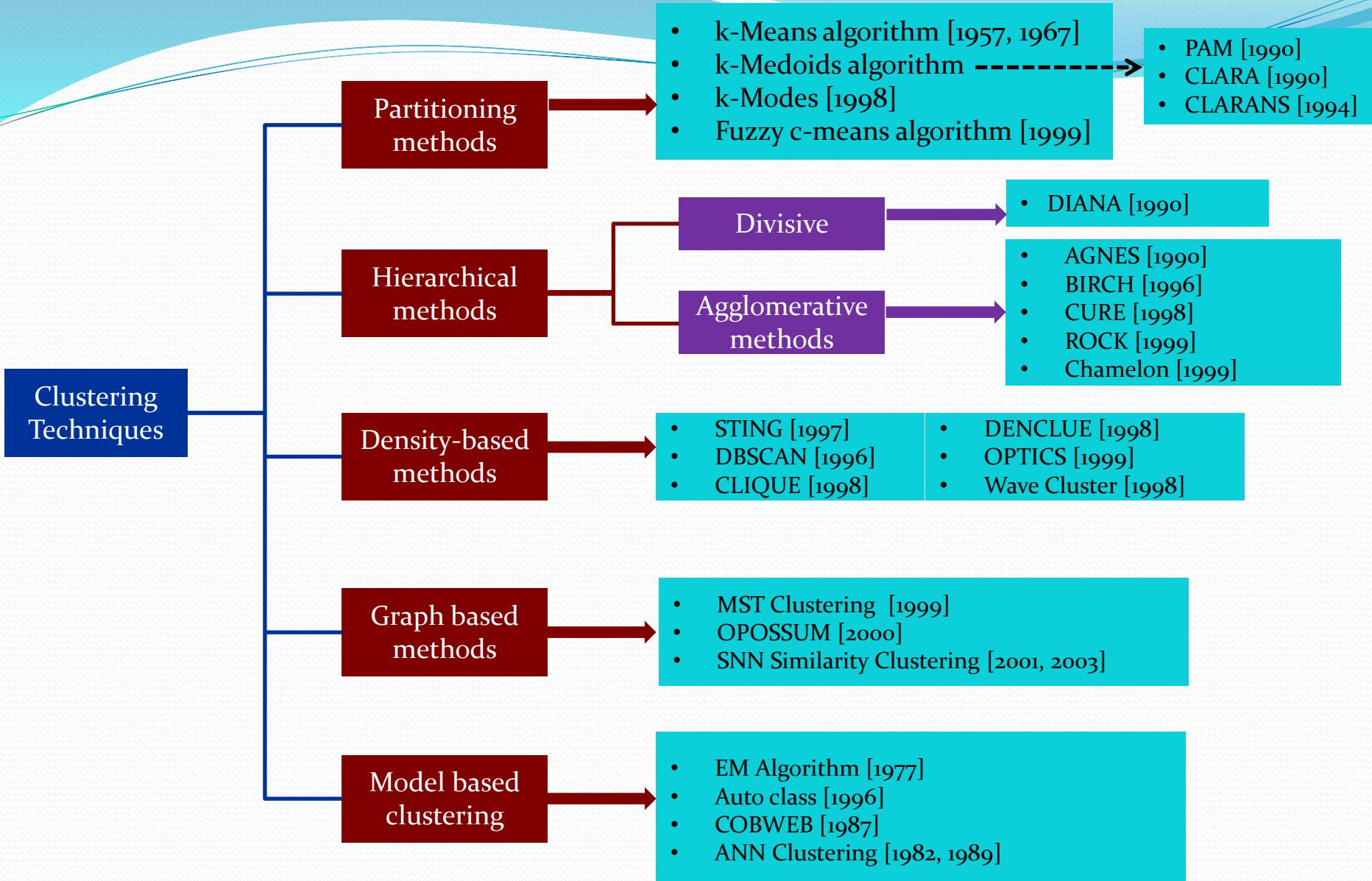
Clustering

Topics to be covered...

- Introduction to clustering
- Similarity and dissimilarity measures
- Clustering techniques
- Partitioning algorithms
- Hierarchical algorithms
- Density-based algorithm

Clustering techniques

- Clustering has been studied extensively for more than 40 years and across many disciplines due to its broad applications.
- As a result, many clustering techniques have been reported in the literature.
- Let us categorize the clustering methods. In fact, it is difficult to provide a crisp categorization because many techniques overlap to each other in terms of clustering paradigms or features.
- A broad taxonomy of existing clustering methods is shown in Fig. 16.1.
- It is not possible to cover all the techniques in this lecture series. We emphasize on major techniques belong to partitioning and hierarchical algorithms.



Clustering techniques

- In this lecture, we shall cover the following clustering techniques only.
 - Partitioning
 - k-Means algorithm
 - PAM (k-Medoids algorithm)
 - Hierarchical
 - DIANA (divisive algorithm)
 - AGNES } (Agglomerative algorithm)
 - ROCK
 - Density – Based
 - DBSCAN

k-Means Algorithm

- k-Means clustering algorithm proposed by J. Hartigan and M. A. Wong [1979].
- Given a set of n distinct objects, the k-Means clustering algorithm partitions the objects into k number of clusters such that intracluster similarity is high but the intercluster similarity is low.
- In this algorithm, user has to specify k , the number of clusters and consider the objects are defined with numeric attributes and thus using any one of the distance metric to demarcate the clusters.

k-Means Algorithm

The algorithm can be stated as follows.

- First it selects k number of objects at random from the set of n objects. These k objects are treated as the **centroids** or center of gravities of k clusters.
- For each of the **remaining objects**, it is assigned to one of the **closest centroid**. Thus, it forms a **collection of objects assigned to each centroid** and is called a **cluster**.
- Next, the centroid of each cluster is then updated (by calculating the mean values of attributes of each object).
- The assignment and update procedure is until it reaches some stopping criteria (such as, number of iteration, centroids remain unchanged or no assignment, etc.)

k-Means Algorithm

Algorithm : k-Means clustering

Input: D is a dataset containing n objects, k is the number of cluster

Output: A set of k clusters

Steps:

1. Randomly choose k objects from D as the initial cluster centroids.
2. **For** each of the objects in D **do**
 - Compute distance between the current objects and k cluster centroids
 - Assign the current object to that cluster to which it is closest.
3. Compute the “cluster centers” of each cluster. These become the new cluster centroids.
4. Repeat step 2-3 until the convergence criterion is satisfied
5. Stop

k-Means Algorithm

Note:

- 1) Objects are defined in terms of set of attributes. $A = \{A_1, A_2, \dots, A_m\}$ where each A_i is continuous data type.
- 2) Distance computation: Any distance such as L_1, L_2, L_3 or cosine similarity.
- 3) Minimum distance is the measure of closeness between an object and centroid.
- 4) Mean Calculation: It is the mean value of each attribute values of all objects.
- 5) Convergence criteria: Any one of the following are termination condition of the algorithm.
 - Number of maximum iteration permissible.
 - No change of centroid values in any cluster.
 - Zero (or no significant) movement(s) of object from one cluster to another.
 - Cluster quality reaches to a certain level of acceptance.

Illustration of k-Means clustering algorithms

Table 16.1: 16 objects with two attributes A_1 and A_2 .

A_1	A_2
6.8	12.6
0.8	9.8
1.2	11.6
2.8	9.6
3.8	9.9
4.4	6.5
4.8	1.1
6.0	19.9
6.2	18.5
7.6	17.4
7.8	12.2
6.6	7.7
8.2	4.5
8.4	6.9
9.0	3.4
9.6	11.1

Fig 16.1: Plotting data of Table 16.1

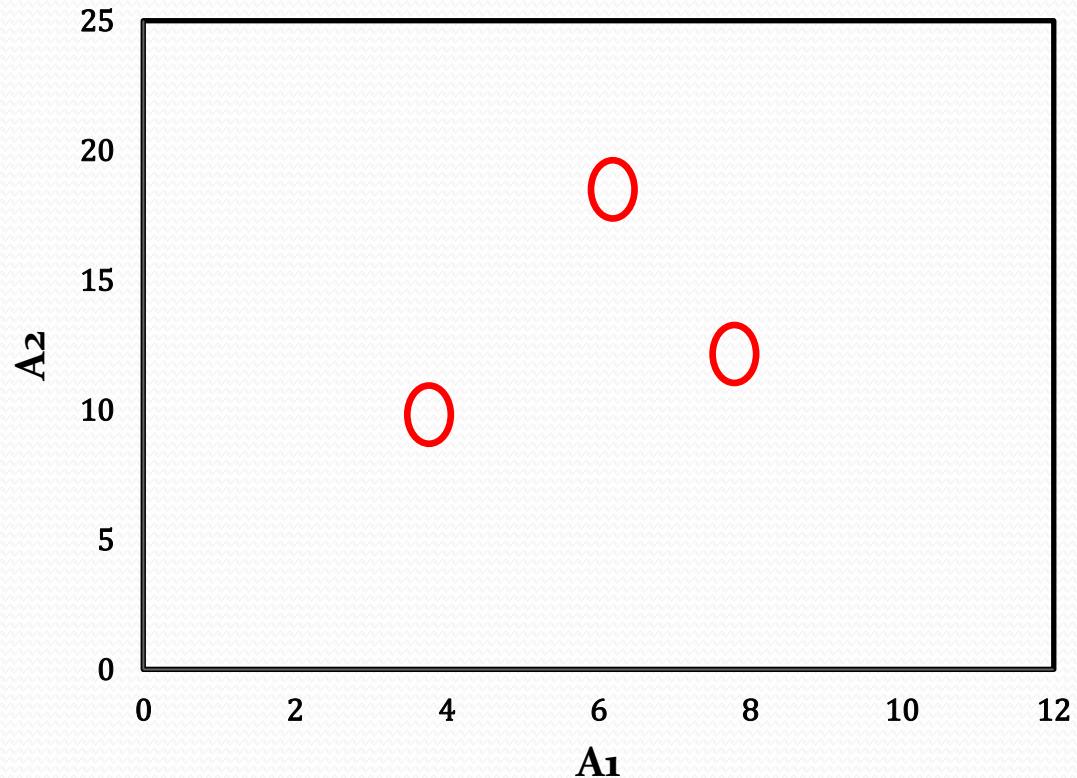


Illustration of k-Means clustering algorithms

- Suppose, $k=3$. Three objects are chosen at random shown as circled (see Fig 16.1). These three centroids are shown below.

Initial Centroids chosen randomly

Centroid	Objects	
	A1	A2
c_1	3.8	9.9
c_2	7.8	12.2
c_3	6.2	18.5

- Let us consider the Euclidean distance measure (L_2 Norm) as the distance measurement in our illustration.
- Let d_1 , d_2 and d_3 denote the distance from an object to c_1 , c_2 and c_3 respectively. The distance calculations are shown in Table 16.2.
- Assignment of each object to the respective centroid is shown in the right-most column and the clustering so obtained is shown in Fig 16.2.

Illustration of k-Means clustering algorithms

Table 16.2: Distance calculation

A ₁	A ₂	d ₁	d ₂	d ₃	cluster
6.8	12.6	4.0	1.1	5.9	2
0.8	9.8	3.0	7.4	10.2	1
1.2	11.6	3.1	6.6	8.5	1
2.8	9.6	1.0	5.6	9.5	1
3.8	9.9	0.0	4.6	8.9	1
4.4	6.5	3.5	6.6	12.1	1
4.8	1.1	8.9	11.5	17.5	1
6.0	19.9	10.2	7.9	1.4	3
6.2	18.5	8.9	6.5	0.0	3
7.6	17.4	8.4	5.2	1.8	3
7.8	12.2	4.6	0.0	6.5	2
6.6	7.7	3.6	4.7	10.8	1
8.2	4.5	7.0	7.7	14.1	1
8.4	6.9	5.5	5.3	11.8	2
9.0	3.4	8.3	8.9	15.4	1
9.6	11.1	5.9	2.1	8.1	2

Fig 16.2: Initial cluster with respect to Table 16.2

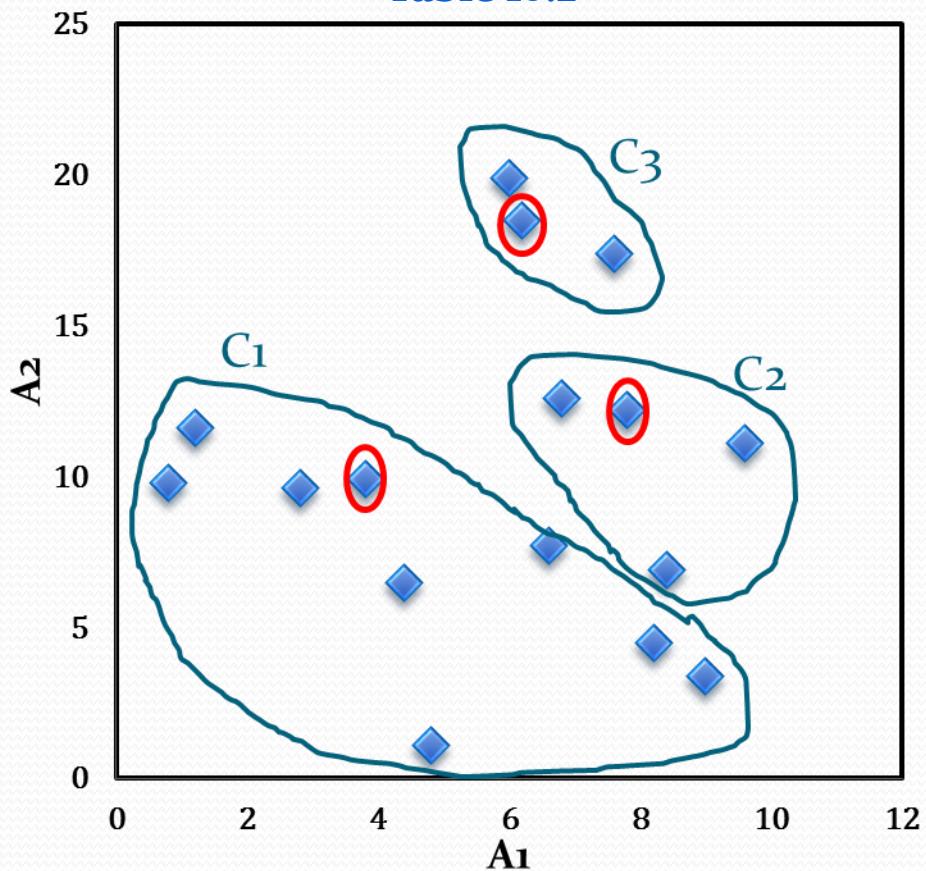


Illustration of k-Means clustering algorithms

The calculation new centroids of the three cluster using the mean of attribute values of A_1 and A_2 is shown in the Table below. The cluster with new centroids are shown in Fig 16.3.

Calculation of new centroids

New Centroid	Objects	
	A_1	A_2
c_1	4.6	7.1
c_2	8.2	10.7
c_3	6.6	18.6

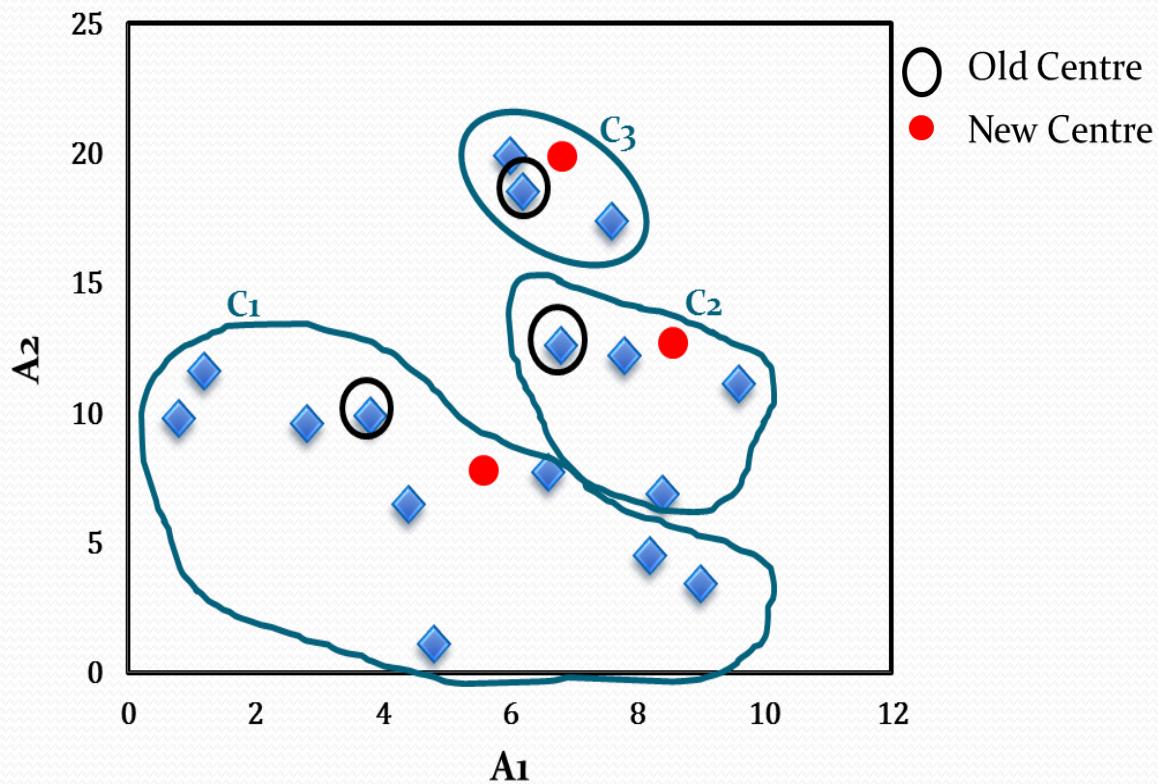


Fig 16.3: Initial cluster with new centroids

Illustration of k-Means clustering algorithms

We next reassign the 16 objects to three clusters by determining which centroid is closest to each one. This gives the revised set of clusters shown in Fig 16.4.

Note that point p moves from cluster C_2 to cluster C_1 .

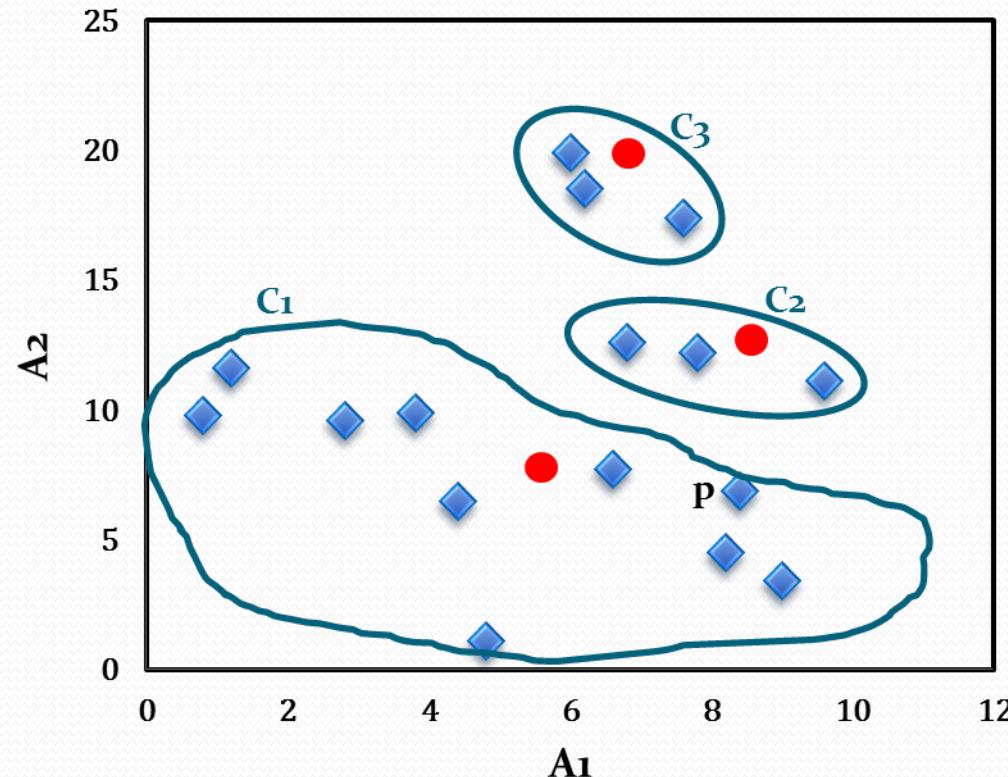


Fig 16.4: Cluster after first iteration

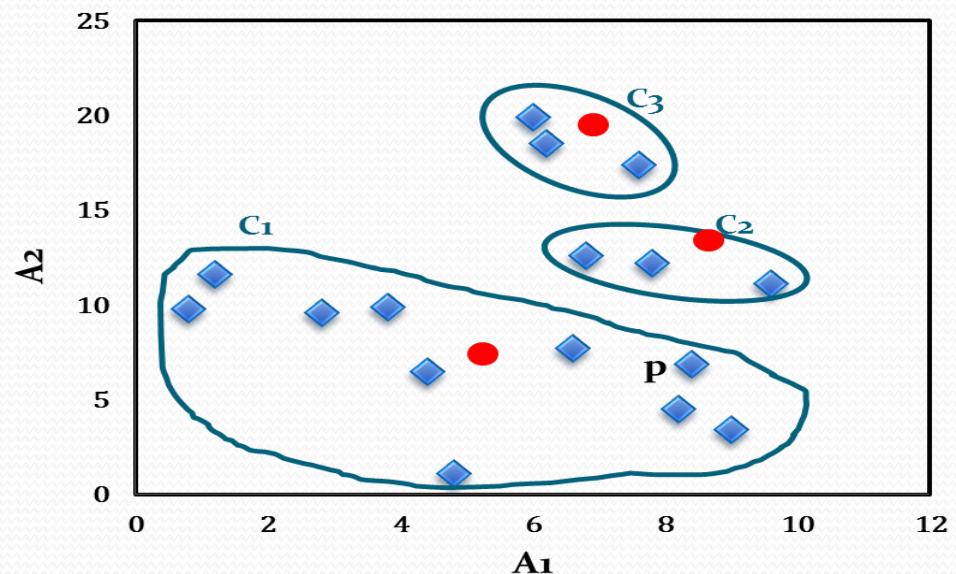
Illustration of k-Means clustering algorithms

- The newly obtained centroids after second iteration are given in the table below. Note that the centroid c_3 remains unchanged, where c_2 and c_1 changed a little.
- With respect to newly obtained cluster centres, 16 points are reassigned again. These are the same clusters as before. Hence, their centroids also remain unchanged.
- Considering this as the termination criteria, the k-means algorithm stops here. Hence, the final cluster in Fig 16.5 is same as Fig 16.4.

Cluster centres after second iteration

Centroid	Revised Centroids	
	A1	A2
c_1	5.0	7.1
c_2	8.1	12.0
c_3	6.6	18.6

Fig 16.5: Cluster after Second iteration



Another Example

- $\mathcal{D} = \{2, 4, 10, 12, 3, 20, 30, 11, 25\}$ and $k = 2$
- initial centroids 2 and 4
- $C_1 = \{2, 3\}, C_2 = \{4, 10, 12, 20, 30, 11, 25\}$ using L_2 distance.
- $m^{C_1} = 2.5, m^{C_2} = 16$
- Next Iteration:
- $C_1 = \{2, 3, 4\}, C_2 = \{10, 12, 20, 30, 11, 25\}$
- $m^{C_1} = 3.0, m^{C_2} = 18$
- $C_1 = \{2, 3, 4, 10\}, C_2 = \{12, 20, 30, 11, 25\}$
- $m^{C_1} = 4.75, m^{C_2} = 19.6$
- $C_1 = \{2, 3, 4, 10, 11, 12\}, C_2 = \{20, 30, 25\}$
- $m^{C_1} = 7, m^{C_2} = 25$
- $C_1 = \{2, 3, 4, 10, 11, 12\}, C_2 = \{20, 30, 25\}$

Comments on k-Means algorithm

Let us analyse the k-Means algorithm and discuss the pros and cons of the algorithm.
We shall refer to the following notations in our discussion.

- **Notations:**

- x : an object under clustering
- n : number of objects under clustering
- C_i : the i -th cluster
- c_i : the centroid of cluster C_i
- n_i : number of objects in the cluster C_i
- c : denotes the centroid of all objects
- k : number of clusters

Comments on k-Means algorithm

1. Value of k:

- The k-means algorithm produces only one set of clusters, for which, user must specify the desired number, k of clusters.
- In fact, k should be the best guess on the number of clusters present in the given data. Choosing the best value of k for a given dataset is, therefore, an issue.
- We may not have an idea about the possible number of clusters for high dimensional data, and for data that are not scatter-plotted.
- Further, possible number of clusters is hidden or ambiguous in image, audio, video and multimedia clustering applications etc.
- There is no principled way to know what the value of k ought to be. We may try with successive value of k starting with 2.
- The process is stopped when two consecutive k values produce more-or-less identical results (with respect to some cluster quality estimation).
- Normally $k \ll n$ and there is heuristic to follow $k \approx \sqrt{n}$.

Comments on k-Means algorithm

Example 16.1: k versus cluster quality

- Usually, there is some objective function to be met as a goal of clustering. One such objective function is **sum-square-error** denoted by **SSE** and defined as

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} (x - c_i)^2$$

- Here, $x - c_i$ denotes the error, if x is in cluster C_i with cluster centroid c_i .
- Usually, this error is measured as distance norms like L_1 , L_2 , L_3 or Cosine similarity, etc.

Comments on k-Means algorithm

Example 16.1: k versus cluster quality

- With reference to an arbitrary experiment, suppose the following results are obtained.

k	SSE
1	62.8
2	12.3
3	9.4
4	9.3
5	9.2
6	9.1
7	9.05
8	9.0

- With respect to this observation, we can choose the value of $k \approx 3$, as with this smallest value of k it gives reasonably good result.
- Note: If $k = n$, then SSE=0; However, the cluster is useless! This is another example of overfitting.

Comments on k-Means algorithm

2. Choosing initial centroids:

- Another requirement in the k-Means algorithm to choose initial cluster centroid for each k would be clusters.
- It is observed that the k-Means algorithm terminate whatever be the initial choice of the cluster centroids.
- It is also observed that initial choice influences the ultimate cluster quality. In other words, the result may be trapped into local optima, if initial centroids are chosen properly.
- One technique that is usually followed [to avoid the above problem](#) is to choose initial centroids in multiple runs, each with a different set of randomly chosen initial centroids, and then select the best cluster (with respect to some quality measurement criterion, e.g. SSE).
- However, this strategy suffers from the combinational explosion problem due to the number of all possible solutions.

Comments on k-Means algorithm

3. Distance Measurement:

- To assign a point to the closest centroid, we need a proximity measure that should quantify the notion of “closest” for the objects under clustering.
- Usually Euclidean distance (L_2 norm) is the best measure when object points are defined in n-dimensional Euclidean space.
- Other measure namely cosine similarity is more appropriate when objects are of document type.
- Further, there may be other type of proximity measures that appropriate in the context of applications.
- For example, Manhattan distance (L_1 norm), Jaccard measure, etc.

Comments on k-Means algorithm

3. Distance Measurement:

Thus, in the context of different measures, the **sum-of-squared error** (i.e., objective function/convergence criteria) of a clustering can be stated as under.

Data in Euclidean space (L_2 norm):

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} (c_i - x)^2$$

Data in Euclidean space (L_1 norm):

The Manhattan distance (L_1 norm) is used as a proximity measure, where the objective is to minimize the **sum-of-absolute error** denoted as **SAE** and defined as

$$SAE = \sum_{i=1}^k \sum_{x \in C_i} |c_i - x|$$

Comments on k-Means algorithm

Distance with document objects

Suppose a set of n document objects is defined as d document term matrix (DTM) (a typical look is shown in the below form).

Document	Term			
	t_1	t_2	t_n
D_1	f_{11}	f_{12}		f_{1n}
D_2	f_{21}	f_{22}		f_{2n}
:				
D_n	f_{n1}	f_{n2}		f_{nn}

Here, the objective function, which is called Total cohesion denoted as TC and defined as

$$TC = \sum_{i=1}^k \sum_{x \in C_i} \cos(x, c_i)$$

$$\text{where } \cos(x, c_i) = \frac{x \cdot c_i}{\|x\| \|c_i\|}$$

$$x \cdot c_i = \sum_j x_j c_{ij} \quad \text{and} \quad \|x\| = \sqrt{\sum_j^p x_j^2}$$

$$\hat{x} = \sum_{j=1}^p \hat{x}_j \quad \hat{c}_i = \sum_{j=1}^p \hat{c}_{ij} \quad \|\hat{c}_{ij}\| = \sqrt{\sum_j^p \hat{c}_{ij}^2}$$

Comments on k-Means algorithm

Note: The criteria of objective function with different proximity measures

1. SSE (using L_2 norm) : To **minimize** the SSE.
2. SAE (using L_1 norm) : To **minimize** the SAE.
3. TC(using cosine similarity) : To **maximize** the TC.

Comments on k-Means algorithm

4. Type of objects under clustering:

- The k-Means algorithm can be applied only when the mean of the cluster is defined (hence it named **k-Means**). The cluster mean (also called centroid) of a cluster C_i is defied as

$$c_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

- In other words, the mean calculation assumed that each object is defined with numerical attribute(s). Thus, we cannot apply the k-Means to objects which are defined with categorical attributes.
- More precisely, the k-means algorithm require some definition of cluster mean exists, but not necessarily it does have as defined in the above equation.
- In fact, the k-Means is a very general clustering algorithm and can be used with a wide variety of data types, such as documents, time series, etc.

? | How to find the mean of objects with composite attributes? |

Comments on k-Means algorithm

Note:

- 1) When SSE (L_2 norm) is used as objective function and the objective is to minimize, then the cluster centroid (i.e. mean) is the mean value of the objects in the cluster.
- 2) When the objective function is defined as SAE (L_1 norm), minimizing the objective function implies the cluster centroid as the median of the cluster.

The above two interpretations can be readily verified as given in the next slide.

Comments on k-Means algorithm

Case 1: SSE

We know,

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} (c_i - x)^2$$

To minimize SSE means, $\frac{\partial(SSE)}{\partial c_i} = 0$

Thus,

$$\frac{\partial}{\partial c_i} \left(\sum_{i=1}^k \sum_{x \in C_i} (c_i - x)^2 \right) = 0$$

Or,

$$\sum_{i=1}^k \sum_{x \in C_i} \frac{\partial}{\partial c_i} (c_i - x)^2 = 0$$

Comments on k-Means algorithm

Or,

$$\sum_{x \in C_i} 2(c_i - x) = 0$$

Or,

$$n_i \cdot c_i = \sum_{x \in C_i} x$$

Or,

$$c_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

- Thus, **the best centroid for minimizing SSE of a cluster is the mean of the objects in the cluster.**

Comments on k-Means algorithm

Case 2: SAE

We know,

$$SAE = \sum_{i=1}^k \sum_{x \in C_i} |c_i - x|$$

To minimize SAE means, $\frac{\partial(SAE)}{\partial c_i} = 0$

Thus,

$$\frac{\partial}{\partial c_i} \left(\sum_{i=1}^k \sum_{x \in C_i} |c_i - x| \right) = 0$$

Or,

$$\sum_{i=1}^k \sum_{x \in C_i} \frac{\partial}{\partial c_i} |c_i - x| = 0$$

Comments on k-Means algorithm

Or,

$$\sum_{x \in C_i} \left\{ (x - c_i) \Big|_{if \ x > c_i} + (c_i - x) \Big|_{if \ c_i > x} \right\} = 0$$

Solving the above equation, we get

$$c_i = median \{x | x \in C_i\}$$

- Thus, the best centroid for minimizing SAE of a cluster is the median of the objects in the cluster.



Interpret the best centroid for maximizing TC (with Cosine similarity measure) of a cluster.

The above mentioned discussion is quite sufficient for the validation of k-Means algorithm.

Comments on k-Means algorithm

5. Complexity analysis of k-Means algorithm

Let us analyse the time and space complexities of k-Means algorithm.

Time complexity:

The time complexity of the k-Means algorithm can be expressed as

$$T(n) = O(n \times m \times k \times l)$$

where n = number of objects

m = number of attributes in the object definition

k = number of clusters

l = number of iterations.

Thus, time requirement is a linear order of number of objects and the algorithm runs in a modest time if $k \ll n$ and $l \ll n$ (the iteration can be moderately controlled to check the value of l).

Comments on k-Means algorithm

5. Complexity analysis of k-Means algorithm

Space complexity: The storage complexity can be expressed as follows.

It requires $n \times m$ space to store the objects and $n \times k$ space to store the proximity measure from n objects to the centroids of k clusters.

Thus the total storage complexity is

$$S(n) = O(n \times (m + k))$$

That is, space requirement is in the linear order of n if $k \ll n$.

Comments on k-Means algorithm

6. Final comments:

Advantages:

- k-Means is simple and can be used for a wide variety of object types.
- It is also efficient both from storage requirement and execution time point of views. By saving distance information from one iteration to the next, the actual number of distance calculations, that must be made can be reduced (specially, as it reaches towards the termination).



How similarity metric can be utilized to run k-Means faster? What is the updation in each iteration?

Limitations:

- The k-Means is not suitable for all types of data. For example, k-Means does not work on categorical data because mean cannot be defined.
- k-means finds a local optima and may actually minimize the global optimum.

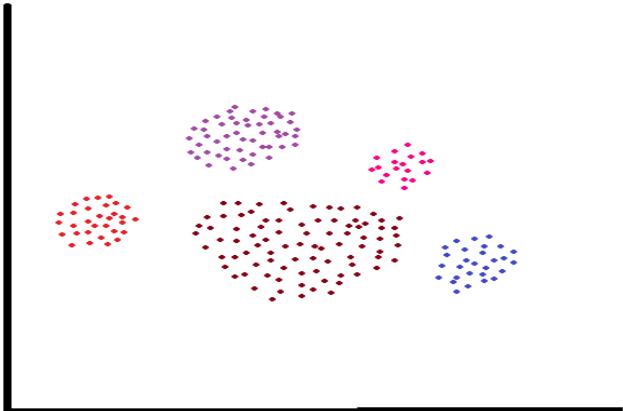
Comments on k-Means algorithm

6. Final comments:

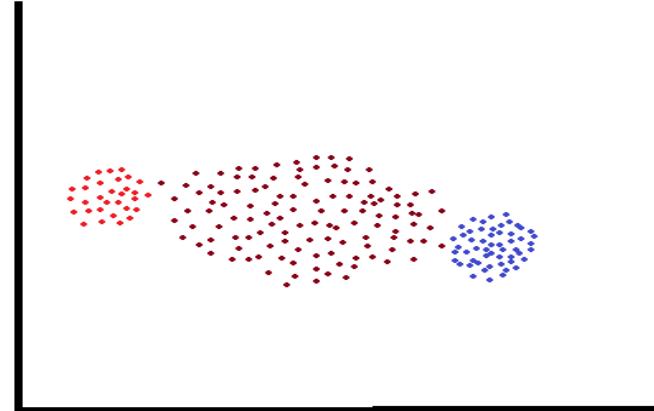
Limitations :

- k-means has trouble clustering data that contains outliers. When the SSE is used as objective function, outliers can unduly influence the cluster that are produced. More precisely, in the presence of outliers, the cluster centroids, in fact, not truly as representative as they would be otherwise. It also influence the SSE measure as well.
- k-Means algorithm cannot handle non-globular clusters, clusters of different sizes and densities (see Fig 16.6 in the next slide).
- k-Means algorithm not really beyond the scalability issue (and not so practical for large databases).

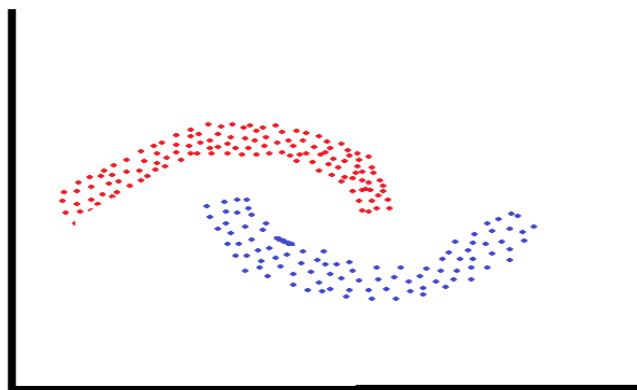
Comments on k-Means algorithm



Cluster with different sizes



Cluster with different densities



Non-convex shaped clusters

Fig 16.6: Some failure instance of k-Means algorithm

Different variants of k-means algorithm

There are quite a few variants of the k-Means algorithm. These can differ in the procedure of selecting the initial k means, the calculation of proximity and strategy for calculating cluster means. Another variants of k-means to cluster categorical data.

Few variant of k-Means algorithm includes

- Bisecting k-Means (addressing the issue of initial choice of cluster means).
 1. M. Steinbach, G. Karypis and V. Kumar “A comparison of document clustering techniques”, *Proceedings of KDD workshop on Text mining*, 2000.
- Mean of clusters (Proposing various strategies to define means and variants of means).
 - B. zhan “Generalised k-Harmonic means – Dynamic weighting of data in unsupervised learning”, *Technical report, HP Labs*, 2000.
 - A. D. Chaturvedi, P. E. Green, J. D. Carroll, “k-Modes clustering”, *Journal of classification*, Vol. 18, PP. 35-36, 2001.
 - D. Pelleg, A. Moore, “x-Means: Extending k-Means with efficient estimation of the number of clusters”, *17th International conference on Machine Learning*, 2000.

Different variants of k-means algorithm

- N. B. Karayiannis, M. M. Randolph, “Non-Euclidean c-Means clustering algorithm”, *Intelligent data analysis journal*, Vol 7(5), PP 405-425, 2003.
- V. J. Olivera, W. Pedrycy, “Advances in Fuzzy clustering and its applications”, Edited book. John Wiley [2007]. (Fuzzy c-Means algorithm).
- A. K. Jain and R. C. Bubes, “Algorithms for clustering Data”, Prentice Hall, 1988.
Online book at http://www.cse.msu.edu/~jain/clustering_Jain_Dubes.pdf
- A. K. Jain, M. N. Munty and P. J. Flynn, “Data clustering: A Review”, *ACM computing surveys*, 31(3), 264-323 [1999]. Also available online.

The k-Medoids algorithm

Now, we shall study a variant of partitioning algorithm called k-Medoids algorithm.

Motivation: We have learnt that the k-Means algorithm is sensitive to outliers because an object with an “extremely large value” may substantially distort the distribution. The effect is particularly exacerbated due to the use of the SSE (sum-of-squared error) objective function. The k-Medoids algorithm aims to diminish the effect of outliers.

Basic concepts:

- The basic concepts of this algorithm is to **select an object as a cluster center** (one representative object per cluster) instead of taking the mean value of the objects in a cluster (as in k-Means algorithm).
- We call this cluster representative as a **cluster medoid** or simply **medoid**.
 1. Initially, it selects a random set of k objects as the set of medoids.
 2. Then at each step, all objects from the set of objects, which are not currently medoids are examined one by one to see if they should be medoids.

The k-Medoids algorithm

- That is, the k-Medoids algorithm **determines** whether there is an object that should replace one of the current medoids.
- This is accomplished by looking all pair of medoid, non-medoid objects, and then choosing a pair that improves the objective function of clustering the best and exchange them.
- The sum-of-absolute error (SAE) function is used as the objective function.

$$SAE = \sum_{i=1}^k \sum_{x \in C_i, x \notin M \text{ and } c_m \in M} |x - c_m|$$

Where c_m denotes a medoid

M is the set of all medoids at any instant

x is an object belongs to set of non-medoid object, that is, x belongs to some cluster and is not a medoid. i.e. $x \in C_i, x \notin M$

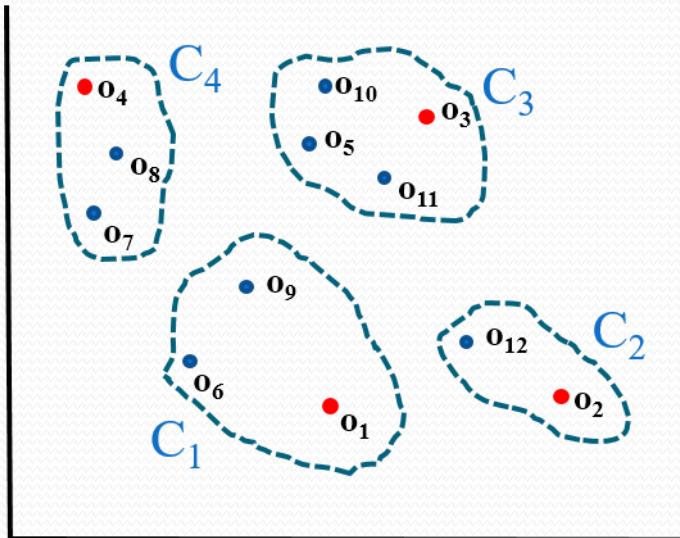
PAM (Partitioning around Medoids)

- For a given set of medoids, at any iteration, it select that exchange which has minimum SAE.
- The procedure terminates, if there is no any change in SAE in successive iteration (i.e. there is no change in medoid).
- This k-Medoids algorithm is also known as PAM (Partitioning around Medoids).

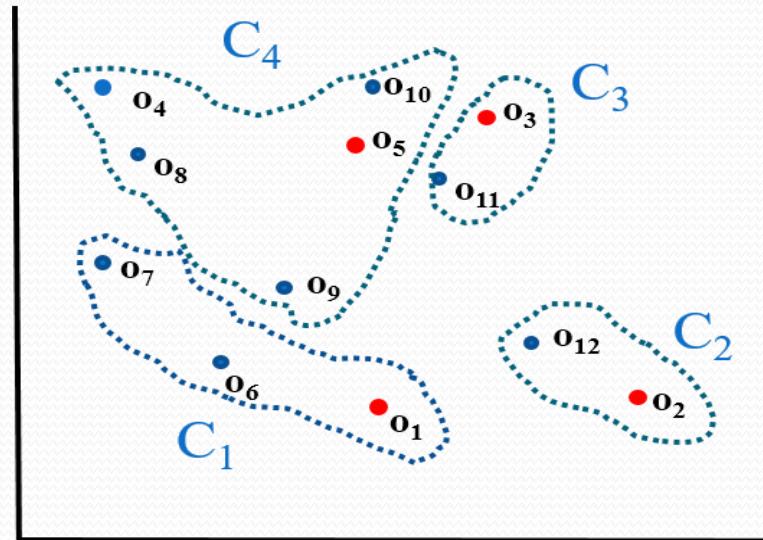
Illustration of PAM

- Suppose, there are set of 12 objects $O(o_1, o_2, \dots, o_{12})$ and we are to cluster them into four clusters. At any instant, the four cluster C_1, C_2, C_3 and C_4 are shown in Fig. 16.7 (a). Also assume that o_1, o_2, o_3 , and o_4 are the medoids in the clusters C_1, C_2, C_3 and C_4 , respectively. For this clustering we can calculate SAE.
- There are many ways to choose a non-medoid object to be replaced any one medoid object. Out of these, suppose, if o_5 is considered as candidate medoid instead of o_4 , then it gives the lowest SAE. Thus, the new set of medoids would be o_1, o_2, o_3 , and o_5 . The new cluster is shown in Fig 16.7 (b).

PAM (Partitioning around Medoids)



(a) Cluster with o_1 , o_2 , o_3 , and o_4 as medoids



(b) Cluster after swapping o_4 and o_5 (o_5 becomes the new medoid).

Fig 16.7: Illustration of PAM

PAM (Partitioning around Medoids)

PAM algorithm is thus a procedure of iterative selection of medoids and it is precisely stated in Algorithm 16.2.

Algorithm 16.2: PAM

Input: Database of objects D.

k, the number of desired clusters.

Output: Set of k clusters

Steps:

1. Arbitrarily select k medoids from D.
2. **For** each object o_i not a medoid **do**
 3. **For** each medoid o_j **do**
 4. Let $M = \{o_1, o_2, \dots, o_{i-1}, o_i, o_{i+1}, o_k\}$ //Set of current medoids
 $M' = \{o_1, o_2, \dots, o_{j-1}, o_j, o_{j+1}, o_k\}$ //set of medoids but swap with non-medoids o_j
 5. Calculate $\text{cost}(o_i, o_j) = SAE|_M - SAE_M,$
 6. **End** of 2 for loop

PAM (Partitioning around Medoids)

Algorithm 16.2: PAM

7. Find o_i, o_j for which the $\text{cost}(o_i, o_j)$ is the smallest.
8. Replace o_i with o_j and accordingly update the set M .
9. Repeat step 2 - step 8 until $\text{cost}(o_i, o_i) \leq 0$.
10. Return the cluster with M as the set of cluster centers.
11. Stop

Comments on PAM

1. Comparing k-Means with k-Medoids:

- Both algorithms needs to fix k , the number of cluster prior to the algorithms. Also, oth algorithm arbitrarily choose the initial cluster centroids.
- The k-Medoid method is more robust than k-Means in the presence of outliers, because a medoid is less influenced by outliers than a mean.

2. Time complexity of PAM:

- For each iteration, PAM consider $k(n - k)$ pairs of object o_i, o_j for which a cost $\text{cost}(o_i, o_j)$ determines. Calculating the cost during each iteration requires that the cost be calculated for all other non-medoids o_j . There are $n - k$ of these. Thus, the total time complexity per iteration is $n(n - k)^2$. The total number of iterations may be quite large.

3. Applicability of PAM:

- PAM does not scale well to large database because of its computation complexity.

Other variants of k-Medoids algorithms

- There are some variants of PAM that are targeted mainly large datasets are **CLARA** (*Clustering LARge Applications*) and **CLARANS** (*Clustering Large Applications based upon RANdomized Search*), it is an improvement of CLARA.

References:

For PAM and CLARA:

- L. kaufman and P. J. Roussew, “Finding Groups in Data: An introduction to cluster analysis”, John and Wiley, 1990.

For CLARANS:

- R. Ng and J. Han, “Efficient and effective clustering method for spatial Data mining”, Proceeding very large databases [VLDB-94], 1994.

Any question?

Nonparametric Methods

Lecture 5

Jason Corso

SUNY at Buffalo

17 Feb. 2009

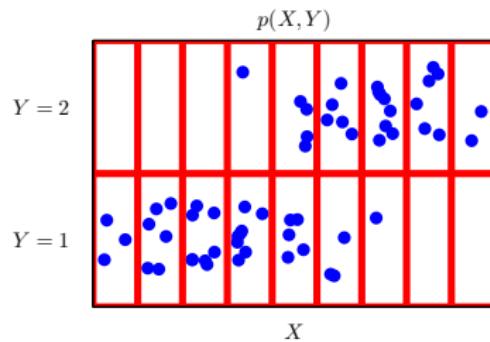
Nonparametric Methods Lecture 5 Overview

- Previously, we've assumed that the forms of the underlying densities were of some particular known parametric form.
- **But, what if this is not the case?**
- Indeed, for most real-world pattern recognition scenarios this assumption is suspect.
- For example, most real-world entities have multimodal distributions whereas all classical parametric densities are unimodal.

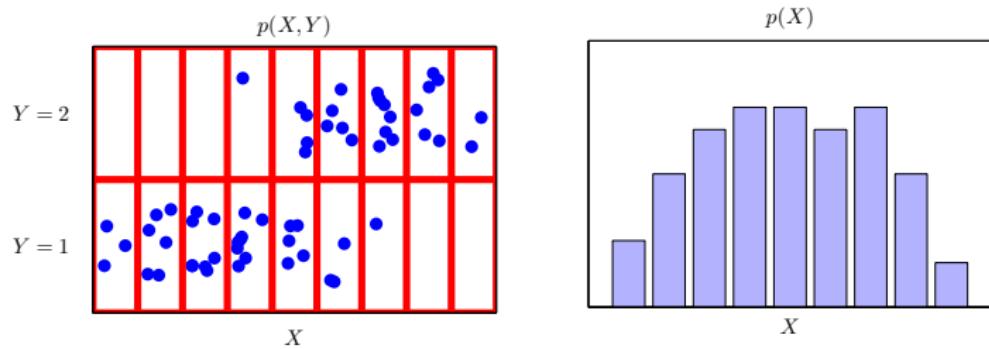
Nonparametric Methods Lecture 5 Overview

- Previously, we've assumed that the forms of the underlying densities were of some particular known parametric form.
- **But, what if this is not the case?**
- Indeed, for most real-world pattern recognition scenarios this assumption is suspect.
- For example, most real-world entities have multimodal distributions whereas all classical parametric densities are unimodal.
- We will examine **nonparametric** procedures that can be used with arbitrary distributions and without the assumption that the underlying form of the densities are known.
 - Histograms.
 - Kernel Density Estimation / Parzen Windows.
 - k-Nearest Neighbor Density Estimation.
 - Real Example in Figure-Ground Segmentation

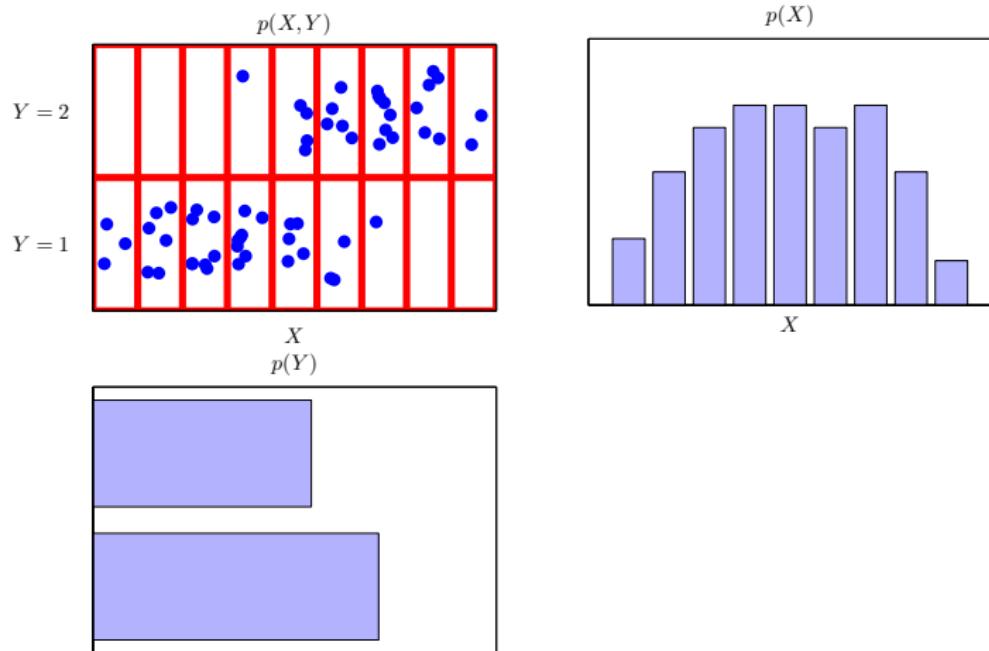
Histograms



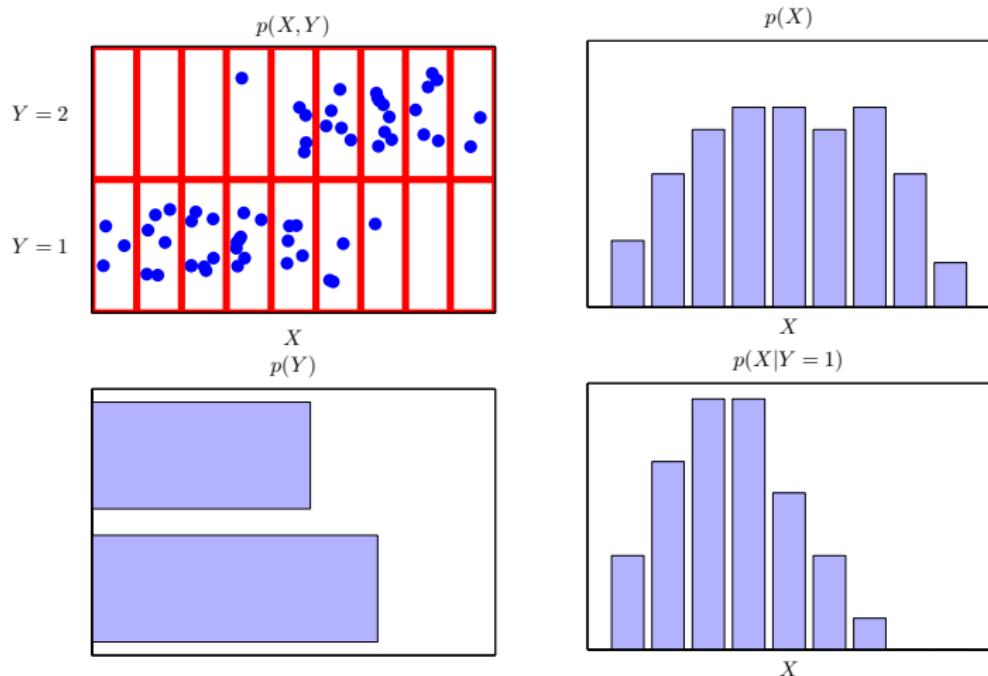
Histograms



Histograms



Histograms



Histogram Density Representation

- Consider a single continuous variable x and let's say we have a set \mathcal{D} of N of them $\{x_1, \dots, x_N\}$. Our goal is to model $p(x)$ from \mathcal{D} .

Histogram Density Representation

- Consider a single continuous variable x and let's say we have a set \mathcal{D} of N of them $\{x_1, \dots, x_N\}$. Our goal is to model $p(x)$ from \mathcal{D} .
- Standard histograms simply partition x into distinct bins of width Δ_i and then count the number n_i of observations x falling into bin i .

Histogram Density Representation

- Consider a single continuous variable x and let's say we have a set \mathcal{D} of N of them $\{x_1, \dots, x_N\}$. Our goal is to model $p(x)$ from \mathcal{D} .
- Standard histograms simply partition x into distinct bins of width Δ_i and then count the number n_i of observations x falling into bin i .
- To turn this count into a normalized probability density, we simply divide by the total number of observations N and by the width Δ_i of the bins.
- This gives us:

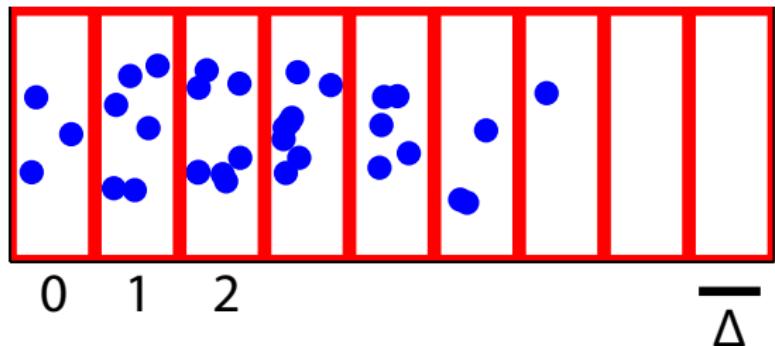
$$p_i = \frac{n_i}{N\Delta_i} \quad (1)$$

Histogram Density Representation

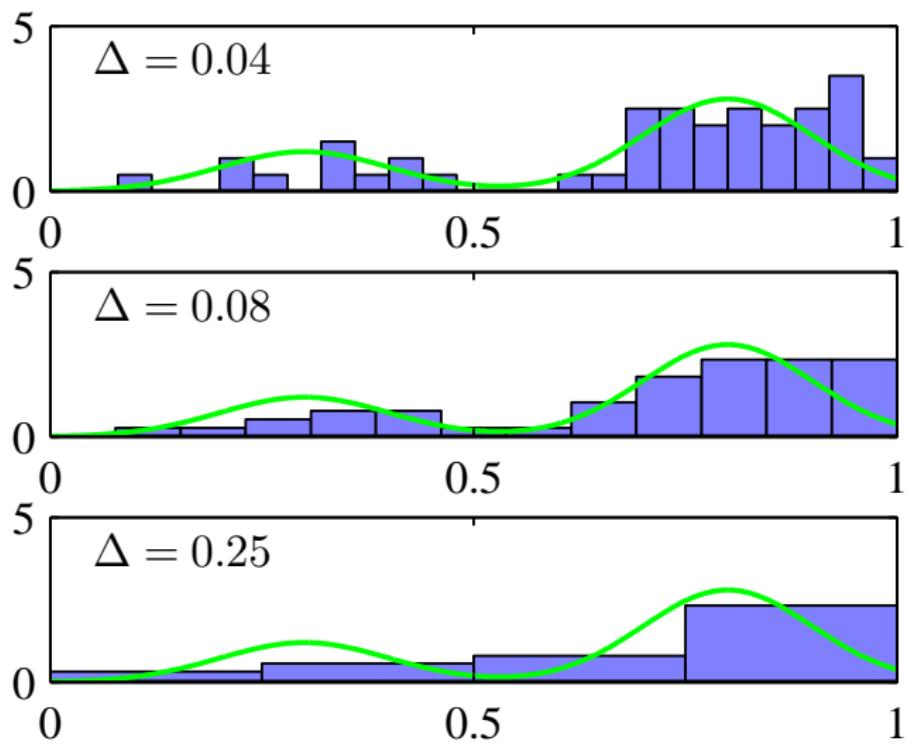
- Consider a single continuous variable x and let's say we have a set \mathcal{D} of N of them $\{x_1, \dots, x_N\}$. Our goal is to model $p(x)$ from \mathcal{D} .
- Standard histograms simply partition x into distinct bins of width Δ_i and then count the number n_i of observations x falling into bin i .
- To turn this count into a normalized probability density, we simply divide by the total number of observations N and by the width Δ_i of the bins.
- This gives us:

$$p_i = \frac{n_i}{N\Delta_i} \tag{1}$$

- Hence the model for the density $p(x)$ is constant over the width of each bin. (And often the bins are chosen to have the same width $\Delta_i = \Delta$.)

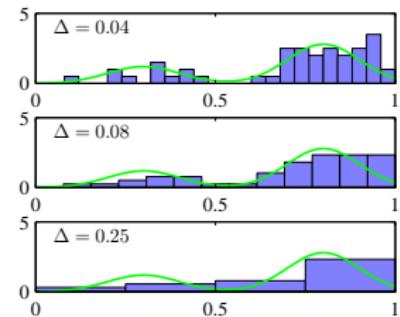


Histogram Density as a Function of Bin Width



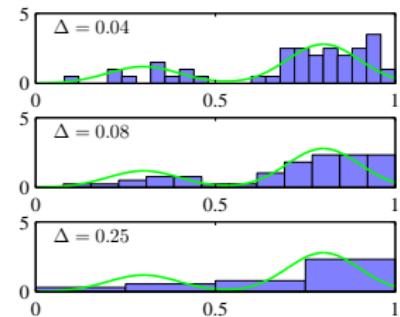
Histogram Density as a Function of Bin Width

- The green curve is the underlying true density from which the samples were drawn. It is a mixture of two Gaussians.



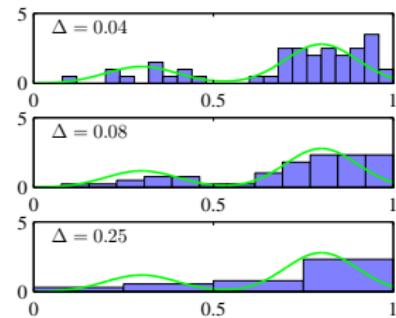
Histogram Density as a Function of Bin Width

- The green curve is the underlying true density from which the samples were drawn. It is a mixture of two Gaussians.
- When Δ is very small (top), the resulting density is quite spiky and hallucinates a lot of structure not present in $p(x)$.



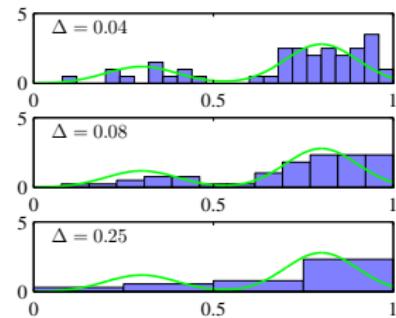
Histogram Density as a Function of Bin Width

- The green curve is the underlying true density from which the samples were drawn. It is a mixture of two Gaussians.
- When Δ is very small (top), the resulting density is quite spiky and hallucinates a lot of structure not present in $p(x)$.
- When Δ is very big (bottom), the resulting density is quite smooth and consequently fails to capture the bimodality of $p(x)$.



Histogram Density as a Function of Bin Width

- The green curve is the underlying true density from which the samples were drawn. It is a mixture of two Gaussians.
- When Δ is very small (top), the resulting density is quite spiky and hallucinates a lot of structure not present in $p(x)$.
- When Δ is very big (bottom), the resulting density is quite smooth and consequently fails to capture the bimodality of $p(x)$.
- It appears that the *best results* are obtained for some intermediate value of Δ , which is given in the middle figure.



Histogram Density as a Function of Bin Width

- The green curve is the underlying true density from which the samples were drawn. It is a mixture of two Gaussians.
 - When Δ is very small (top), the resulting density is quite spiky and hallucinates a lot of structure not present in $p(x)$.
 - When Δ is very big (bottom), the resulting density is quite smooth and consequently fails to capture the bimodality of $p(x)$.
 - It appears that the *best results* are obtained for some intermediate value of Δ , which is given in the middle figure.
 - In principle, a histogram density model is also dependent on the choice of the edge location of each bin.
-
- Detailed description of the figure: The figure consists of three vertically stacked subplots. Each subplot shows a histogram estimate (blue bars) overlaid with a green curve representing the true density $p(x)$. The x-axis for all plots is labeled from 0 to 1. The y-axis for the top plot is labeled 5, and for the middle and bottom plots is labeled 0. The top subplot is for $\Delta = 0.04$, showing a highly spiky histogram that fails to capture the underlying bimodal structure. The middle subplot is for $\Delta = 0.08$, showing a histogram that captures the bimodal structure well. The bottom subplot is for $\Delta = 0.25$, showing a very smooth histogram that fails to capture the bimodality.

Analyzing the Histogram Density

- What are the advantages and disadvantages of the histogram density estimator?

Analyzing the Histogram Density

- What are the advantages and disadvantages of the histogram density estimator?
- Advantages:
 - Simple to evaluate and simple to use.
 - One can throw away \mathcal{D} once the histogram is computed.
 - Can be computed sequentially if data continues to come in.

Analyzing the Histogram Density

- What are the advantages and disadvantages of the histogram density estimator?
- Advantages:
 - Simple to evaluate and simple to use.
 - One can throw away \mathcal{D} once the histogram is computed.
 - Can be computed sequentially if data continues to come in.
- Disadvantages:
 - The estimated density has discontinuities due to the bin edges rather than any property of the underlying density.
 - Scales poorly (curse of dimensionality): we would have M^D bins if we divided each variable in a D -dimensional space into M bins.

What can we learn from Histogram Density Estimation?

- Lesson 1: To estimate the probability density at a particular location, we should consider the data points that lie within some local neighborhood of that point.
 - This requires we define some distance measure.
 - There is a natural smoothness parameter describing the spatial extent of the regions (this was the bin width for the histograms).

What can we learn from Histogram Density Estimation?

- Lesson 1: To estimate the probability density at a particular location, we should consider the data points that lie within some local neighborhood of that point.
 - This requires we define some distance measure.
 - There is a natural smoothness parameter describing the spatial extent of the regions (this was the bin width for the histograms).
- Lesson 2: The value of the smoothing parameter should neither be too large or too small in order to obtain good results.

What can we learn from Histogram Density Estimation?

- Lesson 1: To estimate the probability density at a particular location, we should consider the data points that lie within some local neighborhood of that point.
 - This requires we define some distance measure.
 - There is a natural smoothness parameter describing the spatial extent of the regions (this was the bin width for the histograms).
- Lesson 2: The value of the smoothing parameter should neither be too large or too small in order to obtain good results.
- With these two lessons in mind, we proceed to kernel density estimation and nearest neighbor density estimation, two closely related methods for density estimation.

The Space-Averaged / Smoothed Density

- Consider again samples \mathbf{x} from underlying density $p(\mathbf{x})$.
- Let \mathcal{R} denote a small region containing \mathbf{x} .

The Space-Averaged / Smoothed Density

- Consider again samples \mathbf{x} from underlying density $p(\mathbf{x})$.
- Let \mathcal{R} denote a small region containing \mathbf{x} .
- The probability mass associated with \mathcal{R} is given by

$$P = \int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \tag{2}$$

The Space-Averaged / Smoothed Density

- Consider again samples \mathbf{x} from underlying density $p(\mathbf{x})$.
- Let \mathcal{R} denote a small region containing \mathbf{x} .
- The probability mass associated with \mathcal{R} is given by

$$P = \int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \tag{2}$$

- Suppose we have n samples $\mathbf{x} \in \mathcal{D}$. The probability of each sample falling into \mathcal{R} is P .
- How will the total number of k points falling into \mathcal{R} be distributed?

The Space-Averaged / Smoothed Density

- Consider again samples \mathbf{x} from underlying density $p(\mathbf{x})$.
- Let \mathcal{R} denote a small region containing \mathbf{x} .
- The probability mass associated with \mathcal{R} is given by

$$P = \int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \quad (2)$$

- Suppose we have n samples $\mathbf{x} \in \mathcal{D}$. The probability of each sample falling into \mathcal{R} is P .
- How will the total number of k points falling into \mathcal{R} be distributed?
- This will be a **binomial distribution**:

$$P_k = \binom{n}{k} P^k (1 - P)^{n-k} \quad (3)$$

The Space-Averaged / Smoothed Density

- The expected value for k is thus

$$\mathcal{E}[k] = nP \quad (4)$$

The Space-Averaged / Smoothed Density

- The expected value for k is thus

$$\mathcal{E}[k] = nP \quad (4)$$

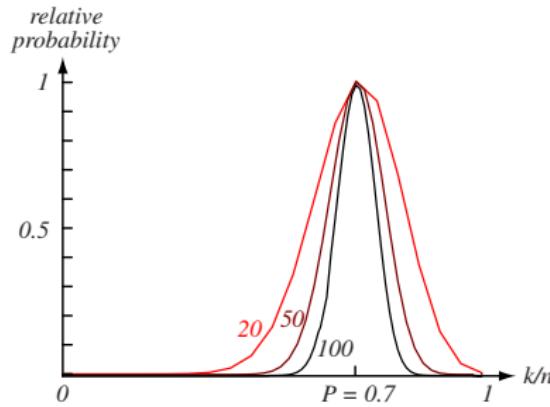
- The binomial for k peaks very sharply about the mean. So, we expect k/n to be a very good estimate for the probability P (and thus for the space-averaged density).

The Space-Averaged / Smoothed Density

- The expected value for k is thus

$$\mathcal{E}[k] = nP \quad (4)$$

- The binomial for k peaks very sharply about the mean. So, we expect k/n to be a very good estimate for the probability P (and thus for the space-averaged density).
- This estimate is increasingly accurate as n increases.



The Space-Averaged / Smoothed Density

- Assuming continuous $p(\mathbf{x})$ and that \mathcal{R} is so small that $p(\mathbf{x})$ does not appreciably vary within it, we can write:

$$\int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \simeq p(\mathbf{x})V \quad (5)$$

where \mathbf{x} is a point within \mathcal{R} and V is the volume enclosed by \mathcal{R} .

The Space-Averaged / Smoothed Density

- Assuming continuous $p(\mathbf{x})$ and that \mathcal{R} is so small that $p(\mathbf{x})$ does not appreciably vary within it, we can write:

$$\int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \simeq p(\mathbf{x})V \quad (5)$$

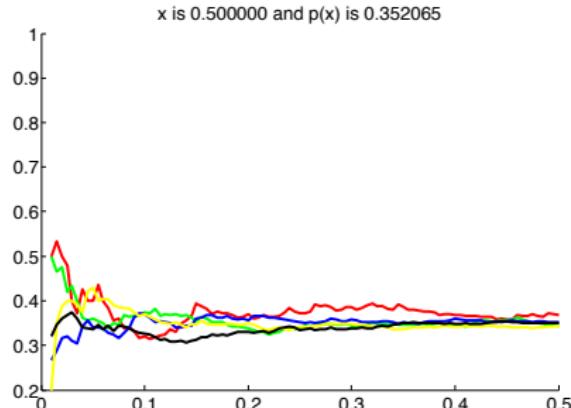
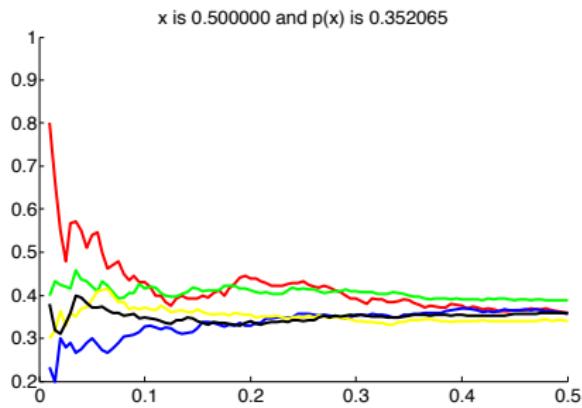
where \mathbf{x} is a point within \mathcal{R} and V is the volume enclosed by \mathcal{R} .

- After some rearranging, we get the following estimate for $p(\mathbf{x})$

$$p(\mathbf{x}) \simeq \frac{k}{nV} \quad (6)$$

Example

- Simulated an example of example the density at 0.5 for an underlying zero-mean, unit variance Gaussian.
- Varied the volume used to estimate the density.
- Red=1000, Green=2000, Blue=3000, Yellow=4000, Black=5000.



Practical Concerns

- The validity of our estimate depends on two contradictory assumptions:
 - ① The region \mathcal{R} must be sufficiently small the the density is approximately constant over the region.
 - ② The region \mathcal{R} must be sufficiently large that the number k of points falling inside it is sufficient to yield a sharply peaked binomial.

Practical Concerns

- The validity of our estimate depends on two contradictory assumptions:
 - ① The region \mathcal{R} must be sufficiently small the the density is approximately constant over the region.
 - ② The region \mathcal{R} must be sufficiently large that the number k of points falling inside it is sufficient to yield a sharply peaked binomial.
- Another way of looking it is to fix the volume V and increase the number of training samples. Then, the ratio k/n will converge as desired. But, this will only yield an estimate of the space-averaged density (P/V).

Practical Concerns

- The validity of our estimate depends on two contradictory assumptions:
 - ① The region \mathcal{R} must be sufficiently small the the density is approximately constant over the region.
 - ② The region \mathcal{R} must be sufficiently large that the number k of points falling inside it is sufficient to yield a sharply peaked binomial.
- Another way of looking it is to fix the volume V and increase the number of training samples. Then, the ratio k/n will converge as desired. But, this will only yield an estimate of the space-averaged density (P/V).
- We want $p(\mathbf{x})$, so we need to let V approach 0. However, with a fixed n , \mathcal{R} will become so small, that no points will fall into it and our estimate would be useless: $p(\mathbf{x}) \simeq 0$.

Practical Concerns

- The validity of our estimate depends on two contradictory assumptions:
 - ① The region \mathcal{R} must be sufficiently small the the density is approximately constant over the region.
 - ② The region \mathcal{R} must be sufficiently large that the number k of points falling inside it is sufficient to yield a sharply peaked binomial.
- Another way of looking it is to fix the volume V and increase the number of training samples. Then, the ratio k/n will converge as desired. But, this will only yield an estimate of the space-averaged density (P/V).
- We want $p(\mathbf{x})$, so we need to let V approach 0. However, with a fixed n , \mathcal{R} will become so small, that no points will fall into it and our estimate would be useless: $p(\mathbf{x}) \simeq 0$.
- Note that in practice, we cannot let V to become arbitrarily small because the number of samples is always limited.

How can we skirt these limitations when an unlimited number of samples if available?

- To estimate the density at x , form a sequence of regions $\mathcal{R}_1, \mathcal{R}_2, \dots$ containing x with the \mathcal{R}_1 having 1 sample, \mathcal{R}_2 having 2 samples and so on.

How can we skirt these limitations when an unlimited number of samples if available?

- To estimate the density at x , form a sequence of regions $\mathcal{R}_1, \mathcal{R}_2, \dots$ containing x with the \mathcal{R}_1 having 1 sample, \mathcal{R}_2 having 2 samples and so on.
- Let V_n be the volume of \mathcal{R}_n , k_n be the number of samples falling in \mathcal{R}_n , and $p_n(x)$ be the n th estimate for $p(x)$:

$$p_n(x) = \frac{k_n}{nV_n} \quad (7)$$

How can we skirt these limitations when an unlimited number of samples if available?

- To estimate the density at x , form a sequence of regions $\mathcal{R}_1, \mathcal{R}_2, \dots$ containing x with the \mathcal{R}_1 having 1 sample, \mathcal{R}_2 having 2 samples and so on.
- Let V_n be the volume of \mathcal{R}_n , k_n be the number of samples falling in \mathcal{R}_n , and $p_n(x)$ be the n th estimate for $p(x)$:

$$p_n(x) = \frac{k_n}{nV_n} \quad (7)$$

- If $p_n(x)$ is to converge to $p(x)$ we need the following three conditions

$$\lim_{n \rightarrow \infty} V_n = 0 \quad (8)$$

$$\lim_{n \rightarrow \infty} k_n = \infty \quad (9)$$

$$\lim_{n \rightarrow \infty} k_n/n = 0 \quad (10)$$

- $\lim_{n \rightarrow \infty} V_n = 0$ ensures that our space-averaged density will converge to $p(\mathbf{x})$.

- $\lim_{n \rightarrow \infty} V_n = 0$ ensures that our space-averaged density will converge to $p(\mathbf{x})$.
- $\lim_{n \rightarrow \infty} k_n = \infty$ basically ensures that the frequency ratio will converge to the probability P (the binomial will be sufficiently peaked).

- $\lim_{n \rightarrow \infty} V_n = 0$ ensures that our space-averaged density will converge to $p(\mathbf{x})$.
- $\lim_{n \rightarrow \infty} k_n = \infty$ basically ensures that the frequency ratio will converge to the probability P (the binomial will be sufficiently peaked).
- $\lim_{n \rightarrow \infty} k_n/n = 0$ is required for $p_n(\mathbf{x})$ to converge at all. It also says that although a huge number of samples will fall within the region \mathcal{R}_n , they will form a negligibly small fraction of the total number of samples.

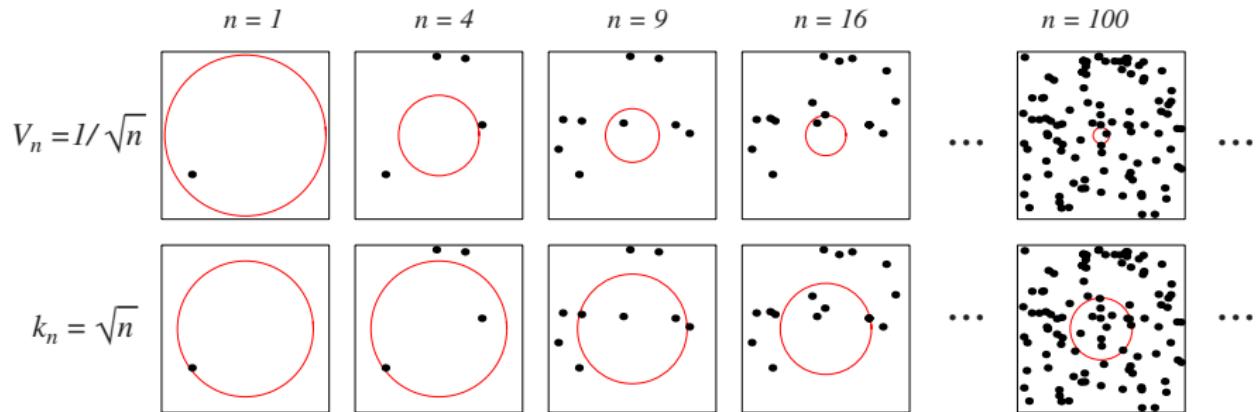
- $\lim_{n \rightarrow \infty} V_n = 0$ ensures that our space-averaged density will converge to $p(\mathbf{x})$.
- $\lim_{n \rightarrow \infty} k_n = \infty$ basically ensures that the frequency ratio will converge to the probability P (the binomial will be sufficiently peaked).
- $\lim_{n \rightarrow \infty} k_n/n = 0$ is required for $p_n(\mathbf{x})$ to converge at all. It also says that although a huge number of samples will fall within the region \mathcal{R}_n , they will form a negligibly small fraction of the total number of samples.
- There are two common ways of obtaining regions that satisfy these conditions:

- $\lim_{n \rightarrow \infty} V_n = 0$ ensures that our space-averaged density will converge to $p(\mathbf{x})$.
- $\lim_{n \rightarrow \infty} k_n = \infty$ basically ensures that the frequency ratio will converge to the probability P (the binomial will be sufficiently peaked).
- $\lim_{n \rightarrow \infty} k_n/n = 0$ is required for $p_n(\mathbf{x})$ to converge at all. It also says that although a huge number of samples will fall within the region \mathcal{R}_n , they will form a negligibly small fraction of the total number of samples.
- There are two common ways of obtaining regions that satisfy these conditions:
 - ① Shrink an initial region by specifying the volume V_n as some function of n such as $V_n = 1/\sqrt{n}$. Then, we need to show that $p_n(\mathbf{x})$ converges to $p(\mathbf{x})$. (This is like the Parzen window we'll talk about next.)

- $\lim_{n \rightarrow \infty} V_n = 0$ ensures that our space-averaged density will converge to $p(\mathbf{x})$.
- $\lim_{n \rightarrow \infty} k_n = \infty$ basically ensures that the frequency ratio will converge to the probability P (the binomial will be sufficiently peaked).
- $\lim_{n \rightarrow \infty} k_n/n = 0$ is required for $p_n(\mathbf{x})$ to converge at all. It also says that although a huge number of samples will fall within the region \mathcal{R}_n , they will form a negligibly small fraction of the total number of samples.
- There are two common ways of obtaining regions that satisfy these conditions:
 - ① Shrink an initial region by specifying the volume V_n as some function of n such as $V_n = 1/\sqrt{n}$. Then, we need to show that $p_n(\mathbf{x})$ converges to $p(\mathbf{x})$. (This is like the Parzen window we'll talk about next.)
 - ② Specify k_n as some function of n such as $k_n = \sqrt{n}$. Then, we grow the volume V_n until it encloses k_n neighbors of \mathbf{x} . (This is the k-nearest-neighbor).

- $\lim_{n \rightarrow \infty} V_n = 0$ ensures that our space-averaged density will converge to $p(\mathbf{x})$.
- $\lim_{n \rightarrow \infty} k_n = \infty$ basically ensures that the frequency ratio will converge to the probability P (the binomial will be sufficiently peaked).
- $\lim_{n \rightarrow \infty} k_n/n = 0$ is required for $p_n(\mathbf{x})$ to converge at all. It also says that although a huge number of samples will fall within the region \mathcal{R}_n , they will form a negligibly small fraction of the total number of samples.
- There are two common ways of obtaining regions that satisfy these conditions:
 - ① Shrink an initial region by specifying the volume V_n as some function of n such as $V_n = 1/\sqrt{n}$. Then, we need to show that $p_n(\mathbf{x})$ converges to $p(\mathbf{x})$. (This is like the Parzen window we'll talk about next.)
 - ② Specify k_n as some function of n such as $k_n = \sqrt{n}$. Then, we grow the volume V_n until it encloses k_n neighbors of \mathbf{x} . (This is the k-nearest-neighbor).

Both of these methods converge...



Parzen Windows

- Let's temporarily assume the region \mathcal{R} is a d -dimensional hypercube with h_n being the length of an edge.

Parzen Windows

- Let's temporarily assume the region \mathcal{R} is a d -dimensional hypercube with h_n being the length of an edge.
- The volume of the hypercube is given by

$$V_n = h_n^d . \quad (11)$$

Parzen Windows

- Let's temporarily assume the region \mathcal{R} is a d -dimensional hypercube with h_n being the length of an edge.
- The volume of the hypercube is given by

$$V_n = h_n^d . \quad (11)$$

- We can derive an analytic expression for k_n :
 - Define a **windowing function**:

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq 1/2 \\ 0 & \text{otherwise} \end{cases} \quad j = 1, \dots, d \quad (12)$$

- This windowing function φ defines a unit hypercube centered at the origin.
- Hence, $\varphi((\mathbf{x} - \mathbf{x}_i)/h_n)$ is equal to unity if \mathbf{x}_i falls within the hypercube of volume V_n centered at \mathbf{x} , and is zero otherwise.

- The number of samples in this hypercube is therefore given by

$$k_n = \sum_{i=1}^n \varphi \left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n} \right) . \quad (13)$$

- The number of samples in this hypercube is therefore given by

$$k_n = \sum_{i=1}^n \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) . \quad (13)$$

- Substituting in equation (7), $p_n(\mathbf{x}) = k_n/(nV_n)$ yields the estimate

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) . \quad (14)$$

- The number of samples in this hypercube is therefore given by

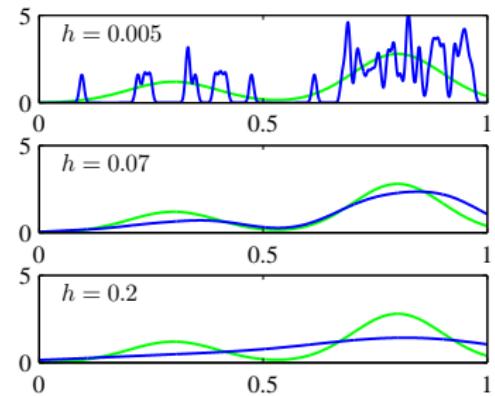
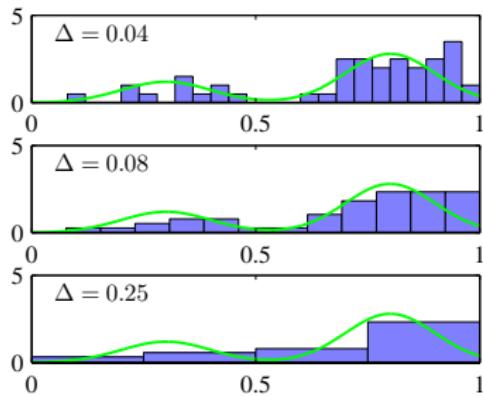
$$k_n = \sum_{i=1}^n \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) . \quad (13)$$

- Substituting in equation (7), $p_n(\mathbf{x}) = k_n/(nV_n)$ yields the estimate

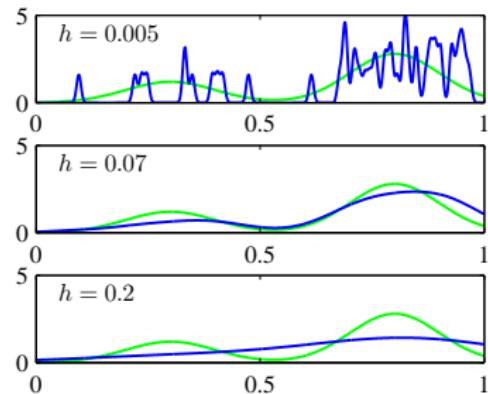
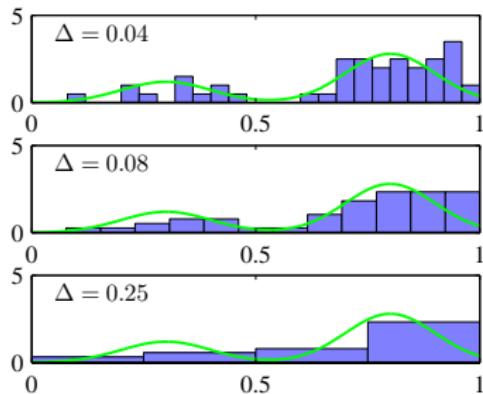
$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) . \quad (14)$$

- Hence, the windowing function φ , in this context called a **Parzen window**, tells us how to **weight** all of the samples in \mathcal{D} to determine $p(\mathbf{x})$ at a particular \mathbf{x} .

Example

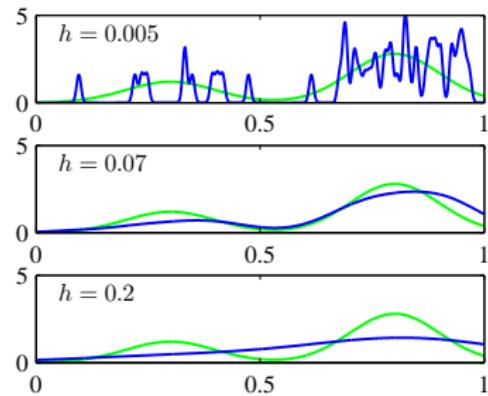
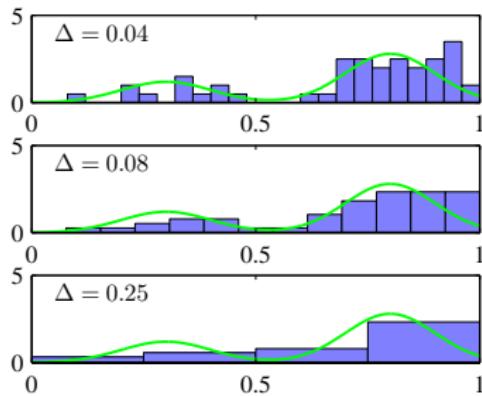


Example



- But, what undesirable trait from histograms are inherited by Parzen window density estimates of the form we've just defined?

Example



- But, what undesirable trait from histograms are inherited by Parzen window density estimates of the form we've just defined?
- Discontinuities...

Generalizing the Kernel Function

- What if we allow a more general class of windowing functions rather than the hypercube?
- If we think of the windowing function as an interpolator, rather than considering the window function about \mathbf{x} only, we can visualize it as a kernel sitting on each data sample \mathbf{x}_i in \mathcal{D} .

Generalizing the Kernel Function

- What if we allow a more general class of windowing functions rather than the hypercube?
- If we think of the windowing function as an interpolator, rather than considering the window function about \mathbf{x} only, we can visualize it as a kernel sitting on each data sample \mathbf{x}_i in \mathcal{D} .
- And, if we require the following two conditions on the kernel function φ , then we can be assured that the resulting density $p_n(\mathbf{x})$ will be proper: non-negative and integrate to 1.

$$\varphi(\mathbf{x}) \geq 0 \quad (15)$$

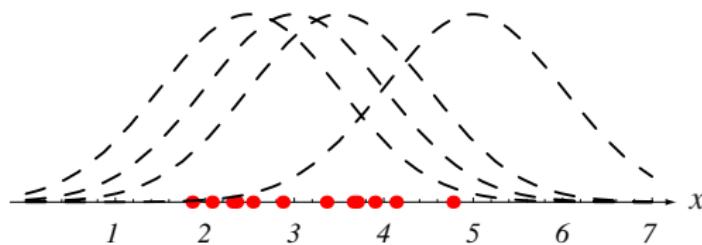
$$\int \varphi(\mathbf{u}) d\mathbf{u} = 1 \quad (16)$$

- For our previous case of $V_n = h_n^d$, then it follows $p_n(\mathbf{x})$ will also satisfy these conditions.

Example: A Univariate Gaussian Kernel

- A popular choice of the kernel is the Gaussian kernel:

$$\varphi_h(u) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}u^2\right] \quad (17)$$



- The resulting density is given by:

$$p(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_n \sqrt{2\pi}} \exp\left[-\frac{1}{2h_n^2}(x - x_i)^2\right] \quad (18)$$

- It will give us smoother estimates without the discontinuities from the hypercube kernel.

Effect of the Window Width

Slide 1

- An important question is what effect does the window width h_n have on $p_n(\mathbf{x})$?
- Define $\delta_n(\mathbf{x})$ as

$$\delta_n(\mathbf{x}) = \frac{1}{V_n} \varphi\left(\frac{\mathbf{x}}{h_n}\right) \quad (19)$$

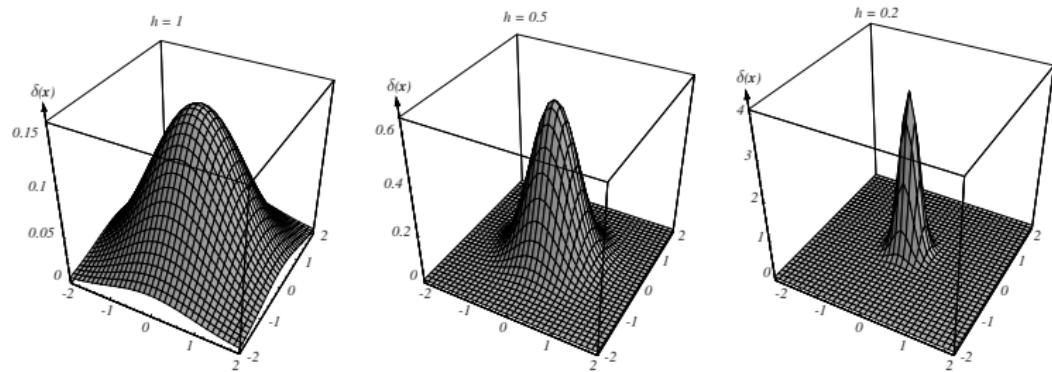
and rewrite $p_n(\mathbf{x})$ as the average

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \delta_n(\mathbf{x} - \mathbf{x}_i) \quad (20)$$

Effect of the Window Width

Slide II

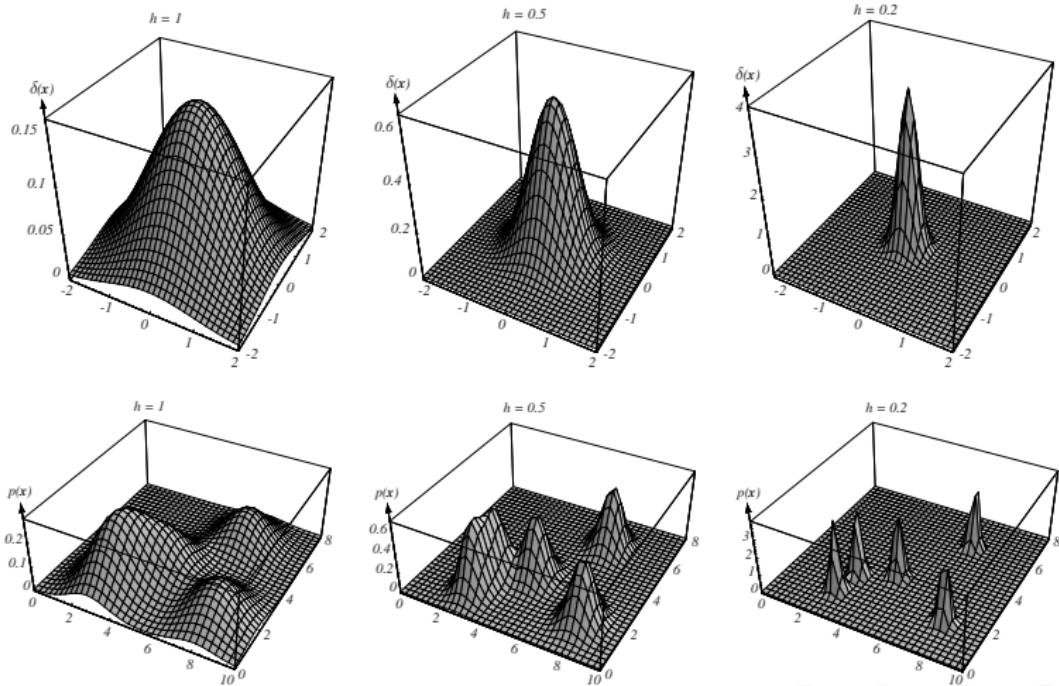
- h_n clearly affects both the amplitude and the width of $\delta_n(x)$.



Effect of the Window Width

Slide II

- h_n clearly affects both the amplitude and the width of $\delta_n(x)$.



Effect of Window Width (And, hence, Volume V_n)

- But, for any value of h_n , the distribution is normalized:

$$\int \delta(\mathbf{x} - \mathbf{x}_i) d\mathbf{x} = \int \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) d\mathbf{x} = \int \varphi(\mathbf{u}) d\mathbf{u} = 1 \quad (21)$$

Effect of Window Width (And, hence, Volume V_n)

- But, for any value of h_n , the distribution is normalized:

$$\int \delta(\mathbf{x} - \mathbf{x}_i) d\mathbf{x} = \int \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) d\mathbf{x} = \int \varphi(\mathbf{u}) d\mathbf{u} = 1 \quad (21)$$

- If V_n is too large, the estimate will suffer from too little resolution.

Effect of Window Width (And, hence, Volume V_n)

- But, for any value of h_n , the distribution is normalized:

$$\int \delta(\mathbf{x} - \mathbf{x}_i) d\mathbf{x} = \int \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) d\mathbf{x} = \int \varphi(\mathbf{u}) d\mathbf{u} = 1 \quad (21)$$

- If V_n is too large, the estimate will suffer from too little resolution.
- If V_n is too small, the estimate will suffer from too much variability.

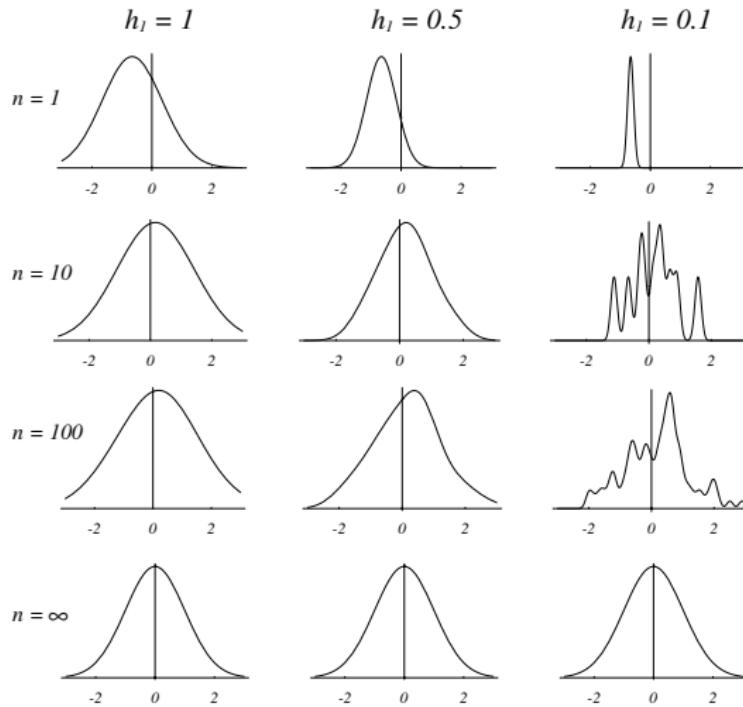
Effect of Window Width (And, hence, Volume V_n)

- But, for any value of h_n , the distribution is normalized:

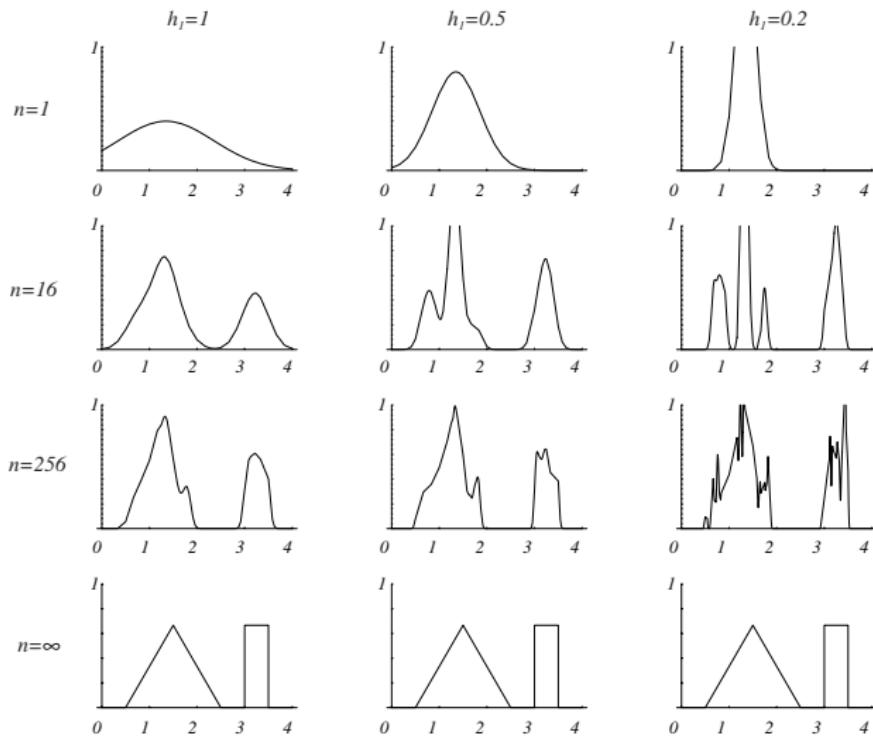
$$\int \delta(\mathbf{x} - \mathbf{x}_i) d\mathbf{x} = \int \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) d\mathbf{x} = \int \varphi(\mathbf{u}) d\mathbf{u} = 1 \quad (21)$$

- If V_n is too large, the estimate will suffer from too little resolution.
- If V_n is too small, the estimate will suffer from too much variability.
- In theory (with an unlimited number of samples), we can let V_n slowly approach zero as n increases and then $p_n(\mathbf{x})$ will converge to the unknown $p(\mathbf{x})$. But, in practice, we can, at best, seek some compromise.

Example: Revisiting the Univariate Gaussian Kernel



Example: A Bimodal Distribution

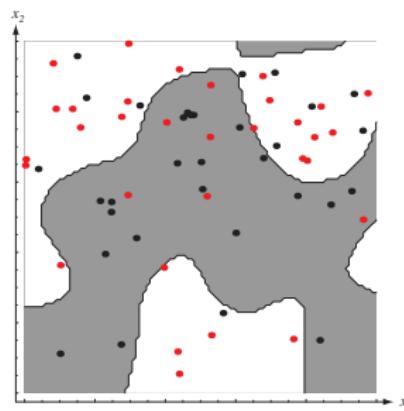
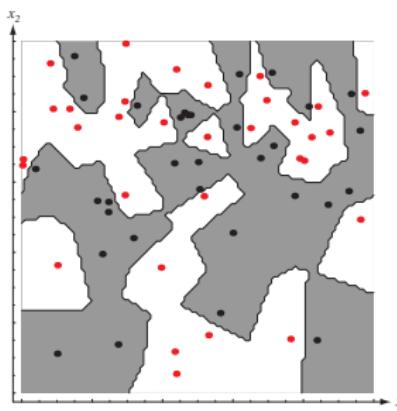


Parzen Window-Based Classifiers

- Estimate the densities for each category.
- Classify a query point by the label corresponding to the maximum posterior (i.e., one can include priors).

Parzen Window-Based Classifiers

- Estimate the densities for each category.
- Classify a query point by the label corresponding to the maximum posterior (i.e., one can include priors).
- As you guessed it, the **decision regions for a Parzen window-based classifier depend upon the kernel function**.



Parzen Window-Based Classifiers

- During training, we can make the error arbitrarily low by making the window sufficiently small, but this will have an ill-effect during testing (which is our ultimate need).
- Think of any possibilities for system rules of choosing the kernel?

Parzen Window-Based Classifiers

- During training, we can make the error arbitrarily low by making the window sufficiently small, but this will have an ill-effect during testing (which is our ultimate need).
- Think of any possibilities for system rules of choosing the kernel?
- One possibility is to use cross-validation. Break up the data into a training set and a validation set. Then, perform training on the training set with varying bandwidths. Select the bandwidth that minimizes the error on the validation set.

Parzen Window-Based Classifiers

- During training, we can make the error arbitrarily low by making the window sufficiently small, but this will have an ill-effect during testing (which is our ultimate need).
- Think of any possibilities for system rules of choosing the kernel?
- One possibility is to use cross-validation. Break up the data into a training set and a validation set. Then, perform training on the training set with varying bandwidths. Select the bandwidth that minimizes the error on the validation set.
- There is little theoretical justification for choosing one window width over another.

k_n Nearest Neighbor Methods

- Selecting the best window / bandwidth is a severe limiting factor for Parzen window estimators.
- k_n -NN methods circumvent this problem by making the window size a function of the actual training data.

k_n Nearest Neighbor Methods

- Selecting the best window / bandwidth is a severe limiting factor for Parzen window estimators.
- k_n -NN methods circumvent this problem by making the window size a function of the actual training data.
- The basic idea here is to center our window around \mathbf{x} and let it grow until it capture k_n samples, where k_n is a function of n .
 - These samples are the k_n nearest neighbors of \mathbf{x} .
 - If the density is high near \mathbf{x} then the window will be relatively small leading to good resolution.
 - If the density is low near \mathbf{x} , the window will grow large, but it will stop soon after it enters regions of higher density.

k_n Nearest Neighbor Methods

- Selecting the best window / bandwidth is a severe limiting factor for Parzen window estimators.
- k_n -NN methods circumvent this problem by making the window size a function of the actual training data.
- The basic idea here is to center our window around \mathbf{x} and let it grow until it capture k_n samples, where k_n is a function of n .
 - These samples are the k_n nearest neighbors of \mathbf{x} .
 - If the density is high near \mathbf{x} then the window will be relatively small leading to good resolution.
 - If the density is low near \mathbf{x} , the window will grow large, but it will stop soon after it enters regions of higher density.
 - In either case, we estimate $p_n(\mathbf{x})$ according to

$$p_n(\mathbf{x}) = \frac{k_n}{nV_n} \quad (22)$$

$$p_n(\mathbf{x}) = \frac{k_n}{nV_n}$$

- We want k_n to go to infinity as n goes to infinity thereby assuring us that k_n/n will be a good estimate of the probability that a point will fall in the window of volume V_n .

$$p_n(\mathbf{x}) = \frac{k_n}{nV_n}$$

- We want k_n to go to infinity as n goes to infinity thereby assuring us that k_n/n will be a good estimate of the probability that a point will fall in the window of volume V_n .
- But, we also want k_n to grow sufficiently slowly so that the size of our window will go to zero.

$$p_n(\mathbf{x}) = \frac{k_n}{nV_n}$$

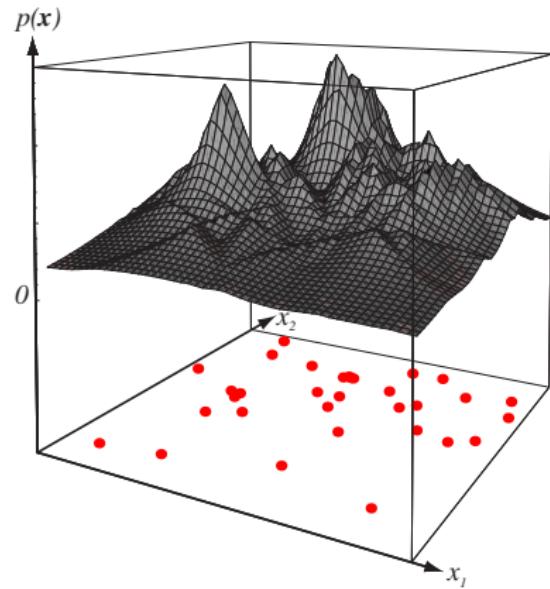
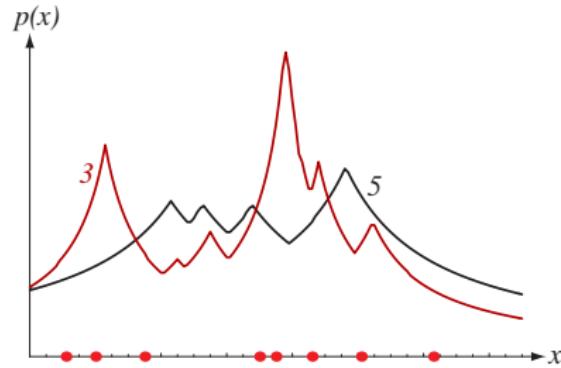
- We want k_n to go to infinity as n goes to infinity thereby assuring us that k_n/n will be a good estimate of the probability that a point will fall in the window of volume V_n .
- But, we also want k_n to grow sufficiently slowly so that the size of our window will go to zero.
- Thus, we want k_n/n to go to zero.

$$p_n(\mathbf{x}) = \frac{k_n}{nV_n}$$

- We want k_n to go to infinity as n goes to infinity thereby assuring us that k_n/n will be a good estimate of the probability that a point will fall in the window of volume V_n .
- But, we also want k_n to grow sufficiently slowly so that the size of our window will go to zero.
- Thus, we want k_n/n to go to zero.
- Recall these conditions from the earlier discussion; these will ensure that $p_n(\mathbf{x})$ converges to $p(\mathbf{x})$ as n approaches infinity.

Examples of k_n -NN Estimation

- Notice the discontinuities in the slopes of the estimate.

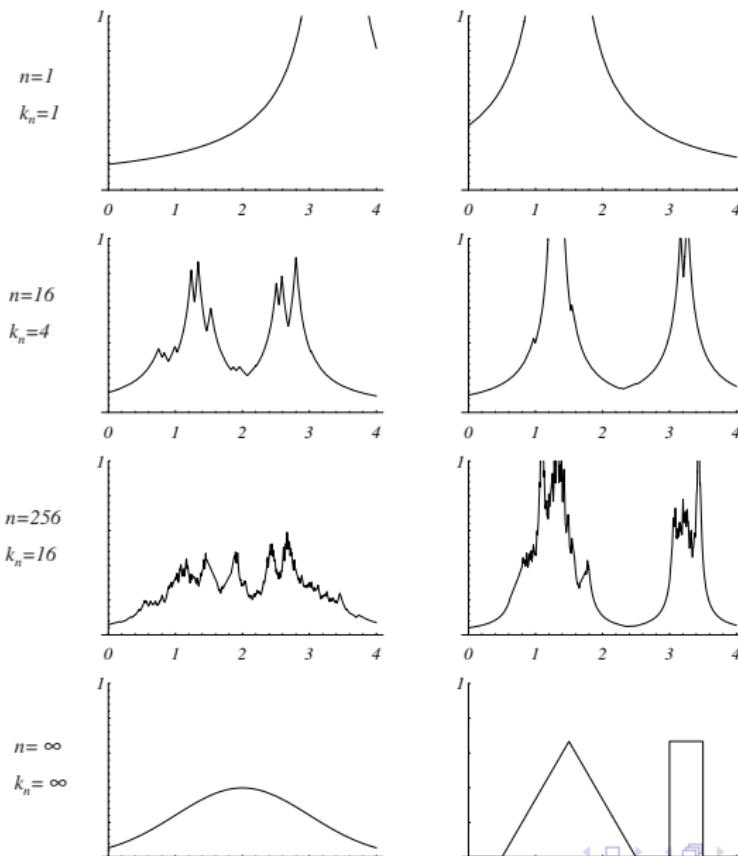


k -NN Estimation From 1 Sample

- We don't expect the density estimate from 1 sample to be very good, but in the case of k -NN it will diverge!
- With $n = 1$ and $k_n = \sqrt{n} = 1$, the estimate for $p_n(x)$ is

$$p_n(x) = \frac{1}{2|x - x_1|} \quad (23)$$

But, as we increase the number of samples, the estimate will improve.



Limitations

- The k_n -NN Estimator suffers from an analogous flaw from which the Parzen window methods suffer. What is it?

Limitations

- The k_n -NN Estimator suffers from an analogous flaw from which the Parzen window methods suffer. What is it?
- How do we specify the k_n ?
- We saw earlier that the specification of k_n can lead to radically different density estimates (in practical situations where the number of training samples is limited).

Limitations

- The k_n -NN Estimator suffers from an analogous flaw from which the Parzen window methods suffer. What is it?
- How do we specify the k_n ?
- We saw earlier that the specification of k_n can lead to radically different density estimates (in practical situations where the number of training samples is limited).
- One could obtain a sequence of estimates by taking $k_n = k_1\sqrt{n}$ and choose different values of k_1 .

Limitations

- The k_n -NN Estimator suffers from an analogous flaw from which the Parzen window methods suffer. What is it?
- How do we specify the k_n ?
- We saw earlier that the specification of k_n can lead to radically different density estimates (in practical situations where the number of training samples is limited).
- One could obtain a sequence of estimates by taking $k_n = k_1\sqrt{n}$ and choose different values of k_1 .
- But, like the Parzen window size, one choice is as good as another absent any additional information.

Limitations

- The k_n -NN Estimator suffers from an analogous flaw from which the Parzen window methods suffer. What is it?
- How do we specify the k_n ?
- We saw earlier that the specification of k_n can lead to radically different density estimates (in practical situations where the number of training samples is limited).
- One could obtain a sequence of estimates by taking $k_n = k_1\sqrt{n}$ and choose different values of k_1 .
- But, like the Parzen window size, one choice is as good as another absent any additional information.
- Similarly, in classification scenarios, we can base our judgement on classification error.

k -NN Posterior Estimation for Classification

- We can directly apply the k -NN methods to estimate the posterior probabilities $P(\omega_i|\mathbf{x})$ from a set of n labeled samples.

k -NN Posterior Estimation for Classification

- We can directly apply the k -NN methods to estimate the posterior probabilities $P(\omega_i|\mathbf{x})$ from a set of n labeled samples.
- Place a window of volume V around \mathbf{x} and capture k samples, with k_i turning out to be of label ω_i .

k -NN Posterior Estimation for Classification

- We can directly apply the k -NN methods to estimate the posterior probabilities $P(\omega_i|\mathbf{x})$ from a set of n labeled samples.
- Place a window of volume V around \mathbf{x} and capture k samples, with k_i turning out to be of label ω_i .
- The estimate for the joint probability is thus

$$p_n(\mathbf{x}, \omega_i) = \frac{k_i}{nV} \quad (24)$$

k -NN Posterior Estimation for Classification

- We can directly apply the k -NN methods to estimate the posterior probabilities $P(\omega_i|\mathbf{x})$ from a set of n labeled samples.
- Place a window of volume V around \mathbf{x} and capture k samples, with k_i turning out to be of label ω_i .
- The estimate for the joint probability is thus

$$p_n(\mathbf{x}, \omega_i) = \frac{k_i}{nV} \quad (24)$$

- A reasonable estimate for the posterior is thus

$$P_n(\omega_i|\mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_c p_n(\mathbf{x}, \omega_c)} = \frac{k_i}{k} \quad (25)$$

k -NN Posterior Estimation for Classification

- We can directly apply the k -NN methods to estimate the posterior probabilities $P(\omega_i|\mathbf{x})$ from a set of n labeled samples.
- Place a window of volume V around \mathbf{x} and capture k samples, with k_i turning out to be of label ω_i .
- The estimate for the joint probability is thus

$$p_n(\mathbf{x}, \omega_i) = \frac{k_i}{nV} \quad (24)$$

- A reasonable estimate for the posterior is thus

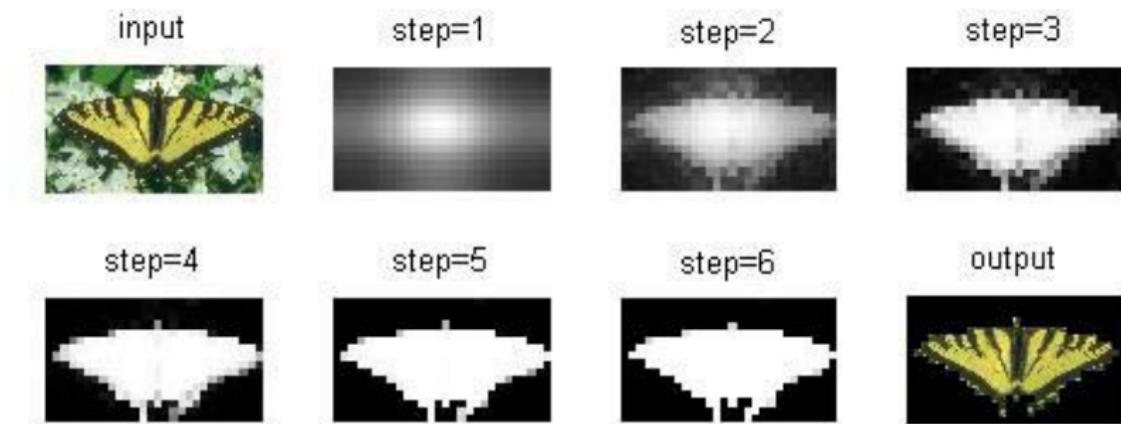
$$P_n(\omega_i|\mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_c p_n(\mathbf{x}, \omega_c)} = \frac{k_i}{k} \quad (25)$$

- Hence, the posterior probability for ω_i is simply the fraction of samples within the window that are labeled ω_i . This is a simple and intuitive result.

Example: Figure-Ground Discrimination

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- Figure-ground discrimination is an important low-level vision task.
- Want to separate the pixels that contain some foreground object (specified in some meaningful way) from the background.



Example: Figure-Ground Discrimination

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- This paper presents a method for figure-ground discrimination based on non-parametric densities for the foreground and background.
- They use a subset of the pixels from each of the two regions.
- They propose an algorithm called **iterative sampling-expectation** for performing the actual segmentation.
- The required input is simply a region of interest (mostly) containing the object.

Example: Figure-Ground Discrimination

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- Given a set of n samples $S = \{\mathbf{x}_i\}$ where each \mathbf{x}_i is a d -dimensional vector.
- We know the kernel density estimate is defined as

$$\hat{p}(\mathbf{y}) = \frac{1}{n\sigma_1 \dots \sigma_d} \sum_{i=1}^n \prod_{j=1}^d \varphi\left(\frac{\mathbf{y}_j - \mathbf{x}_{ij}}{\sigma_j}\right) \quad (26)$$

where the same kernel φ with different bandwidth σ_j is used in each dimension.

The Representation

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- The representation used here is a function of RGB:

$$r = R/(R + G + B) \quad (27)$$

$$g = G/(R + G + B) \quad (28)$$

$$s = (R + G + B)/3 \quad (29)$$

- Separating the chromaticity from the brightness allows them to use a wider bandwidth in the brightness dimension to account for variability due to shading effects.
- And, much narrower kernels can be used on the r and g chromaticity channels to enable better discrimination.

The Color Density

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- Given a sample of pixels $S = \{\mathbf{x}_i = (r_i, g_i, s_i)\}$, the color density estimate is given by

$$\hat{P}(\mathbf{x} = (r, g, s)) = \frac{1}{n} \sum_{i=1}^n K_{\sigma_r}(r - r_i) K_{\sigma_g}(g - g_i) K_{\sigma_s}(s - s_i) \quad (30)$$

where we have simplified the kernel definition:

$$K_{\sigma}(t) = \frac{1}{\sigma} \varphi\left(\frac{t}{\sigma}\right) \quad (31)$$

- They use Gaussian kernels

$$K_{\sigma}(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2\right] \quad (32)$$

with a different bandwidth in each dimension.

Data-Driven Bandwidth

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- The bandwidth for each channel is calculated directly from the image based on sample statistics.

$$\sigma \approx 1.06\hat{\sigma}n^{-1/5} \quad (33)$$

where $\hat{\sigma}^2$ is the sample variance.

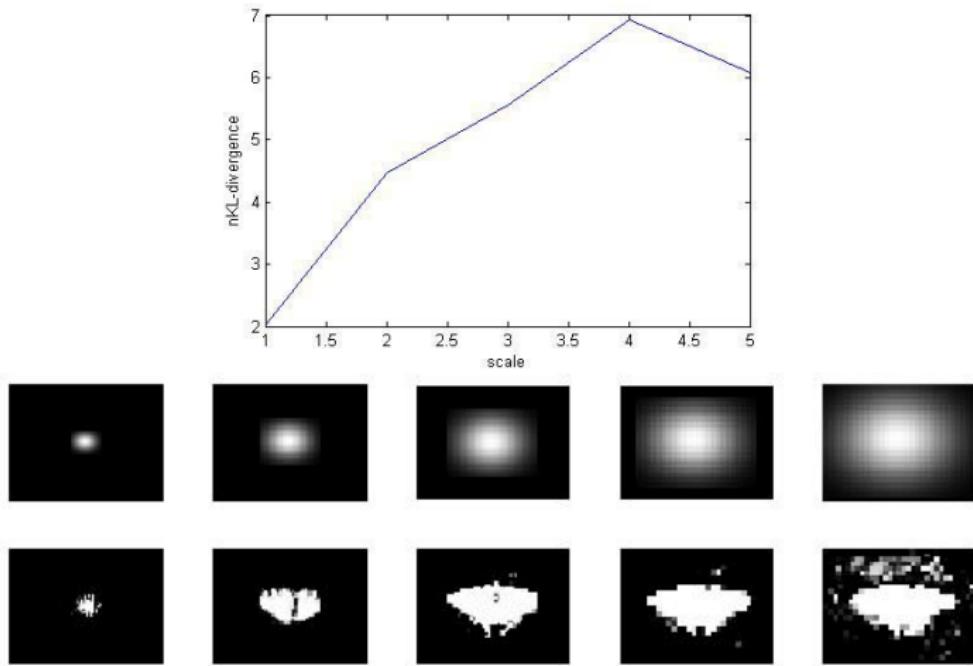
Initialization: Choosing the Initial Scale

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- For initialization, they compute a distance between the foreground and background distribution by varying the scale of a single Gaussian kernel (on the foreground).
- To evaluate the “significance” of a particular scale, they compute the normalized KL-divergence:

$$\text{nKL}(\hat{P}_{fg} \parallel \hat{P}_{bg}) = \frac{-\sum_{i=1}^n \hat{P}_{fg}(\mathbf{x}_i) \log \frac{\hat{P}_{fg}(\mathbf{x}_i)}{\hat{P}_{bg}(\mathbf{x}_i)}}{\sum_{i=1}^n \hat{P}_{fg}(\mathbf{x}_i)} \quad (34)$$

where \hat{P}_{fg} and \hat{P}_{bg} are the density estimates for the foreground and background regions respectively. To compute each, they use about 6% of the pixels (using all of the pixels would lead to quite slow performance).



Iterative Sampling-Expectation Algorithm

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- Given the initial segmentation, they need to refine the models **and** labels to adapt better to the image.
- However, this is a chicken-and-egg problem. If we know the labels, we could compute the models, and if we knew the models, we could compute the best labels.

Iterative Sampling-Expectation Algorithm

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.

- Given the initial segmentation, they need to refine the models **and** labels to adapt better to the image.
- However, this is a chicken-and-egg problem. If we know the labels, we could compute the models, and if we knew the models, we could compute the best labels.
- They propose an EM algorithm for this. The basic idea is to alternate between estimating the probability that each pixel is of the two classes, and then given this probability to refine the underlying models.
- EM is guaranteed to converge (but only to a local minimum).

- ① Initialize using the normalized KL-divergence.

- ① Initialize using the normalized KL-divergence.
- ② Uniformly sample a set of pixel from the image to use in the kernel density estimation. This is essentially the 'M' step (because we have a non-parametric density).

- ① Initialize using the normalized KL-divergence.
- ② Uniformly sample a set of pixel from the image to use in the kernel density estimation. This is essentially the 'M' step (because we have a non-parametric density).
- ③ Update the pixel assignment based on maximum likelihood (the 'E' step).

- ① Initialize using the normalized KL-divergence.
- ② Uniformly sample a set of pixel from the image to use in the kernel density estimation. This is essentially the 'M' step (because we have a non-parametric density).
- ③ Update the pixel assignment based on maximum likelihood (the 'E' step).
- ④ Repeat until stable.

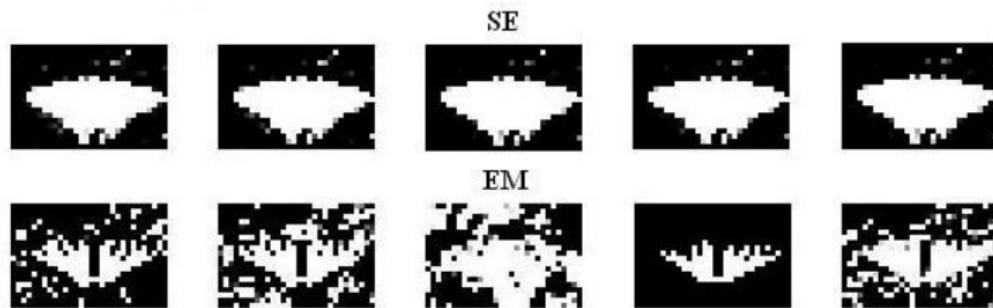
- ① Initialize using the normalized KL-divergence.
- ② Uniformly sample a set of pixel from the image to use in the kernel density estimation. This is essentially the 'M' step (because we have a non-parametric density).
- ③ Update the pixel assignment based on maximum likelihood (the 'E' step).
- ④ Repeat until stable.
 - One can use a hard assignment of the pixels and the kernel density estimator we've discussed, or a soft assignment of the pixels and then a weighted kernel density estimate (the weight is between the different classes).

- ① Initialize using the normalized KL-divergence.
 - ② Uniformly sample a set of pixel from the image to use in the kernel density estimation. This is essentially the 'M' step (because we have a non-parametric density).
 - ③ Update the pixel assignment based on maximum likelihood (the 'E' step).
 - ④ Repeat until stable.
- One can use a hard assignment of the pixels and the kernel density estimator we've discussed, or a soft assignment of the pixels and then a weighted kernel density estimate (the weight is between the different classes).
 - The overall probability of a pixel belonging to the foreground class

$$\hat{P}_{fg}(\mathbf{y}) = \frac{1}{Z} \sum_{i=1}^n \hat{P}_{fg}(\mathbf{x}_i) \prod_{j=1}^d K\left(\frac{y_j - x_{ij}}{\sigma_j}\right) \quad (35)$$

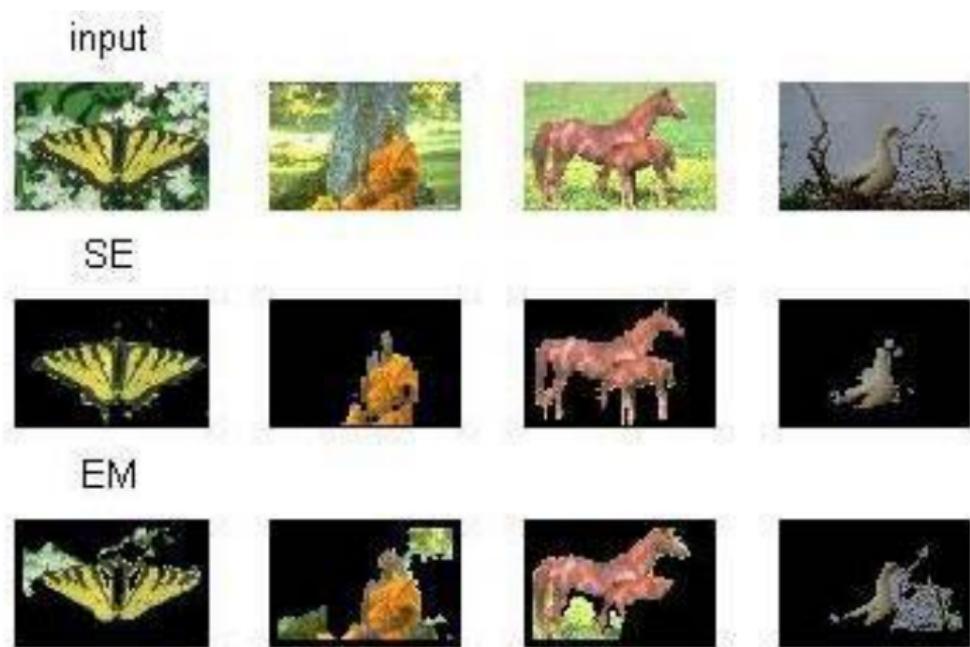
Results: Stability

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.



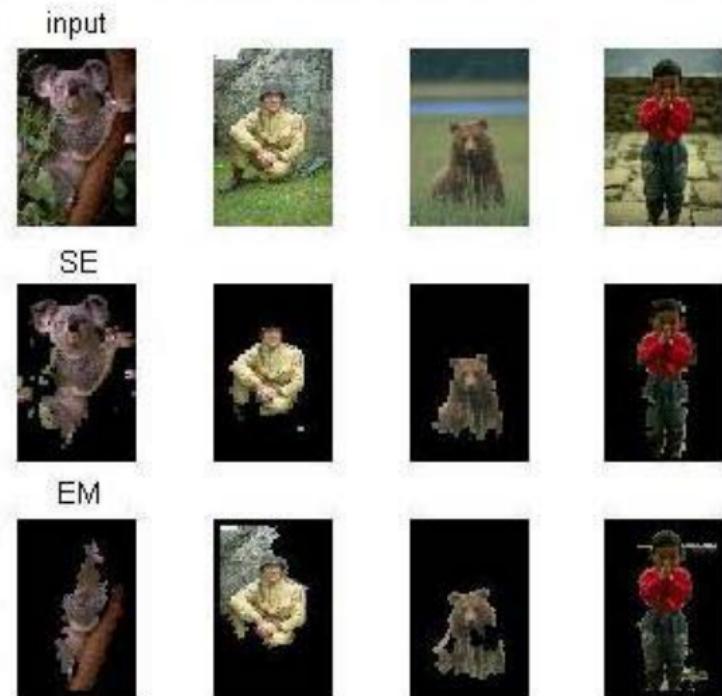
Results

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.



Results

Source: Zhao and Davis. Iterative Figure-Ground Discrimination. ICPR 2004.



Regression Analysis

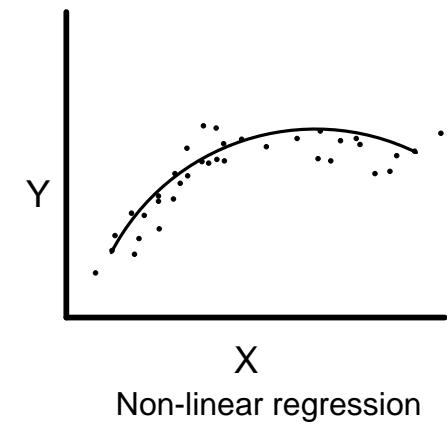
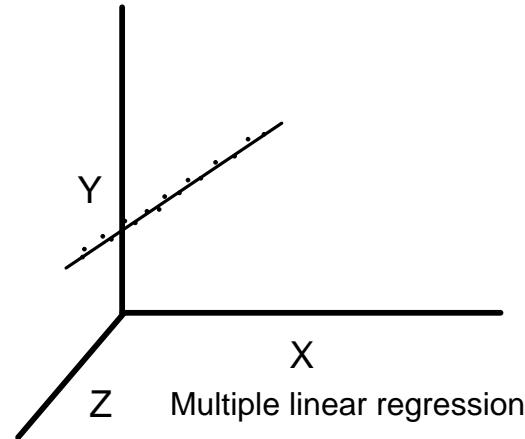
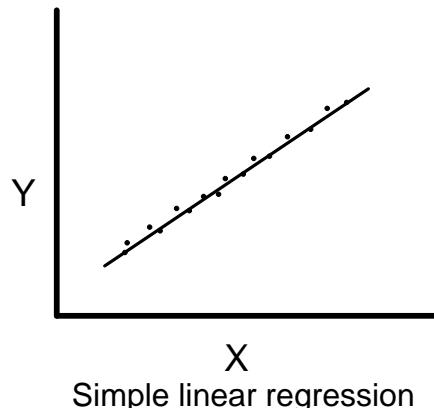
This presentation includes...

- Regression Analysis
 - Simple Linear Regression
 - Multiple Linear Regression
 - Non-Linear Regression Analysis
- Auto-Regression Analysis

Regression Analysis

Regression Analysis

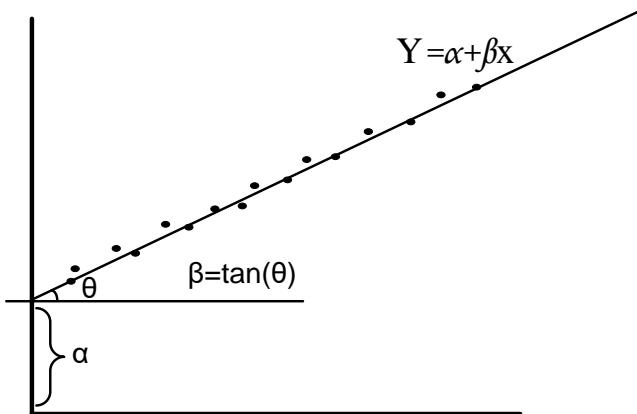
- The regression analysis is a statistical method to deal with the formulation of mathematical model depicting relationship amongst variables, which can be used for the purpose of prediction of the values of dependent variable, given the values of independent variables.
- Classification of Regression Analysis Models**
 - Linear regression models
 - Simple linear regression
 - Multiple linear regression
 - Non-linear regression models



Simple Linear Regression Model

In simple linear regression, we have only two variables:

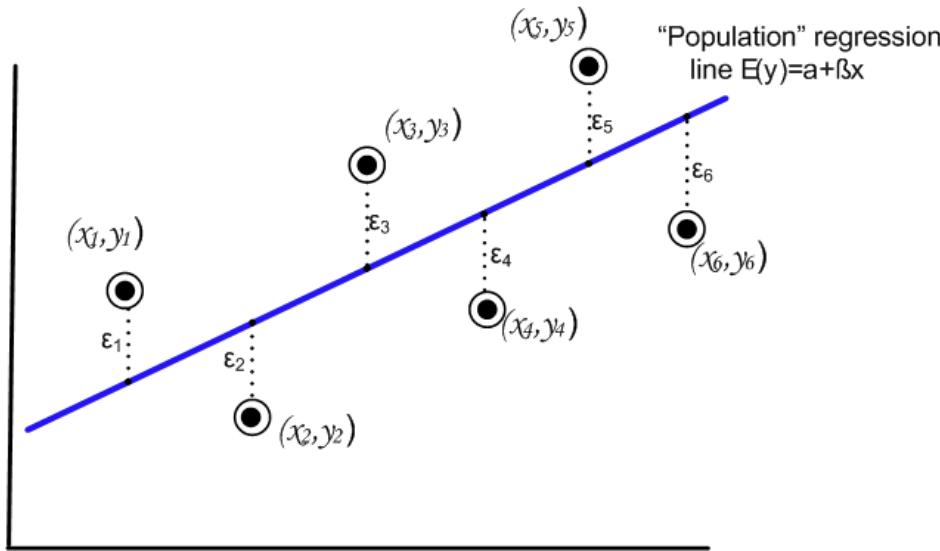
- Dependent variable (also called **Response**), usually denoted as Y .
- Independent variable (alternatively called **Regressor**), usually denoted as x .
- A reasonable form of a relationship between the Response Y and the Regressor x is the linear relationship, that is in the form $Y = \alpha + \beta x$



Note:

- There are infinite number of lines (and hence α_s and β_s)
- The concept of regression analysis deal with finding the best relationship between Y and x (and hence best fitted values of α and β) quantifying the strength of that relationship.

Regression Analysis



Given the set $[(x_i, y_i), i = 1, 2, \dots, n]$ of data involving n pairs of (x, y) values, our objective is to find “true” or population regression line such that $Y = \alpha + \beta x + \epsilon$

Here, ϵ is a random variable with $E(\epsilon) = 0$ and $var(\epsilon) = \sigma^2$. The quantity σ^2 is often called the **error variance**.

Note:

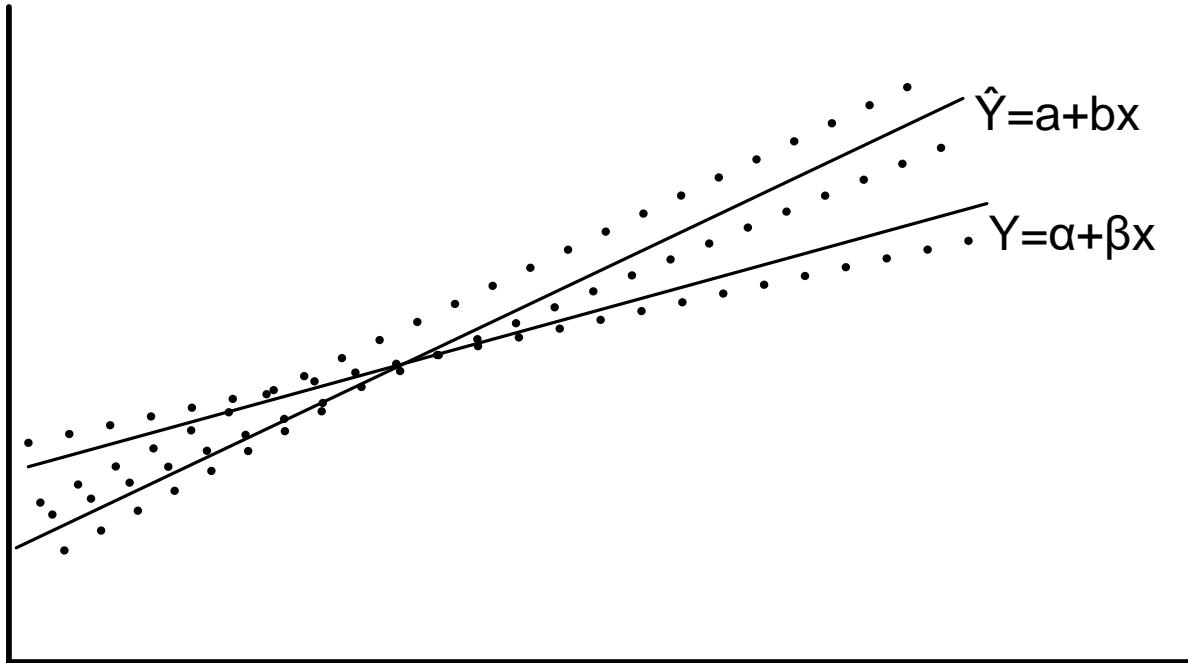
- $E(\epsilon) = 0$ implies that at a specific x , the y values are distributed around the “true” regression line $Y = \alpha + \beta x$ (i.e., the positive and negative errors around the true line is reasonable).
- α and β are called **regression coefficients**.
- α and β values are to be estimated from the data.

True versus Fitted Regression Line

- The task in regression analysis is to estimate the regression coefficients α and β .
- Suppose, we denote the estimates a for α and b for β . Then the fitted regression line is

$$\hat{Y} = a + bx$$

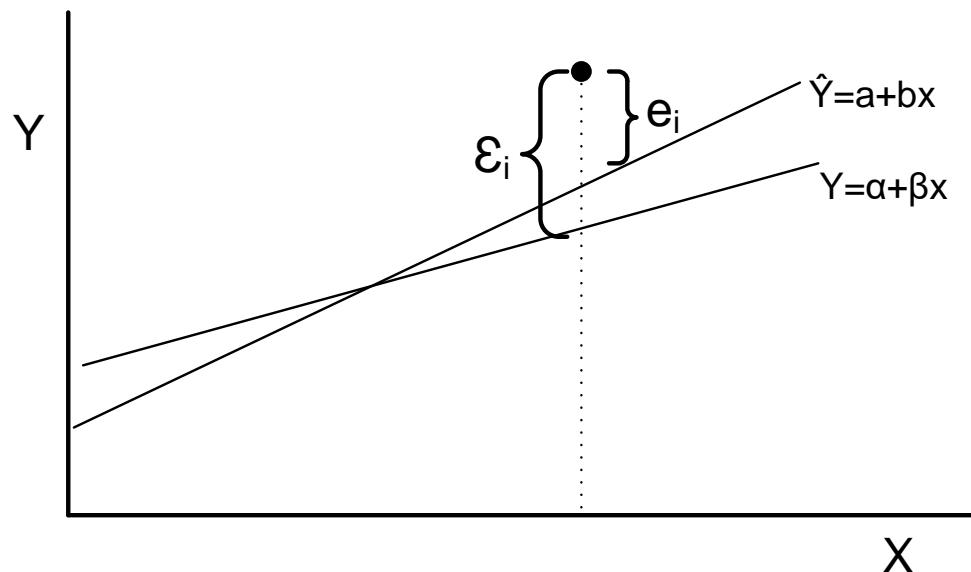
where \hat{Y} is the predicted or fitted value.



Least Square Method to estimate α and β

This method uses the concept of **residual**. A residual is essentially an error in the fit of the model $\hat{Y} = a + bx$. Thus, i^{th} residual is

$$e_i = Y_i - \hat{Y}_i, i = 1, 2, 3, \dots, n$$



Least Square method

- The residual sum of squares is often called **the sum of squares of the errors** about the fitted line and is denoted as SSE

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - a - bx_i)^2$$

- We are to minimize the value of SSE and hence to determine the parameters of a and b .
- Differentiating SSE with respect to a and b , we have

$$\frac{\partial(SSE)}{\partial a} = -2 \sum_{i=1}^n (y_i - a - bx_i)$$

$$\frac{\partial(SSE)}{\partial b} = -2 \sum_{i=1}^n (y_i - a - bx_i) \cdot x_i$$

For minimum value of SSE, $\frac{\partial(SSE)}{\partial a} = 0$

$$\frac{\partial(SSE)}{\partial b} = 0$$

Least Square method to estimate α and β

Thus we set

$$na + b \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

These two equations can be solved to determine the values of a and b , and it can be calculated that

$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$a = \bar{y} - b\bar{x}$$

R^2 : Measure of Quality of Fit

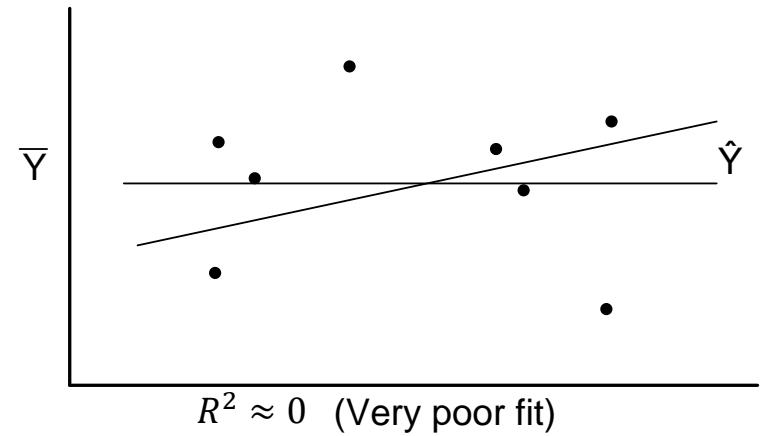
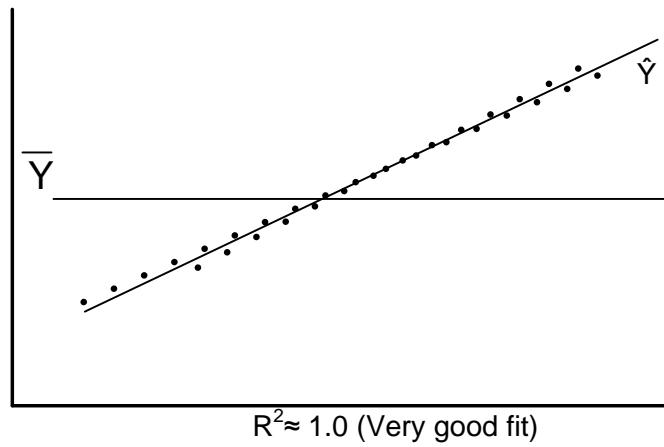
- A quantity R^2 , is called **coefficient of determination** is used to measure the proportion of variability of the fitted model.
- We have $SSE = \sum_{i=1}^n (y_i - \hat{y})^2$
- It signifies the **variability due to error**.
- Now, let us define the **total corrected sum of squares**, defined as
$$SST = \sum_{i=1}^n (y_i - \bar{y})^2$$
- SST represents the variation in the response values. The R^2 is

$$R^2 = 1 - \frac{SSE}{SST}$$

Note:

- If fit is perfect, all residuals are zero and thus $R^2 = 1.0$ (very good fit)
- If SSE is only slightly smaller than SST, then $R^2 \approx 0$ (very poor fit)

R^2 : Measure of Quality of Fit



Multiple Linear Regression

- When more than one variable are independent variable, then the regression can be estimated as a **multiple regression model**
- When this model is linear in coefficients, it is called **multiple linear regression model**
- If k -independent variables $x_1, x_2, x_3, \dots, x_k$ are associated, the multiple linear regression model is given by

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_k x_k + \epsilon$$

- And the estimated response is obtained as

$$\hat{y}_i = b_0 + b_1 x_1 + b_2 x_2 + b_k x_k$$

Multiple Linear Regression

Estimating the coefficients

Let the data points given to us is

$$(x_{1i}, x_{2i}, x_{3i}, \dots, \dots, \dots, x_{ki}, y_i) \quad i = 1, 2, \dots, n, \quad n > k$$

where y_i is the observed response to the values $x_{1i}, x_{2i}, x_{3i}, \dots, \dots, \dots, x_{ki}$ of k independent variables $x_1, x_2, x_3, \dots, \dots, \dots, x_k$.

Thus,

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_k x_{ki} + \epsilon_i \\ \text{and} \quad \hat{y}_i &= b_0 + b_1 x_{1i} + b_2 x_{2i} + b_k x_{ki} + e_i \end{aligned}$$

where ϵ_i and e_i are the random error and residual error, respectively associated with true response y_i and fitted response \hat{y}_i .

Using the concept of **Least Square Method** to estimate $b_0, b_1, b_2, \dots, b_k$, we minimize the expression

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Multiple Linear Regression

- Differentiating SSE in turn with respect to $b_0, b_1, b_2, \dots, b_k$ and equating to zero, we generate the set of $(k+1)$ normal estimation equations for multiple linear regression.

$$nb_0 + b_1 \sum_{i=1}^n x_{1i} + b_2 \sum_{i=1}^n x_{2i} + \dots + b_k \sum_{i=1}^n x_{ki} = \sum_{i=1}^n y_i$$

$$b_0 \sum_{i=1}^n x_{1i} + b_1 \sum_{i=1}^n x_{1i}^2 + b_2 \sum_{i=1}^n x_{1i} \cdot x_{2i} + \dots + b_k \sum_{i=1}^n x_{1i} \cdot x_{ki} = \sum_{i=1}^n x_i \cdot y_i$$

...

...

$$b_0 \sum_{i=1}^n x_{ki} + b_1 \sum_{i=1}^n x_{ki} \cdot x_{1i} + b_2 \sum_{i=1}^n x_{ki} \cdot x_{2i} + \dots + b_k \sum_{i=1}^n x_{ki}^2 = \sum_{i=1}^n x_i \cdot y_i$$

- The system of linear equations can be solved for b_0, b_1, \dots, b_k by any appropriate method for solving system of linear equations.
- Hence, the multiple linear regression model can be built.

Non Linear Regression Model

- When the regression equation is in terms of r -degree, $r>1$, then it is called nonlinear regression model. When more than one independent variables are there, then it is called Multiple Non linear Regression model. Also, alternatively termed as polynomial regression model. In general, it takes the form

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_r x^r + \epsilon$$

- The estimated response is obtained as

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \dots + b_r x^r$$

Solving for Polynomial Regression Model

Given that $(x_i, y_i); i = 1, 2, \dots, n$ are n pairs of observations. Each observations would satisfy the equations:

$$y_i = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_r x^r + \epsilon_i$$

and $\hat{y}_i = b_0 + b_1 x + b_2 x^2 + \dots + b_r x^r + e_i$

where, r is the degree of polynomial

ϵ_i = is the i^{th} random error

e_i = is the i^{th} residual error

Note: The number of observations, n , must be at least as large as $r+1$, the number of parameters to be estimated.

The polynomial model can be transformed into a general linear regression model setting $x_1 = x, x_2 = x^2, \dots, x_n = x^r$. Thus, the equation assumes the form:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_r x^r + \epsilon_i$$

$$\hat{y}_i = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_r x_r + e_i$$

This model then can be solved using the procedure followed for multiple linear regression model.

Auto-Regression Analysis

Auto Regression Analysis

- Regression analysis for time-ordered data is known as Auto-Regression Analysis
- Time series data are data collected on the same observational unit at multiple time periods

Example: Indian rate of price inflation

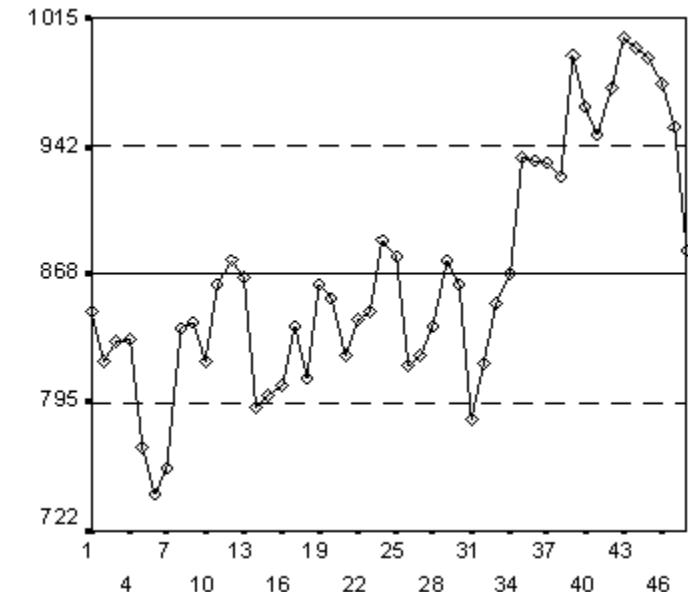
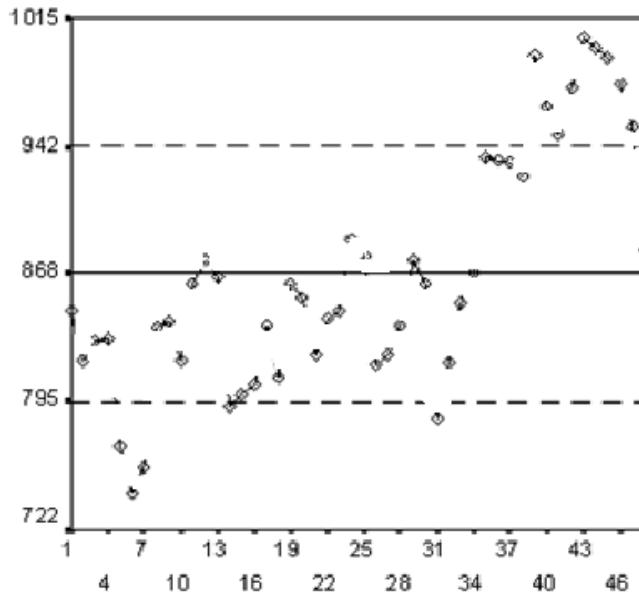
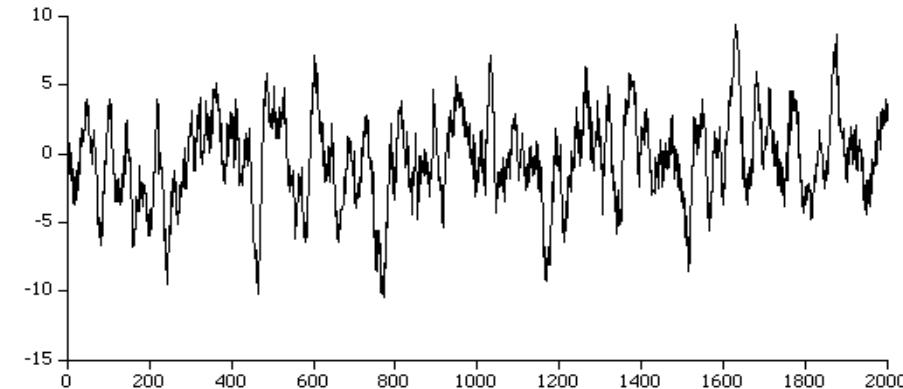
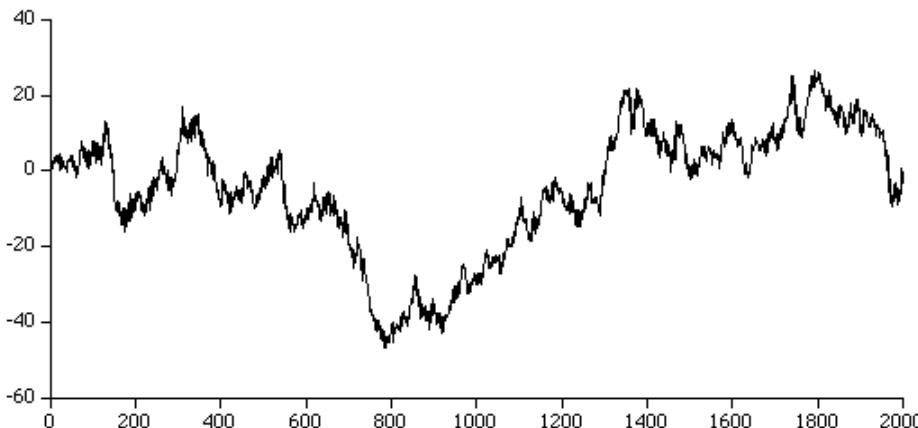


Auto Regression Analysis

- **Examples:** Which of the following is a time-series data?
 - Aggregate consumption and GDP for a country (for example, 20 years of quarterly observations = 80 observations)
 - Yen/\$, pound/\$ and Euro/\$ exchange rates (daily data for 1 year = 365 observations)
 - Cigarette consumption per capita in a state, by years
 - Rainfall data over a year
 - Sales of tea from a tea shop in a season

Auto Regression Analysis

- Examples: Which of the following graph is due to time-series data?



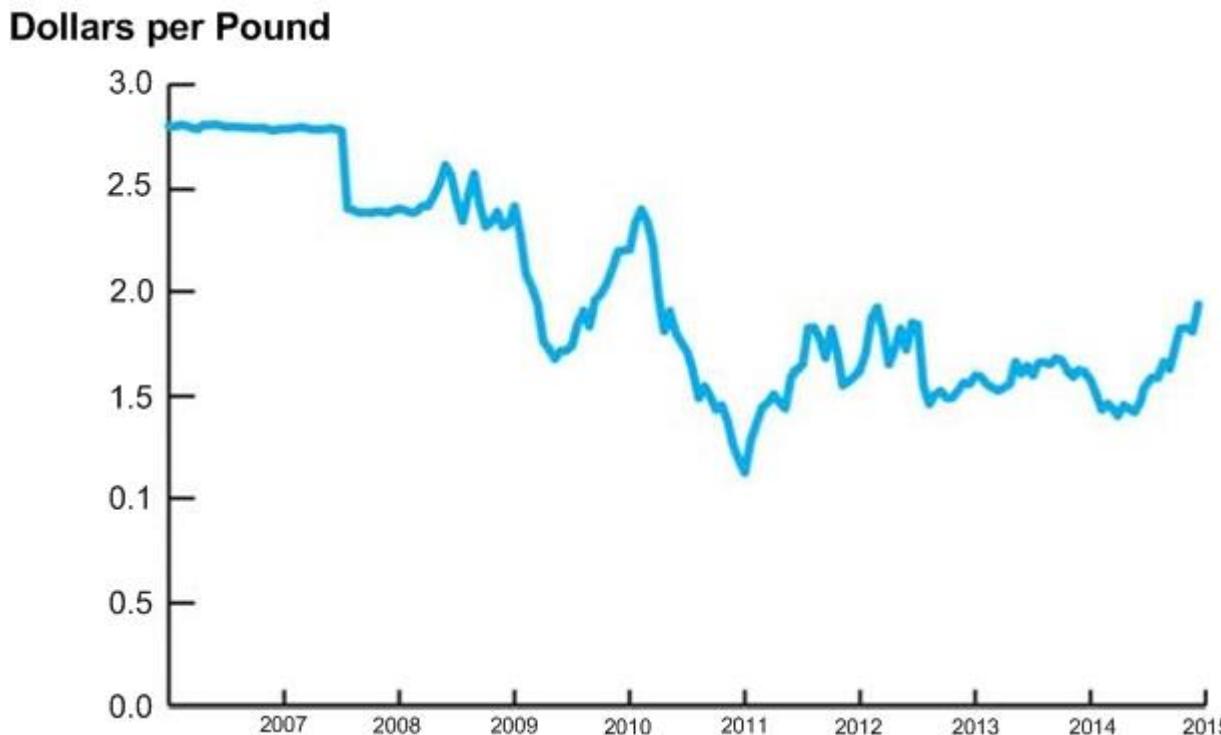
Use of Time Series Data

- To develop forecast model
 - What will the rate of inflation be next year?
- To estimate dynamic causal effects
 - If the rate of interest increases the interest rate now, what will be the effect on the rates of inflation and unemployment in 3 months? in 12 months?
 - What is the effect over time on electronics good consumption of a hike in the excise duty?
- Time dependent analysis
 - Rates of inflation and unemployment in the country can be observed only over time!

Modeling with Time Series Data

- Correlation over time
 - Serial correlation, also called autocorrelation
 - Calculating standard error
- To estimate dynamic causal effects
 - Under which dynamic effects can be estimated?
 - How to estimate?
- Forecasting model
 - Forecasting model build on regression model

Auto-Regression Model for Forecasting



- Can we predict the trend at a time say 2017?

Some Notations and Concepts

- Y_t = Value of Y in a period t
- Data set $[Y_1, Y_2, \dots, Y_{T-1}, Y_T]$: T observations on the time series random variable Y
- **Assumptions**
 - We consider only consecutive, evenly spaced observations
 - For example, monthly, 2000-2015, no missing months
 - A time series Y_t is **stationary** if its probability distribution does not change over time, that is, if the joint distribution of $(Y_{i+1}, Y_{i+2}, \dots, Y_{i+T})$ does not depend on i .
 - Stationary property implies that history is relevant. In other words, Stationary requires the future to be like the past (in a probabilistic sense).
 - Auto Regression analysis assumes that Y_t is stationary.

Some Notations and Concepts

- There are four ways to have the time series data for AutoRegression analysis
 - **Lag:** The first lag of Y_t is Y_{t-1} , its j -th lag is Y_{t-j}
 - **Difference:** The first difference of a series, Y_t is its change between period t and $t-1$, that is, $y_t = Y_t - Y_{t-1}$
 - **Log difference:** $y_t = \log(Y_t) - \log(Y_{t-1})$
 - **Percentage:** $y_t = \frac{Y_{t-1}}{Y_t} \times 100$

Some Notations and Concepts

- **Autocorrelation**

- The correlation of a series with its own lagged values is called autocorrelation (also called serial correlation)

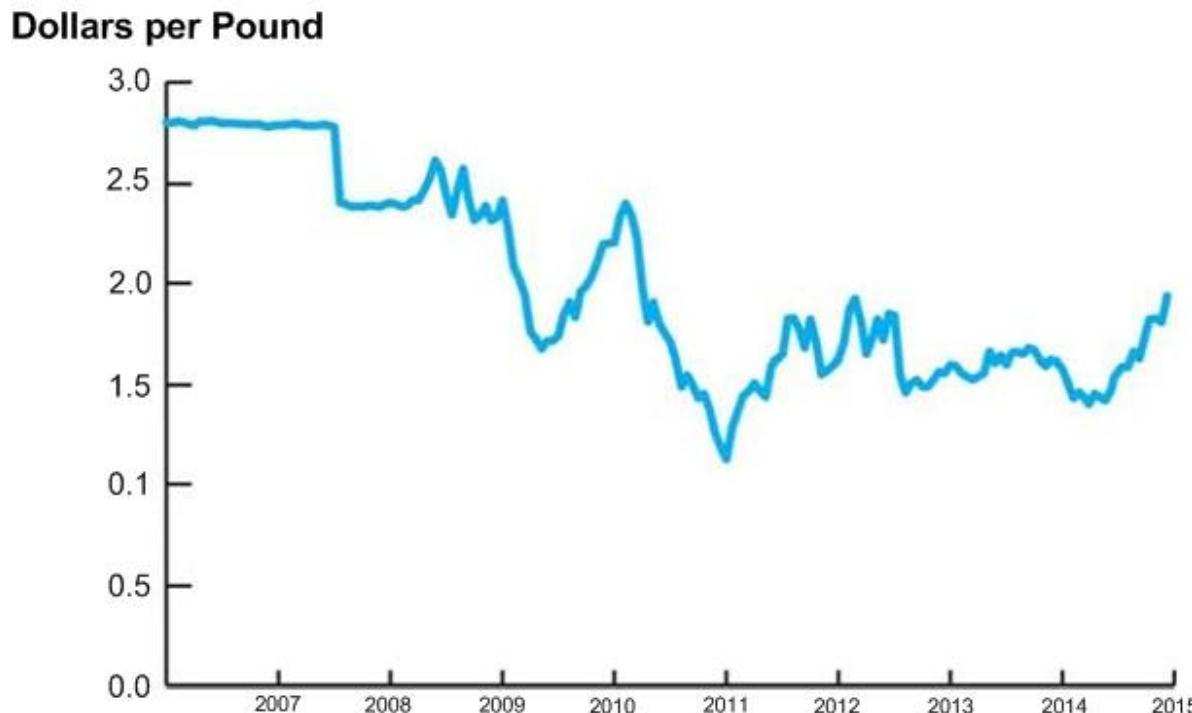
Definition 7.4: **j-th Autocorrelation**

The j-th autocorrelation, denoted by ρ_j is defined as

$$\rho_j = \frac{COV(Y_t, Y_{t-j})}{\sqrt{\sigma_{Y_t} \sigma_{Y_{t-j}}}}$$

where, $COV(Y_t, Y_{t-j})$ is the **j-th autocovariance**

Some Notations and Concepts



- For the given data, say $\rho_1 = 0.84$
 - This implies that the Dollars per Pound is highly serially correlated
- Similarly, we can determine ρ_2, ρ_3, \dots etc., and hence different regression analyses

Auto-Regression Model for Forecasting

- A natural starting point for forecasting model is to use past values of Y , that is, Y_{t-1}, Y_{t-2}, \dots to predict Y_t
- An autoregression is a regression model in which Y_t is regressed against its own lagged values.
- The number of lags used as regressors is called the **order** of autoregression
 - In first order autoregression (denoted as AR(1)), Y_t is regressed against Y_{t-1}
 - In p -th order autoregression (denoted as AR(p)), Y_t is regressed against, $Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$

p -th Order AutoRegression Model

Definition 7.5: p -th AutoRegression Model

In general, the p -th order autoregression model is defined as

$$Y_t = \beta_0 + \sum_{i=1}^p \beta_i Y_{t-i} + \varepsilon_t$$

where, $\beta_0, \beta_1, \dots, \beta_p$ is called autoregression coefficients and ε_t is the noise term or residue and in practice it is assumed to Gaussian white noise

- For example, AR(1) is $Y_t = \beta_0 + \beta_1 Y_{t-1} + \varepsilon_t$
- The task in AR analysis is to derive the "best" values for β_i $i = 0, 1, \dots, p$ given a time series Y_t .

Computing AR Coefficients

- A number of techniques known for computing the AR coefficients
- The most common method is called **Least Squares Method (LSM)**
- The LSM is based upon the **Yule-Walker equations**

$$\begin{bmatrix} 1 & r_1 & r_2 & r_3 & r_4 & \dots & r_{p-2} & r_{p-1} \\ r_1 & 1 & r_1 & r_2 & r_3 & \dots & r_{p-3} & r_{p-2} \\ r_2 & r_1 & 1 & r_1 & r_2 & \dots & r_{p-4} & r_{p-3} \\ r_3 & r_2 & r_1 & 1 & r_2 & \dots & r_{p-5} & r_{p-4} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{p-1} & r_{p-2} & r_{p-3} & r_{p-4} & r_{p-5} & \dots & r_1 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \vdots \\ \vdots \\ \beta_{p-1} \\ \beta_p \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ \vdots \\ \vdots \\ r_{p-1} \\ r_p \end{bmatrix}$$

- Here, r_i ($i = 1, 2, 3, \dots, p-1$) denotes the i -th auto correlation coefficient.
- β_0 can be chosen empirically, usually taken as zero.

Reference

- The detail material related to this lecture can be found in
The Elements of Statistical Learning, Data Mining, Inference, and Prediction (2nd Edn.), Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2014.

Chapter 1

Introduction

Linear models play a central part in modern statistical methods. On the one hand, these models can approximate a large amount of metric data structures in their entire range of definition or at least piecewise.

Linear Models and Regression Analysis

Suppose the outcome of any process is denoted by a random variable y , called as dependent (or study) variable, depends on k independent (or explanatory) variables denoted by X_1, X_2, \dots, X_k . Suppose the behaviour of y can be explained by a relationship given by

$$y = f(X_1, X_2, \dots, X_k, \beta_1, \beta_2, \dots, \beta_k) + \varepsilon$$

where f is some well-defined function and $\beta_1, \beta_2, \dots, \beta_k$ are the parameters which characterize the role and contribution of X_1, X_2, \dots, X_k , respectively. The term ε reflects the stochastic nature of the relationship between y and X_1, X_2, \dots, X_k indicates that such a relationship is not exact in nature. When $\varepsilon = 0$, then the relationship is called the mathematical model otherwise the statistical model. The term “**model**” is broadly used to represent any phenomenon in a mathematical framework.

A model or relationship is termed as linear if it is linear in parameters and nonlinear if it is not linear in parameters. In other words, if all the partial derivatives of y with respect to each of the parameters $\beta_1, \beta_2, \dots, \beta_k$, are independent of the parameters, then the model is called a **linear model**. If any of the partial derivatives of y with respect to any of the $\beta_1, \beta_2, \dots, \beta_k$ is not independent of the parameters, then the model is called as nonlinear. Note that the linearity or non-linearity of the model is not described by the linearity or nonlinearity of explanatory variables in the model.

For example

$$y = \beta_1 X_1^2 + \beta_2 \sqrt{X_2} + \beta_3 \log X_3 + \varepsilon$$

is a linear model because $\partial y / \partial \beta_i$, ($i = 1, 2, 3$) are independent of the parameters β_i , ($i = 1, 2, 3$). On the other hand,

$$y = \beta_1^2 X_1 + \beta_2 X_2 + \beta_3 \log X + \varepsilon$$

is a nonlinear model because $\partial y / \partial \beta_1 = 2\beta_1 X_1$ depends on β_1 although $\partial y / \partial \beta_2$ and $\partial y / \partial \beta_3$ are independent of any of the β_1, β_2 or β_3 .

When the function f is linear in parameters, then $y = f(X_1, X_2, \dots, X_k, \beta_1, \beta_2, \dots, \beta_k) + \varepsilon$ is called a linear model and when the function f is nonlinear in parameters, then it is called a nonlinear model. In general, the function f is chosen as

$$f(X_1, X_2, \dots, X_k, \beta_1, \beta_2, \dots, \beta_k) = \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$

to describe a linear model. Since X_1, X_2, \dots, X_k are pre-determined variables and y is the outcome, so both are known. Thus the knowledge of the model depends on the knowledge of the parameters $\beta_1, \beta_2, \dots, \beta_k$.

The linear statistical modelling essentially consists of developing approaches and tools to determine $\beta_1, \beta_2, \dots, \beta_k$ in the linear model

$$y = \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

given the observations on y and X_1, X_2, \dots, X_k .

Different statistical estimation procedures, e.g., method of maximum likelihood, principal of least squares, method of moments etc. can be employed to estimate the parameters of the model. The method of maximum likelihood needs further knowledge of the distribution of y . In contrast, the method of moments and the principle of least squares do not need any knowledge about the distribution of y .

The regression analysis is a tool to determine the values of the parameters given the data on y and X_1, X_2, \dots, X_k . The literal meaning of regression is “to move in the backward direction”. Before discussing and understanding the meaning of “backward direction”, let us find which of the following statement is correct:

S1: model generates data or

S2: data generates model.

Obviously, S1 is correct. It can be broadly thought that the model exists in nature but is unknown to the experimenter. When some values to the explanatory variables are provided, then the values for the output or study variable are generated accordingly, depending on the form of the function f and the nature of the phenomenon. So ideally, the pre-existing model gives rise to the data. Our objective is to determine the functional form of this model. Now we move in the backward direction. We propose to first collect the data

on study and explanatory variables. Then we employ some statistical techniques and use this data to know the form of function f . Equivalently, the data from the model is recorded first and then used to determine the parameters of the model. The regression analysis is a technique which helps in determining the statistical model by using the data on study and explanatory variables. The classification of linear and nonlinear regression analysis is based on the determination of linear and nonlinear models, respectively.

Consider a simple example to understand the meaning of “regression”. Suppose the yield of the crop (y) depends linearly on two explanatory variables, viz., the quality of fertilizer (X_1) and level of irrigation (X_2) as

$$y = \beta_1 X_1 + \beta_2 X_2 + \varepsilon.$$

There exist the true values of β_1 and β_2 in nature but are unknown to the experimenter. Some values on y are recorded by providing different values to X_1 and X_2 . There exists some relationship between y and X_1, X_2 which gives rise to a systematically behaved data on y , X_1 and X_2 . Such a relationship is unknown to the experimenter. To determine the model, we move in the backward direction in the sense that the collected data is used to determine the parameters β_1 and β_2 of the model. In this sense, such an approach is termed as regression analysis.

The theory and fundamentals of linear models lay the foundation for developing the tools for regression analysis that are based on valid statistical theory and concepts.

Steps in regression analysis

Regression analysis includes the following steps:

- Statement of the problem under consideration
- Choice of relevant variables
- Collection of data on relevant variables
- Specification of model
- Choice of method for fitting the data
- Fitting of model
- Model validation and criticism
- Using the chosen model(s) for the solution of the posed problem.

These steps are examined below.

1. Statement of the problem under consideration:

The first important step in conducting any regression analysis is to specify the problem and the objectives to be addressed by the regression analysis. The wrong formulation or the wrong understanding of the problem will give the erroneous statistical inferences. The choice of variables depends upon the objectives of the study and understanding of the problem. For example, the height and weight of children are related. Now there can be two issues to be addressed.

- (i) Determination of height for a given weight, or
- (ii) determination of weight for a given height.

In case 1, the height is a response variable, whereas weight is a response variable in case 2. The role of explanatory variables is also interchanged in cases 1 and 2.

2. Choice of relevant variables:

Once the problem is carefully formulated and objectives have been decided, the next question is to choose the relevant variables. It has to be kept in mind that the correct choice of variables will determine the statistical inferences correctly. For example, in an agricultural experiment, the yield depends on explanatory variables like quantity of fertilizer, rainfall, irrigation, temperature etc. These variables are denoted by X_1, X_2, \dots, X_k as a set of k explanatory variables.

3. Collection of data on relevant variables:

Once the objective of the study is clearly stated, and the variables are chosen, the next question arises is to collect data on such relevant variables. The data is essentially the measurement on these variables. For example, it is important to know how to record the data on age. For example, the data collection is on age in total complete years or date of birth is to be recorded, which can give the exact age of a specific time. Moreover, it is also important to decide that the data has to be collected on variables as quantitative variables or qualitative variables. For example, if the ages (in years) are 15,17,19,21,23, then these are quantitative values. If the ages are defined by a variable that takes value 1 if ages are less than 18 years and 0 if the ages are more than 18 years, then the earlier recorded data is converted to 1,1,0,0,0. Note that there is a loss of information in converting the quantitative data into qualitative data. The methods and approaches for qualitative and quantitative data are also different. If the study variable is binary, then **logistic regression** is used. If all explanatory variables are qualitative, then **analysis of variance** technique is used. If some explanatory variables are qualitative and others are quantitative, then **analysis of covariance** technique is used. The techniques of analysis of variance and analysis of covariance are the special cases of regression analysis.

Generally, the data is collected on n subjects, then y on data, then y denotes the response or study variable and y_1, y_2, \dots, y_n are the n values. If there are k explanatory variables, X_1, X_2, \dots, X_k then x_{ij} denotes the i^{th} value of j^{th} variable. The observation can be presented in the following table:

Notation for the data used in regression analysis

Observation Number	Response y	Explanatory variables			
		X_1	X_2	...	X_k
1	y_1	x_{11}	x_{12}	...	x_{1k}
2	y_2	x_{21}	x_{22}	...	x_{2k}
3	y_3	x_{31}	x_{32}	...	x_{3k}
:	:	:	:	:	:
n	y_n	x_{n1}	x_{n2}	...	x_{nk}

4. Specification of the model:

The experimenter or the person working in the subject usually help in determining the form of the model. Only the form of the tentative model can be ascertained, and it will depend on some unknown parameters. For example, a general form will be like

$$y = f(X_1, X_2, \dots, X_k; \beta_1, \beta_2, \dots, \beta_k) + \varepsilon$$

where ε is the random error reflecting mainly the difference in the observed value of y and the value of y obtained through the model. The form of $f(X_1, X_2, \dots, X_k; \beta_1, \beta_2, \dots, \beta_k)$ can be linear as well as nonlinear depending on the form of parameters $\beta_1, \beta_2, \dots, \beta_k$. A model is said to be linear if it is linear in parameters.

For example,

$$y = \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \varepsilon$$

$$y = \beta_1 + \beta_2 \ln X_2 + \varepsilon$$

are linear models whereas

$$y = \beta_1 X_1 + \beta_2^2 X_2 + \beta_3 X_2 + \varepsilon$$

$$y = \ln \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

are the non-linear models. Many times, the nonlinear models can be converted into linear models through some transformations. So the class of linear models is wider than what it appears initially.

If a model contains only one explanatory variable, then it is called a **simple regression model**. When there are more than one independent variables, then it is called a **multiple regression model**. When there is only one study variable, the regression is termed as **univariate regression**. When there are more than one study variables, the regression is termed as **multivariate regression**. Note that the simple and multiple regressions are not same as univariate and multivariate regressions. The choice between simple and multiple regression is determined by the number of explanatory variables, whereas the choice between univariate and multivariate regressions is determined by the number of study variables.

5. Choice of method for fitting the data:

After the model has been defined, and the data have been collected, the next task is to estimate the parameters of the model based on the collected data. This is also referred to as **parameter estimation** or **model fitting**. The most commonly used method of estimation is called the least-squares method. Under certain assumptions, the least-squares method produces estimators with desirable properties. The other estimation methods are the maximum likelihood method, ridge method, principal components method etc.

6. Fitting of the model:

The estimation of unknown parameters using appropriate method provides the values of the parameter. Substituting these values in the equation gives us a usable model. This is termed as model fitting. The estimates of parameters $\beta_1, \beta_2, \dots, \beta_k$ in the model

$$y = f(X_1, X_2, \dots, X_k, \beta_1, \beta_2, \dots, \beta_k) + \varepsilon$$

are denoted as $\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_k$ which gives the fitted model as

$$y = f(X_1, X_2, \dots, X_k, \hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_k).$$

When the value of y is obtained for the given values of X_1, X_2, \dots, X_k , it is denoted as \hat{y} and called as fitted value.

The fitted equation is used for prediction. In this case, \hat{y} is termed as the **predicted value**. Note that the fitted value is where the values used for explanatory variables correspond to one of the n observations in the data, whereas predicted value is the one obtained for any set of values of explanatory variables. It is not generally recommended to predict the y -values for the set of those values of explanatory variables which lie for outside the range of data. When the values of explanatory variables are the future values of explanatory variables, the predicted values are called as forecasted values.

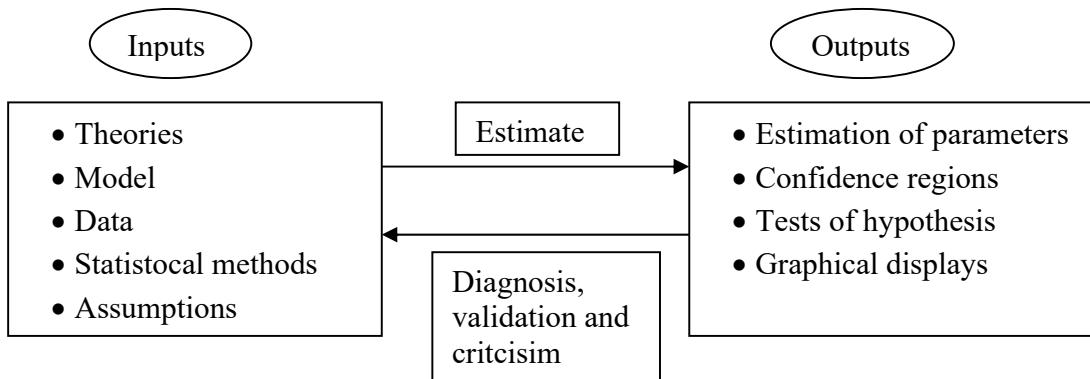
There are different methodologies based on regression analysis. They are described in the following table:

Type of Regression	Conditions
Univariate	Only one quantitative response variable
Multivariate	Two or more quantitative response variables
Simple	Only one explanatory variable
Multiple	Two or more explanatory variables
Linear	All parameters enter the equation linearly, possibly after transformation of the data
Nonlinear	The relationship between the response and some of the explanatory variables is nonlinear or some of the parameters appear nonlinearly, but no transformation is possible to make the parameters appear linearly
Analysis of variance	All explanatory variables are qualitative variables
Analysis of Covariance	Some explanatory variables are quantitative variables and others are qualitative variables
Logistic	The response variable is qualitative

7. Model criticism and selection

The validity of the statistical method to be used for regression analysis depends on various assumptions. These assumptions become essentially the assumptions for the model and the data. The quality of statistical inferences heavily depends on whether these assumptions are satisfied or not. For making these assumptions to be valid and to be satisfied, care is needed from the beginning of the experiment. One has to be careful in choosing the required assumptions and to decide as well to determine if the assumptions are valid for the given experimental conditions or not? It is also important to determine that the situations in which the assumptions may not meet.

The validation of the assumptions must be made before drawing any statistical conclusion. Any departure from the validity of assumptions will be reflected in the statistical inferences. In fact, the regression analysis is an iterative process where the outputs are used to diagnose, validate, criticize and modify the inputs. The iterative process is illustrated in the following figure.



8. Objectives of regression analysis

The determination of the explicit form of the regression equation is the ultimate objective of regression analysis. It is finally a good and valid relationship between study variable and explanatory variables. Such a regression equation can be used for several purposes. For example, to determine the role of any explanatory variable in the joint relationship in any policy formulation, to forecast the values of the response variable for a given set of values of explanatory variables. The regression equation helps in understanding the interrelationships of variables among them.

Chapter 2

Simple Linear Regression Analysis

The simple linear regression model

We consider the modelling between the dependent and one independent variable. When there is only one independent variable in the linear regression model, the model is generally termed as a simple linear regression model. When there are more than one independent variables in the model, then the linear model is termed as the multiple linear regression model.

The linear model

Consider a simple linear regression model

$$y = \beta_0 + \beta_1 X + \varepsilon$$

where y is termed as the dependent or study variable and X is termed as the independent or explanatory variable. The terms β_0 and β_1 are the parameters of the model. The parameter β_0 is termed as an intercept term, and the parameter β_1 is termed as the slope parameter. These parameters are usually called as **regression coefficients**. The unobservable error component ε accounts for the failure of data to lie on the straight line and represents the difference between the true and observed realization of y . There can be several reasons for such difference, e.g., the effect of all deleted variables in the model, variables may be qualitative, inherent randomness in the observations etc. We assume that ε is observed as independent and identically distributed random variable with mean zero and constant variance σ^2 . Later, we will additionally assume that ε is normally distributed.

The independent variables are viewed as controlled by the experimenter, so it is considered as non-stochastic whereas y is viewed as a random variable with

$$E(y) = \beta_0 + \beta_1 X$$

and

$$Var(y) = \sigma^2.$$

Sometimes X can also be a random variable. In such a case, instead of the sample mean and sample variance of y , we consider the conditional mean of y given $X = x$ as

$$E(y | x) = \beta_0 + \beta_1 x$$

and the conditional variance of y given $X = x$ as

$$\text{Var}(y|x) = \sigma^2.$$

When the values of β_0, β_1 and σ^2 are known, the model is completely described. The parameters β_0, β_1 and σ^2 are generally unknown in practice and ε is unobserved. The determination of the statistical model $y = \beta_0 + \beta_1 X + \varepsilon$ depends on the determination (i.e., estimation) of β_0, β_1 and σ^2 . In order to know the values of these parameters, n pairs of observations (x_i, y_i) ($i = 1, \dots, n$) on (X, y) are observed/collected and are used to determine these unknown parameters.

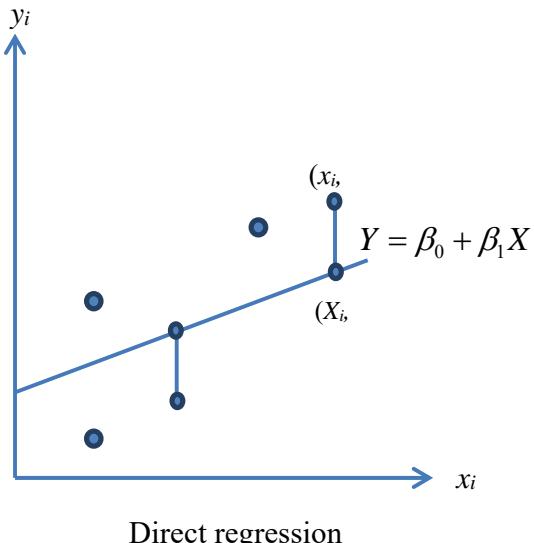
Various methods of estimation can be used to determine the estimates of the parameters. Among them, the methods of least squares and maximum likelihood are the popular methods of estimation.

Least squares estimation

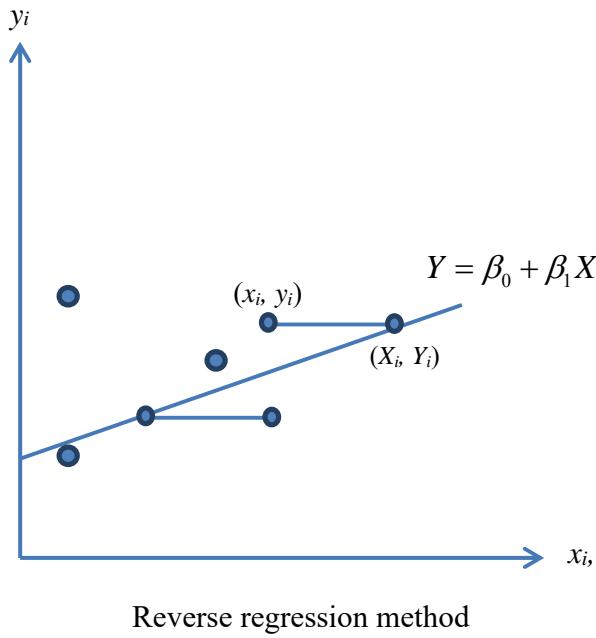
Suppose a sample of n sets of paired observations (x_i, y_i) ($i = 1, 2, \dots, n$) is available. These observations are assumed to satisfy the simple linear regression model, and so we can write

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (i = 1, 2, \dots, n).$$

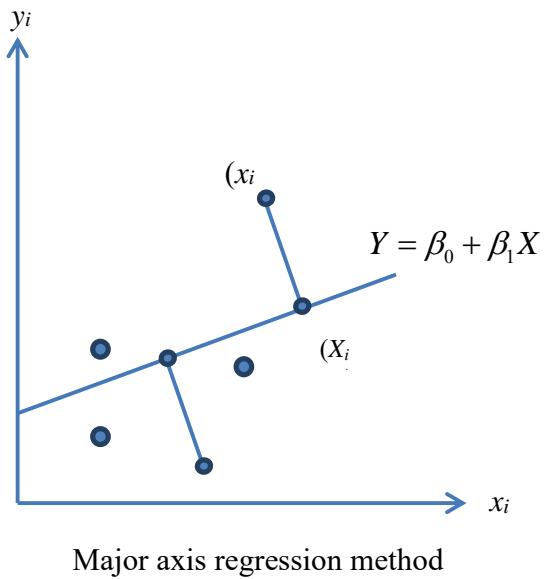
The principle of least squares estimates the parameters β_0 and β_1 by minimizing the sum of squares of the difference between the observations and the line in the scatter diagram. Such an idea is viewed from different perspectives. When the **vertical difference** between the observations and the line in the scatter diagram is considered, and its sum of squares is minimized to obtain the estimates of β_0 and β_1 , the method is known as **direct regression**.



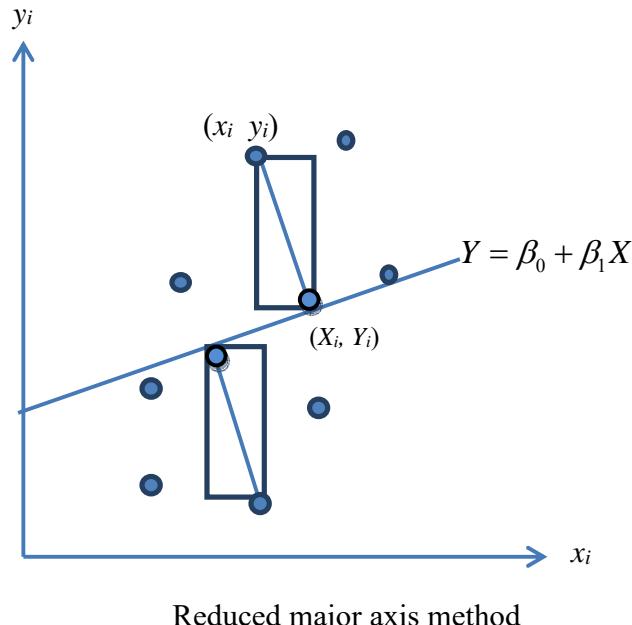
Alternatively, the sum of squares of the difference between the observations and the line in the horizontal direction in the scatter diagram can be minimized to obtain the estimates of β_0 and β_1 . This is known as a **reverse (or inverse) regression method**.



Instead of horizontal or vertical errors, if the sum of squares of perpendicular distances between the observations and the line in the scatter diagram is minimized to obtain the estimates of β_0 and β_1 , the method is known as **orthogonal regression or major axis regression method**.



Instead of minimizing the distance, the area can also be minimized. The **reduced major axis regression method** minimizes the sum of the areas of rectangles defined between the observed data points and the nearest point on the line in the scatter diagram to obtain the estimates of regression coefficients. This is shown in the following figure:



The method of **least absolute deviation regression** considers the sum of the absolute deviation of the observations from the line in the vertical direction in the scatter diagram as in the case of direct regression to obtain the estimates of β_0 and β_1 .

No assumption is required about the form of the probability distribution of ε_i in deriving the least squares estimates. For the purpose of deriving the statistical inferences only, we assume that ε_i 's are random variable with $E(\varepsilon_i) = 0$, $Var(\varepsilon_i) = \sigma^2$ and $Cov(\varepsilon_i, \varepsilon_j) = 0$ for all $i \neq j$ ($i, j = 1, 2, \dots, n$). This assumption is needed to find the mean, variance and other properties of the least-squares estimates. The assumption that ε_i 's are normally distributed is utilized while constructing the tests of hypotheses and confidence intervals of the parameters.

Based on these approaches, different estimates of β_0 and β_1 are obtained which have different statistical properties. Among them, the direct regression approach is more popular. Generally, the direct regression estimates are referred to as the **least-squares estimates** or **ordinary least squares estimates**.

Direct regression method

This method is also known as the **ordinary least squares estimation**. Assuming that a set of n paired observations on (x_i, y_i) , $i = 1, 2, \dots, n$ are available which satisfy the linear regression model $y = \beta_0 + \beta_1 X + \varepsilon$.

So we can write the model for each observation as $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, $(i = 1, 2, \dots, n)$.

The direct regression approach minimizes the sum of squares

$$S(\beta_0, \beta_1) = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

with respect to β_0 and β_1 .

The partial derivatives of $S(\beta_0, \beta_1)$ with respect to β_0 is

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)$$

and the partial derivative of $S(\beta_0, \beta_1)$ with respect to β_1 is

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_1} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) x_i.$$

The solutions of β_0 and β_1 are obtained by setting

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_0} = 0$$

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_1} = 0.$$

The solutions of these two equations are called the **direct regression estimators**, or usually called as the **ordinary least squares (OLS)** estimators of β_0 and β_1 .

This gives the ordinary least squares estimates b_0 of β_0 and b_1 of β_1 as

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{s_{xy}}{s_{xx}}$$

where

$$s_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}), \quad s_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Further, we have

$$\frac{\partial^2 S(\beta_0, \beta_1)}{\partial \beta_0^2} = -2 \sum_{i=1}^n (-1) = 2n,$$

$$\frac{\partial^2 S(\beta_0, \beta_1)}{\partial \beta_1^2} = 2 \sum_{i=1}^n x_i^2$$

$$\frac{\partial^2 S(\beta_0, \beta_1)}{\partial \beta_0 \partial \beta_1} = 2 \sum_{i=1}^n x_i = 2n\bar{x}.$$

The Hessian matrix which is the matrix of second-order partial derivatives, in this case, is given as

$$\begin{aligned} H^* &= \begin{pmatrix} \frac{\partial^2 S(\beta_0, \beta_1)}{\partial \beta_0^2} & \frac{\partial^2 S(\beta_0, \beta_1)}{\partial \beta_0 \partial \beta_1} \\ \frac{\partial^2 S(\beta_0, \beta_1)}{\partial \beta_0 \partial \beta_1} & \frac{\partial^2 S(\beta_0, \beta_1)}{\partial \beta_1^2} \end{pmatrix} \\ &= 2 \begin{pmatrix} n & n\bar{x} \\ n\bar{x} & \sum_{i=1}^n x_i^2 \end{pmatrix} \\ &= 2 \begin{pmatrix} \ell' \\ x' \end{pmatrix} (\ell, x) \end{aligned}$$

where $\ell = (1, 1, \dots, 1)'$ is a n -vector of elements unity and $x = (x_1, \dots, x_n)'$ is a n -vector of observations on X .

The matrix H^* is positive definite if its determinant and the element in the first row and column of H^* are positive. The determinant of H^* is given by

$$\begin{aligned} |H^*| &= 4 \left(n \sum_{i=1}^n x_i^2 - n^2 \bar{x}^2 \right) \\ &= 4n \sum_{i=1}^n (x_i - \bar{x})^2 \\ &\geq 0. \end{aligned}$$

The case when $\sum_{i=1}^n (x_i - \bar{x})^2 = 0$ is not interesting because all the observations, in this case, are identical, i.e.

$x_i = c$ (some constant). In such a case, there is no relationship between x and y in the context of regression

analysis. Since $\sum_{i=1}^n (x_i - \bar{x})^2 > 0$, therefore $|H| > 0$. So H is positive definite for any (β_0, β_1) , therefore, $S(\beta_0, \beta_1)$ has a global minimum at (b_0, b_1) .

The **fitted line** or the **fitted linear regression model** is

$$y = b_0 + b_1 x.$$

The predicted values are

$$\hat{y}_i = b_0 + b_1 x_i \quad (i = 1, 2, \dots, n).$$

The difference between the observed value y_i and the fitted (or predicted) value \hat{y}_i is called a **residual**. The i^{th} residual is defined as

$$\begin{aligned} e_i &= y_i - \hat{y}_i \quad (i = 1, 2, \dots, n) \\ &= y_i - \hat{y}_i \\ &= y_i - (b_0 + b_1 x_i). \end{aligned}$$

Properties of the direct regression estimators:

Unbiased property:

Note that $b_1 = \frac{s_{xy}}{s_{xx}}$ and $b_0 = \bar{y} - b_1 \bar{x}$ are the linear combinations of y_i ($i = 1, \dots, n$).

Therefore

$$b_1 = \sum_{i=1}^n k_i y_i$$

where $k_i = (x_i - \bar{x}) / s_{xx}$. Note that $\sum_{i=1}^n k_i = 0$ and $\sum_{i=1}^n k_i x_i = 1$, so

$$\begin{aligned} E(b_1) &= \sum_{i=1}^n k_i E(y_i) \\ &= \sum_{i=1}^n k_i (\beta_0 + \beta_1 x_i) \\ &= \beta_1. \end{aligned}$$

This b_1 is an unbiased estimator of β_1 . Next

$$\begin{aligned} E(b_0) &= E[\bar{y} - b_1 \bar{x}] \\ &= E[\beta_0 + \beta_1 \bar{x} + \bar{\varepsilon} - b_1 \bar{x}] \\ &= \beta_0 + \beta_1 \bar{x} - \beta_1 \bar{x} \\ &= \beta_0. \end{aligned}$$

Thus b_0 is an unbiased estimator of β_0 .

Variances:

Using the assumption that y_i 's are independently distributed, the variance of b_1 is

$$\begin{aligned} Var(b_1) &= \sum_{i=1}^n k_i^2 Var(y_i) + \sum_i \sum_{j \neq i} k_i k_j Cov(y_i, y_j) \\ &= \sigma^2 \frac{\sum_i (x_i - \bar{x})^2}{s_{xx}^2} \quad (Cov(y_i, y_j) = 0 \text{ as } y_1, \dots, y_n \text{ are independent}) \\ &= \frac{\sigma^2 s_{xx}}{s_{xx}^2} \\ &= \frac{\sigma^2}{s_{xx}}. \end{aligned}$$

The variance of b_0 is

$$Var(b_0) = Var(\bar{y}) + \bar{x}^2 Var(b_1) - 2\bar{x}Cov(\bar{y}, b_1).$$

First, we find that

$$\begin{aligned} Cov(\bar{y}, b_1) &= E[\{\bar{y} - E(\bar{y})\}\{b_1 - E(b_1)\}] \\ &= E\left[\bar{\varepsilon}\left(\sum_i c_i y_i - \beta_1\right)\right] \\ &= \frac{1}{n} E\left[\left(\sum_i \varepsilon_i\right)\left(\beta_0 \sum_i c_i + \beta_1 \sum_i c_i x_i + \sum_i c_i \varepsilon_i\right) - \beta_1 \sum_i \varepsilon_i\right] \\ &= \frac{1}{n}[0 + 0 + 0 + 0] \\ &= 0 \end{aligned}$$

So

$$Var(b_0) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}} \right).$$

Covariance:

The covariance between b_0 and b_1 is

$$\begin{aligned} Cov(b_0, b_1) &= Cov(\bar{y}, b_1) - \bar{x}Var(b_1) \\ &= -\frac{\bar{x}}{s_{xx}} \sigma^2. \end{aligned}$$

It can further be shown that the ordinary least squares estimators b_0 and b_1 possess the minimum variance in the class of linear and unbiased estimators. So they are termed as the Best Linear Unbiased Estimators (BLUE). Such a property is known as the **Gauss-Markov theorem**, which is discussed later in multiple linear regression model.

Residual sum of squares:

The residual sum of squares is given as

$$\begin{aligned}
 SS_{res} &= \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\
 &= \sum_{i=1}^n (y_i - b_0 - b_1 x_i)^2 \\
 &= \sum_{i=1}^n [y_i - \bar{y} + b_1 \bar{x} - b_1 x_i]^2 \\
 &= \sum_{i=1}^n [(y_i - \bar{y}) - b_1(x_i - \bar{x})]^2 \\
 &= \sum_{i=1}^n (y_i - \bar{y})^2 + b_1^2 \sum_{i=1}^n (x_i - \bar{x})^2 - 2b_1 \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \\
 &= s_{yy} + b_1^2 s_{xx} - 2b_1 s_{xy} \\
 &= s_{yy} - b_1^2 s_{xx} \\
 &= s_{yy} - \left(\frac{s_{xy}}{s_{xx}} \right)^2 s_{xx} \\
 &= s_{yy} - \frac{s_{xy}^2}{s_{xx}} \\
 &= s_{yy} - b_1 s_{xy}.
 \end{aligned}$$

where $s_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2$, $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$.

Estimation of σ^2

The estimator of σ^2 is obtained from the residual sum of squares as follows. Assuming that y_i is normally distributed, it follows that SS_{res} has a χ^2 distribution with $(n-2)$ degrees of freedom, so

$$\frac{SS_{res}}{\sigma^2} \sim \chi^2(n-2).$$

Thus using the result about the expectation of a chi-square random variable, we have

$$E(SS_{res}) = (n-2)\sigma^2.$$

Thus an unbiased estimator of σ^2 is

$$s^2 = \frac{SS_{res}}{n-2}.$$

Note that SS_{res} has only $(n-2)$ degrees of freedom. The two degrees of freedom are lost due to estimation of b_0 and b_1 . Since s^2 depends on the estimates b_0 and b_1 , so it is a **model-dependent estimate** of σ^2 .

Estimate of variances of b_0 and b_1 :

The estimators of variances of b_0 and b_1 are obtained by replacing σ^2 by its estimate $\hat{\sigma}^2 = s^2$ as follows:

$$\widehat{Var}(b_0) = s^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}} \right)$$

and

$$\widehat{Var}(b_1) = \frac{s^2}{s_{xx}}.$$

It is observed that since $\sum_{i=1}^n (y_i - \hat{y}_i) = 0$, so $\sum_{i=1}^n e_i = 0$. In the light of this property, e_i can be regarded as an estimate of unknown ε_i ($i = 1, \dots, n$). This helps in verifying the different model assumptions on the basis of the given sample (x_i, y_i) , $i = 1, 2, \dots, n$.

Further, note that

- (i) $\sum_{i=1}^n x_i e_i = 0$,
- (ii) $\sum_{i=1}^n \hat{y}_i e_i = 0$,
- (iii) $\sum_{i=1}^n y_i = \sum_{i=1}^n \hat{y}_i$ and
- (iv) the fitted line always passes through (\bar{x}, \bar{y}) .

Centered Model:

Sometimes it is useful to measure the independent variable around its mean. In such a case, the model $y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$ has a centred version as follows:

$$\begin{aligned} y_i &= \beta_0 + \beta_1(x_i - \bar{x}) + \beta_1 \bar{x} + \varepsilon_i \quad (i = 1, 2, \dots, n) \\ &= \beta_0^* + \beta_1(x_i - \bar{x}) + \varepsilon_i \end{aligned}$$

where $\beta_0^* = \beta_0 + \beta_1 \bar{x}$. The sum of squares due to error is given by

$$S(\beta_0^*, \beta_1) = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n \left[y_i - \beta_0^* - \beta_1(x_i - \bar{x}) \right]^2.$$

Now solving

$$\frac{\partial S(\beta_0^*, \beta_1)}{\partial \beta_0^*} = 0$$

$$\frac{\partial S(\beta_0^*, \beta_1)}{\partial \beta_1} = 0,$$

we get the direct regression least squares estimates of β_0^* and β_1 as

$$b_0^* = \bar{y}$$

and

$$b_1 = \frac{s_{xy}}{s_{xx}},$$

respectively.

Thus the form of the estimate of slope parameter β_1 remains the same in the usual and centered model whereas the form of the estimate of intercept term changes in the usual and centered models.

Further, the Hessian matrix of the second order partial derivatives of $S(\beta_0^*, \beta_1)$ with respect to β_0^* and β_1 is positive definite at $\beta_0^* = b_0^*$ and $\beta_1 = b_1$ which ensures that $S(\beta_0^*, \beta_1)$ is minimized at $\beta_0^* = b_0^*$ and $\beta_1 = b_1$.

Under the assumption that $E(\varepsilon_i) = 0$, $Var(\varepsilon_i) = \sigma^2$ and $Cov(\varepsilon_i \varepsilon_j) = 0$ for all $i \neq j = 1, 2, \dots, n$, it follows that

$$\begin{aligned} E(b_0^*) &= \beta_0^*, \quad E(b_1) = \beta_1, \\ Var(b_0^*) &= \frac{\sigma^2}{n}, \quad Var(b_1) = \frac{\sigma^2}{s_{xx}}. \end{aligned}$$

In this case, the fitted model of $y_i = \beta_0^* + \beta_1(x_i - \bar{x}) + \varepsilon_i$ is

$$y = \bar{y} + b_1(x - \bar{x}),$$

and the predicted values are

$$\hat{y}_i = \bar{y} + b_1(x_i - \bar{x}) \quad (i = 1, \dots, n).$$

Note that in the centered model

$$Cov(b_0^*, b_1) = 0.$$

No intercept term model:

Sometimes in practice, a model without an intercept term is used in those situations when $x_i = 0 \Rightarrow y_i = 0$ for all $i = 1, 2, \dots, n$. A no-intercept model is

$$y_i = \beta_1 x_i + \varepsilon_i \quad (i = 1, 2, \dots, n).$$

For example, in analyzing the relationship between the velocity (y) of a car and its acceleration (X), the velocity is zero when acceleration is zero.

Using the data (x_i, y_i) , $i = 1, 2, \dots, n$, the direct regression least-squares estimate of β_1 is obtained by

minimizing $S(\beta_1) = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \beta_1 x_i)^2$ and solving

$$\frac{\partial S(\beta_1)}{\partial \beta_1} = 0$$

gives the estimator of β_1 as

$$b_1^* = \frac{\sum_{i=1}^n y_i x_i}{\sum_{i=1}^n x_i^2}.$$

The second-order partial derivative of $S(\beta_1)$ with respect to β_1 at $\beta_1 = b_1$ is positive which insures that b_1 minimizes $S(\beta_1)$.

Using the assumption that $E(\varepsilon_i) = 0$, $Var(\varepsilon_i) = \sigma^2$ and $Cov(\varepsilon_i \varepsilon_j) = 0$ for all $i \neq j = 1, 2, \dots, n$, the properties of b_1^* can be derived as follows:

$$\begin{aligned} E(b_1^*) &= \frac{\sum_{i=1}^n x_i E(y_i)}{\sum_{i=1}^n x_i^2} \\ &= \frac{\sum_{i=1}^n x_i^2 \beta_1}{\sum_{i=1}^n x_i^2} \\ &= \beta_1 \end{aligned}$$

This b_1^* is an unbiased estimator of β_1 . The variance of b_1^* is obtained as follows:

$$\begin{aligned}
Var(b_1^*) &= \frac{\sum_{i=1}^n x_i^2 Var(y_i)}{\left(\sum_{i=1}^n x_i^2\right)^2} \\
&= \sigma^2 \frac{\sum_{i=1}^n x_i^2}{\left(\sum_{i=1}^n x_i^2\right)^2} \\
&= \frac{\sigma^2}{\sum_{i=1}^n x_i^2}
\end{aligned}$$

and an unbiased estimator of σ^2 is obtained as

$$\frac{\sum_{i=1}^n y_i^2 - b_1 \sum_{i=1}^n y_i x_i}{n-1}.$$

Maximum likelihood estimation

We assume that ε_i 's ($i = 1, 2, \dots, n$) are independent and identically distributed following a normal distribution $N(0, \sigma^2)$. Now we use the method of maximum likelihood to estimate the parameters of the linear regression model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (i = 1, 2, \dots, n),$$

the observations y_i ($i = 1, 2, \dots, n$) are independently distributed with $N(\beta_0 + \beta_1 x_i, \sigma^2)$ for all $i = 1, 2, \dots, n$.

The likelihood function of the given observations (x_i, y_i) and unknown parameters β_0, β_1 and σ^2 is

$$L(x_i, y_i; \beta_0, \beta_1, \sigma^2) = \prod_{i=1}^n \left(\frac{1}{2\pi\sigma^2} \right)^{1/2} \exp \left[-\frac{1}{2\sigma^2} (y_i - \beta_0 - \beta_1 x_i)^2 \right].$$

The maximum likelihood estimates of β_0, β_1 and σ^2 can be obtained by maximizing $L(x_i, y_i; \beta_0, \beta_1, \sigma^2)$ or equivalently in $\ln L(x_i, y_i; \beta_0, \beta_1, \sigma^2)$ where

$$\ln L(x_i, y_i; \beta_0, \beta_1, \sigma^2) = -\left(\frac{n}{2}\right) \ln 2\pi - \left(\frac{n}{2}\right) \ln \sigma^2 - \left(\frac{1}{2\sigma^2}\right) \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2.$$

The normal equations are obtained by partial differentiation of log-likelihood with respect to β_0 , β_1 and σ^2 and equating them to zero as follows:

$$\frac{\partial \ln L(x_i, y_i; \beta_0, \beta_1, \sigma^2)}{\partial \beta_0} = -\frac{1}{\sigma^2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial \ln L(x_i, y_i; \beta_0, \beta_1, \sigma^2)}{\partial \beta_1} = -\frac{1}{\sigma^2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) x_i = 0$$

and

$$\frac{\partial \ln L(x_i, y_i; \beta_0, \beta_1, \sigma^2)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 = 0.$$

The solution of these normal equations give the maximum likelihood estimates of β_0 , β_1 and σ^2 as

$$\tilde{b}_0 = \bar{y} - \tilde{b}_1 \bar{x}$$

$$\tilde{b}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{s_{xy}}{s_{xx}}$$

and

$$\tilde{s}^2 = \frac{\sum_{i=1}^n (y_i - \tilde{b}_0 - \tilde{b}_1 x_i)^2}{n}$$

respectively.

It can be verified that the Hessian matrix of second-order partial derivation of $\ln L$ with respect to β_0 , β_1 , and σ^2 is negative definite at $\beta_0 = \tilde{b}_0$, $\beta_1 = \tilde{b}_1$, and $\sigma^2 = \tilde{s}^2$ which ensures that the likelihood function is maximized at these values.

Note that the least-squares and maximum likelihood estimates of β_0 and β_1 are identical. The least-squares and maximum likelihood estimates of σ^2 are different. In fact, the least-squares estimate of σ^2 is

$$s^2 = \frac{1}{n-2} \sum_{i=1}^n (y_i - \bar{y})^2$$

so that it is related to the maximum likelihood estimate as

$$\tilde{s}^2 = \frac{n-2}{n} s^2.$$

Thus \tilde{b}_0 and \tilde{b}_1 are unbiased estimators of β_0 and β_1 whereas \tilde{s}^2 is a biased estimate of σ^2 , but it is asymptotically unbiased. The variances of \tilde{b}_0 and \tilde{b}_1 are same as of b_0 and b_1 respectively but $Var(\tilde{s}^2) < Var(s^2)$.

Testing of hypotheses and confidence interval estimation for slope parameter:

Now we consider the tests of hypothesis and confidence interval estimation for the slope parameter of the model under two cases, viz., when σ^2 is known and when σ^2 is unknown.

Case 1: When σ^2 is known:

Consider the simple linear regression model $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (i=1,2,\dots,n)$. It is assumed that ε_i 's are independent and identically distributed and follow $N(0, \sigma^2)$.

First, we develop a test for the null hypothesis related to the slope parameter

$$H_0 : \beta_1 = \beta_{10}$$

where β_{10} is some given constant.

Assuming σ^2 to be known, we know that $E(b_1) = \beta_1$, $Var(b_1) = \frac{\sigma^2}{s_{xx}}$ and b_1 is a linear combination of normally distributed y_i 's. So

$$b_1 \sim N\left(\beta_1, \frac{\sigma^2}{s_{xx}}\right)$$

and so the following statistic can be constructed

$$Z_1 = \frac{b_1 - \beta_{10}}{\sqrt{\frac{\sigma^2}{s_{xx}}}}$$

which is distributed as $N(0,1)$ when H_0 is true.

A decision rule to test $H_1 : \beta_1 \neq \beta_{10}$ can be framed as follows:

Reject H_0 if $|Z_1| > Z_{\alpha/2}$

where $Z_{\alpha/2}$ is the $\alpha/2$ percent points on the normal distribution.

Similarly, the decision rule for one-sided alternative hypothesis can also be framed.

The $100(1-\alpha)\%$ confidence interval for β_1 can be obtained using the Z_1 statistic as follows:

$$P[-z_{\alpha/2} \leq Z_1 \leq z_{\alpha/2}] = 1 - \alpha$$

$$P\left[-z_{\alpha/2} \leq \frac{b_1 - \beta_1}{\sqrt{\frac{\sigma^2}{s_{xx}}}} \leq z_{\alpha/2}\right] = 1 - \alpha$$

$$P\left[b_1 - z_{\alpha/2} \sqrt{\frac{\sigma^2}{s_{xx}}} \leq \beta_1 \leq b_1 + z_{\alpha/2} \sqrt{\frac{\sigma^2}{s_{xx}}}\right] = 1 - \alpha.$$

So $100(1-\alpha)\%$ confidence interval for β_1 is

$$\left[b_1 - z_{\alpha/2} \sqrt{\frac{\sigma^2}{s_{xx}}}, b_1 + z_{\alpha/2} \sqrt{\frac{\sigma^2}{s_{xx}}}\right]$$

where $z_{\alpha/2}$ is the $\alpha/2$ percentage point of the $N(0,1)$ distribution.

Case 2: When σ^2 is unknown:

When σ^2 is unknown then we proceed as follows. We know that

$$\frac{SS_{res}}{\sigma^2} \sim \chi^2(n-2)$$

and

$$E\left(\frac{SS_{res}}{n-2}\right) = \sigma^2.$$

Further, SS_{res}/σ^2 and b_1 are independently distributed. This result will be proved formally later in the next module on multiple linear regression. This result also follows from the result that under normal distribution, the maximum likelihood estimates, viz., the sample mean (estimator of population mean) and the sample variance (estimator of population variance) are independently distributed, so b_1 and s^2 are also independently distributed.

Thus the following statistic can be constructed:

$$t_0 = \frac{b_1 - \beta_1}{\sqrt{\frac{\hat{\sigma}^2}{s_{xx}}}}$$

$$= \frac{b_1 - \beta_1}{\sqrt{\frac{SS_{res}}{(n-2)s_{xx}}}}$$

which follows a t -distribution with $(n-2)$ degrees of freedom, denoted as t_{n-2} , when H_0 is true.

A decision rule to test $H_1: \beta_1 \neq \beta_{10}$ is to

reject H_0 if $|t_0| > t_{n-2,\alpha/2}$

where $t_{n-2,\alpha/2}$ is the $\alpha/2$ percent point of the t -distribution with $(n-2)$ degrees of freedom. Similarly, the decision rule for the one-sided alternative hypothesis can also be framed.

The $100(1-\alpha)\%$ confidence interval of β_1 can be obtained using the t_0 statistic as follows:

Consider

$$P[-t_{\alpha/2} \leq t_0 \leq t_{\alpha/2}] = 1 - \alpha$$

$$P\left[-t_{\alpha/2} \leq \frac{b_1 - \beta_1}{\sqrt{\frac{\hat{\sigma}^2}{s_{xx}}}} \leq t_{\alpha/2}\right] = 1 - \alpha$$

$$P\left[b_1 - t_{\alpha/2} \sqrt{\frac{\hat{\sigma}^2}{s_{xx}}} \leq \beta_1 \leq b_1 + t_{\alpha/2} \sqrt{\frac{\hat{\sigma}^2}{s_{xx}}}\right] = 1 - \alpha.$$

So the $100(1-\alpha)\%$ confidence interval β_1 is

$$\left[b_1 - t_{n-2,\alpha/2} \sqrt{\frac{SS_{res}}{(n-2)s_{xx}}}, b_1 + t_{n-2,\alpha/2} \sqrt{\frac{SS_{res}}{(n-2)s_{xx}}} \right].$$

Testing of hypotheses and confidence interval estimation for intercept term:

Now, we consider the tests of hypothesis and confidence interval estimation for intercept term under two cases, viz., when σ^2 is known and when σ^2 is unknown.

Case 1: When σ^2 is known:

Suppose the null hypothesis under consideration is

$$H_0: \beta_0 = \beta_{00},$$

where σ^2 is known, then using the result that $E(b_0) = \beta_0$, $Var(b_0) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_x^2} \right)$ and b_0 is a linear combination of normally distributed random variables, the following statistic

$$Z_0 = \frac{b_0 - \beta_{00}}{\sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_x^2} \right)}}$$

has a $N(0,1)$ distribution when H_0 is true.

A decision rule to test $H_1: \beta_0 \neq \beta_{00}$ can be framed as follows:

Reject H_0 if $|Z_0| > Z_{\alpha/2}$

where $Z_{\alpha/2}$ is the $\alpha/2$ percentage points on the normal distribution. Similarly, the decision rule for one-sided alternative hypothesis can also be framed.

The $100(1-\alpha)\%$ confidence intervals for β_0 when σ^2 is known can be derived using the Z_0 statistic as follows:

$$P[-z_{\alpha/2} \leq Z_0 \leq z_{\alpha/2}] = 1 - \alpha$$

$$P\left[-z_{\alpha/2} \leq \frac{b_0 - \beta_0}{\sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)}} \leq z_{\alpha/2}\right] = 1 - \alpha$$

$$P\left[b_0 - z_{\alpha/2} \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)} \leq \beta_0 \leq b_0 + z_{\alpha/2} \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)}\right] = 1 - \alpha.$$

So the $100(1-\alpha)\%$ of confidential interval of β_0 is

$$\left[b_0 - z_{\alpha/2} \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)}, b_0 + z_{\alpha/2} \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)} \right].$$

Case 2: When σ^2 is unknown:

When σ^2 is unknown, then the following statistic is constructed

$$t_0 = \frac{b_0 - \beta_{00}}{\sqrt{\frac{SS_{res}}{n-2} \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)}}$$

which follows a t -distribution with $(n-2)$ degrees of freedom, i.e., t_{n-2} when H_0 is true.

A decision rule to test $H_1: \beta_0 \neq \beta_{00}$ is as follows:

Reject H_0 whenever $|t_0| > t_{n-2, \alpha/2}$

where $t_{n-2, \alpha/2}$ is the $\alpha/2$ percentage point of the t -distribution with $(n-2)$ degrees of freedom. Similarly, the decision rule for one-sided alternative hypothesis can also be framed.

The $100(1-\alpha)\%$ confidence interval of β_0 can be obtained as follows:

Consider

$$\begin{aligned} P\left[t_{n-2,\alpha/2} \leq t_0 \leq t_{n-2,\alpha/2}\right] &= 1 - \alpha \\ P\left[t_{n-2,\alpha/2} \leq \frac{b_0 - \beta_0}{\sqrt{\frac{SS_{res}}{n-2} \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)}} \leq t_{n-2,\alpha/2}\right] &= 1 - \alpha \\ P\left[b_0 - t_{n-2,\alpha/2} \sqrt{\frac{SS_{res}}{n-2} \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)} \leq \beta_0 \leq b_0 + t_{n-2,\alpha/2} \sqrt{\frac{SS_{res}}{n-2} \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)}\right] &= 1 - \alpha. \end{aligned}$$

So $100(1-\alpha)\%$ confidence interval for β_0 is

$$\left[b_0 - t_{n-2,\alpha/2} \sqrt{\frac{SS_{res}}{n-2} \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)}, b_0 + t_{n-2,\alpha/2} \sqrt{\frac{SS_{res}}{n-2} \left(\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}}\right)} \right].$$

Test of hypothesis for σ^2

We have considered two types of test statistics for testing the hypothesis about the intercept term and slope parameter- when σ^2 is known and when σ^2 is unknown. While dealing with the case of known σ^2 , the value of σ^2 is known from some external sources like past experience, long association of the experimenter with the experiment, past studies etc. In such situations, the experimenter would like to test the hypothesis like $H_0: \sigma^2 = \sigma_0^2$ against $H_1: \sigma^2 \neq \sigma_0^2$ where σ_0^2 is specified. The test statistic is based on the result

$\frac{SS_{res}}{\sigma^2} \sim \chi^2_{n-2}$. So the test statistic is

$$C_0 = \frac{SS_{res}}{\sigma_0^2} \sim \chi^2_{n-2} \text{ under } H_0.$$

The decision rule is to reject H_0 if $C_0 < \chi^2_{n-2,\alpha/2}$ or $C_0 > \chi^2_{n-2,1-\alpha/2}$.

Confidence interval for σ^2

A confidence interval for σ^2 can also be derived as follows. Since $SS_{res}/\sigma^2 \sim \chi^2_{n-2}$, thus consider

$$P\left[\chi^2_{n-2,\alpha/2} \leq \frac{SS_{res}}{\sigma^2} \leq \chi^2_{n-2,1-\alpha/2}\right] = 1 - \alpha$$

$$P\left[\frac{SS_{res}}{\chi^2_{n-2,1-\alpha/2}} \leq \sigma^2 \leq \frac{SS_{res}}{\chi^2_{n-2,\alpha/2}}\right] = 1 - \alpha.$$

The corresponding $100(1 - \alpha)\%$ confidence interval for σ^2 is

$$\left[\frac{SS_{res}}{\chi^2_{n-2,1-\alpha/2}}, \frac{SS_{res}}{\chi^2_{n-2,\alpha/2}}\right].$$

Joint confidence region for β_0 and β_1 :

A joint confidence region for β_0 and β_1 can also be found. Such a region will provide a $100(1 - \alpha)\%$ confidence that both the estimates of β_0 and β_1 are correct. Consider the centered version of the linear regression model

$$y_i = \beta_0^* + \beta_1(x_i - \bar{x}) + \varepsilon_i$$

where $\beta_0^* = \beta_0 + \beta_1 \bar{x}$. The least squares estimators of β_0^* and β_1 are

$$b_0^* = \bar{y} \text{ and } b_1 = \frac{s_{xy}}{s_{xx}},$$

respectively.

Using the results that

$$E(b_0^*) = \beta_0^*,$$

$$E(b_1) = \beta_1,$$

$$Var(b_0^*) = \frac{\sigma^2}{n},$$

$$Var(b_1) = \frac{\sigma^2}{s_{xx}}.$$

When σ^2 is known, then the statistic

$$\frac{b_0^* - \beta_0^*}{\sqrt{\frac{\sigma^2}{n}}} \sim N(0,1) \quad \text{and} \quad \frac{b_1 - \beta_1}{\sqrt{\frac{\sigma^2}{s_{xx}}}} \sim N(0,1).$$

Moreover, both statistics are independently distributed. Thus

$$\left(\frac{b_0^* - \beta_0^*}{\sqrt{\frac{\sigma^2}{n}}} \right)^2 \sim \chi_1^2 \quad \text{and} \quad \left(\frac{b_1 - \beta_1}{\sqrt{\frac{\sigma^2}{s_{xx}}}} \right)^2 \sim \chi_1^2$$

are also independently distributed because b_0^* and b_1 are independently distributed. Consequently, the sum of these two

$$\frac{n(b_0^* - \beta_0^*)^2}{\sigma^2} + \frac{s_{xx}(b_1 - \beta_1)^2}{\sigma^2} \sim \chi_2^2.$$

Since

$$\frac{SS_{res}}{\sigma^2} \sim \chi_{n-2}^2$$

and SS_{res} is independently distributed of b_0^* and b_1 , so the ratio

$$\frac{\left(\frac{n(b_0^* - \beta_0^*)^2}{\sigma^2} + \frac{s_{xx}(b_1 - \beta_1)^2}{\sigma^2} \right) / 2}{\left(\frac{SS_{res}}{\sigma^2} \right) / (n-2)} \sim F_{2,n-2}.$$

Substituting $b_0^* = b_0 + b_1 \bar{x}$ and $\beta_0^* = \beta_0 + \beta_1 \bar{x}$, we get

$$\left(\frac{n-2}{2} \right) \left[\frac{Q_f}{SS_{res}} \right]$$

where

$$Q_f = n(b_0 - \beta_0)^2 + 2 \sum_{i=1}^n x_i (b_0 - \beta_0)(b_1 - \beta_1) + \sum_{i=1}^n x_i^2 (b_1 - \beta_1)^2.$$

Since

$$P \left[\left(\frac{n-2}{2} \right) \frac{Q_f}{SS_{res}} \leq F_{2,n-2} \right] = 1 - \alpha$$

holds true for all values of β_0 and β_1 , so the $100(1-\alpha)\%$ confidence region for β_0 and β_1 is

$$\left(\frac{n-2}{2} \right) \cdot \frac{Q_f}{SS_{res}} \leq F_{2,n-2;1-\alpha}.$$

This confidence region is an ellipse which gives the $100(1-\alpha)\%$ probability that β_0 and β_1 are contained simultaneously in this ellipse.

Analysis of variance:

The technique of analysis of variance is usually used for testing the hypothesis related to equality of more than one parameters, like population means or slope parameters. It is more meaningful in case of multiple regression model when there are more than one slope parameters. This technique is discussed and illustrated here to understand the related basic concepts and fundamentals which will be used in developing the analysis of variance in the next module in multiple linear regression model where the explanatory variables are more than two.

A test statistic for testing $H_0 : \beta_1 = 0$ can also be formulated using the analysis of variance technique as follows.

On the basis of the identity

$$y_i - \hat{y}_i = (y_i - \bar{y}) - (\hat{y}_i - \bar{y}),$$

the sum of squared residuals is

$$\begin{aligned} S(b) &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^n (y_i - \bar{y})^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 - 2 \sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{y}). \end{aligned}$$

Further, consider

$$\begin{aligned} \sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{y}) &= \sum_{i=1}^n (y_i - \bar{y})b_1(x_i - \bar{x}) \\ &= b_1^2 \sum_{i=1}^n (x_i - \bar{x})^2 \\ &= \sum_{i=1}^n (\hat{y}_i - \bar{y})^2. \end{aligned}$$

Thus we have

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2.$$

The term $\sum_{i=1}^n (y_i - \bar{y})^2$ is called the **sum of squares about the mean, corrected sum of squares** of y (i.e., $SS_{corrected}$), **total sum of squares**, or s_{yy} .

The term $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ describes the deviation: observation minus predicted value, viz., the residual sum of squares, i.e., $SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$

whereas the term $\sum_{i=1}^n (\hat{y}_i - \bar{y})^2$ describes the proportion of variability explained by the regression, $SS_{reg} = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$.

If all observations y_i are located on a straight line, then in this case $\sum_{i=1}^n (y_i - \hat{y}_i)^2 = 0$ and thus $SS_{corrected} = SS_{reg}$.

Note that SS_{reg} is completely determined by b_1 and so has only one degree of freedom. The total sum of squares $s_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2$ has $(n-1)$ degrees of freedom due to constraint $\sum_{i=1}^n (y_i - \bar{y}) = 0$ and SS_{res} has $(n-2)$ degrees of freedom as it depends on the determination of b_0 and b_1 .

All sums of squares are mutually independent and distributed as χ^2_{df} with df degrees of freedom if the errors are normally distributed.

The mean square due to regression is

$$MS_{reg} = \frac{SS_{reg}}{1}$$

and mean square due to residuals is

$$MSE = \frac{SS_{res}}{n-2}.$$

The test statistic for testing $H_0 : \beta_1 = 0$ is

$$F_0 = \frac{MS_{reg}}{MSE}.$$

If $H_0 : \beta_1 = 0$ is true, then MS_{reg} and MSE are independently distributed and thus

$$F_0 \sim F_{1, n-2}.$$

The decision rule for $H_1 : \beta_1 \neq 0$ is to reject H_0 if

$$F_0 > F_{1,n-2;1-\alpha}$$

at α level of significance. The test procedure can be described in an Analysis of variance table.

Analysis of variance for testing $H_0 : \beta_1 = 0$

Source of variation	Sum of squares	Degrees of freedom	Mean square	F
Regression	SS_{reg}	1	MS_{reg}	MS_{reg} / MSE
Residual	SS_{res}	$n - 2$	MSE	
Total	s_{yy}	$n - 1$		

Some other forms of SS_{reg} , SS_{res} and s_{yy} can be derived as follows:

The sample correlation coefficient then may be written as

$$r_{xy} = \frac{s_{xy}}{\sqrt{s_{xx}} \sqrt{s_{yy}}}.$$

Moreover, we have

$$b_1 = \frac{s_{xy}}{s_{xx}} = r_{xy} \sqrt{\frac{s_{yy}}{s_{xx}}}.$$

The estimator of σ^2 in this case may be expressed as

$$\begin{aligned} s^2 &= \frac{1}{n-2} \sum_{i=1}^n e_i^2 \\ &= \frac{1}{n-2} SS_{res}. \end{aligned}$$

Various alternative formulations for SS_{res} are in use as well:

$$\begin{aligned} SS_{res} &= \sum_{i=1}^n [y_i - (b_0 + b_1 x_i)]^2 \\ &= \sum_{i=1}^n [(y_i - \bar{y}) - b_1(x_i - \bar{x})]^2 \\ &= s_{yy} + b_1^2 s_{xx} - 2b_1 s_{xy} \\ &= s_{yy} - b_1^2 s_{xx} \\ &= s_{yy} - \frac{(s_{xy})^2}{s_{xx}}. \end{aligned}$$

Using this result, we find that

$$SS_{corrected} = s_{yy}$$

and

$$\begin{aligned} SS_{reg} &= s_{yy} - SS_{res} \\ &= \frac{(s_{xy})^2}{s_{xx}} \\ &= b_1^2 s_{xx} \\ &= b_1 s_{xy}. \end{aligned}$$

Goodness of fit of regression

It can be noted that a fitted model can be said to be good when residuals are small. Since SS_{res} is based on residuals, so a measure of the quality of a fitted model can be based on SS_{res} . When the intercept term is present in the model, a measure of goodness of fit of the model is given by

$$R^2 = 1 - \frac{SS_{res}}{s_{yy}} = \frac{SS_{reg}}{s_{yy}}.$$

This is known as the **coefficient of determination**. This measure is based on the concept that how much variation in y 's stated by s_{yy} is explainable by SS_{reg} and how much unexplainable part is contained in SS_{res} . The ratio SS_{reg} / s_{yy} describes the proportion of variability that is explained by regression in relation to the total variability of y . The ratio SS_{res} / s_{yy} describes the proportion of variability that is not covered by the regression.

It can be seen that

$$R^2 = r_{xy}^2$$

where r_{xy} is the simple correlation coefficient between x and y . Clearly $0 \leq R^2 \leq 1$, so a value of R^2 closer to one indicates the better fit and value of R^2 closer to zero indicates the poor fit.

Prediction of values of study variable

An important use of linear regression modeling is to predict the average and actual values of the study variable. The term prediction of the value of study variable corresponds to knowing the value of $E(y)$ (in case of average value) and value of y (in case of actual value) for a given value of the explanatory variable. We consider both cases.

Case 1: Prediction of average value

Under the linear regression model, $y = \beta_0 + \beta_1 x + \varepsilon$, the fitted model is $y = b_0 + b_1 x$ where b_0 and b_1 are the OLS estimators of β_0 and β_1 respectively.

Suppose we want to predict the value of $E(y)$ for a given value of $x = x_0$. Then the predictor is given by

$$\widehat{E(y|x_0)} = \hat{\mu}_{y|x_0} = b_0 + b_1 x_0.$$

Predictive bias

Then the prediction error is given as

$$\begin{aligned}\hat{\mu}_{y|x_0} - E(y) &= b_0 + b_1 x_0 - E(\beta_0 + \beta_1 x_0 + \varepsilon) \\ &= b_0 + b_1 x_0 - (\beta_0 + \beta_1 x_0) \\ &= (b_0 - \beta_0) + (b_1 - \beta_1)x_0.\end{aligned}$$

Then

$$\begin{aligned}E[\hat{\mu}_{y|x_0} - E(y)] &= E(b_0 - \beta_0) + E(b_1 - \beta_1)x_0 \\ &= 0 + 0 = 0\end{aligned}$$

Thus the predictor $\hat{\mu}_{y|x_0}$ is an unbiased predictor of $E(y)$.

Predictive variance:

The predictive variance of $\hat{\mu}_{y|x_0}$ is

$$\begin{aligned}PV(\hat{\mu}_{y|x_0}) &= Var(b_0 + b_1 x_0) \\ &= Var[\bar{y} + b_1(x_0 - \bar{x})] \\ &= Var(\bar{y}) + (x_0 - \bar{x})^2 Var(b_1) + 2(x_0 - \bar{x})Cov(\bar{y}, b_1) \\ &= \frac{\sigma^2}{n} + \frac{\sigma^2(x_0 - \bar{x})^2}{s_{xx}} + 0 \\ &= \sigma^2 \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right].\end{aligned}$$

Estimate of predictive variance

The predictive variance can be estimated by substituting σ^2 by $\hat{\sigma}^2 = MSE$ as

$$\begin{aligned}\widehat{PV}(\hat{\mu}_{y|x_0}) &= \hat{\sigma}^2 \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right] \\ &= MSE \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right].\end{aligned}$$

Prediction interval estimation:

The $100(1-\alpha)\%$ prediction interval for $E(y/x_0)$ is obtained as follows:

The predictor $\hat{\mu}_{y|x_0}$ is a linear combination of normally distributed random variables, so it is also normally distributed as

$$\hat{\mu}_{y|x_0} \sim N\left(\beta_0 + \beta_1 x_0, PV\left(\hat{\mu}_{y|x_0}\right)\right).$$

So if σ^2 is known, then the distribution of

$$\frac{\hat{\mu}_{y|x_0} - E(y|x_0)}{\sqrt{PV(\hat{\mu}_{y|x_0})}}$$

is $N(0,1)$. So the $100(1-\alpha)\%$ prediction interval is obtained as

$$P\left[-z_{\alpha/2} \leq \frac{\hat{\mu}_{y|x_0} - E(y|x_0)}{\sqrt{PV(\hat{\mu}_{y|x_0})}} \leq z_{\alpha/2}\right] = 1 - \alpha$$

which gives the prediction interval for $E(y/x_0)$ as

$$\left[\hat{\mu}_{y|x_0} - z_{\alpha/2} \sqrt{\sigma^2 \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right]}, \hat{\mu}_{y|x_0} + z_{\alpha/2} \sqrt{\sigma^2 \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right]}\right].$$

When σ^2 is unknown, it is replaced by $\hat{\sigma}^2 = MSE$ and in this case the sampling distribution of

$$\frac{\hat{\mu}_{y|x_0} - E(y|x_0)}{\sqrt{MSE \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right]}}$$

is t -distribution with $(n-2)$ degrees of freedom, i.e., t_{n-2} .

The $100(1-\alpha)\%$ prediction interval in this case is

$$P\left[-t_{\alpha/2,n-2} \leq \frac{\hat{\mu}_{y|x_0} - E(y|x_0)}{\sqrt{MSE \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right]}} \leq t_{\alpha/2,n-2}\right] = 1 - \alpha.$$

which gives the prediction interval as

$$\left[\hat{\mu}_{y|x_0} - t_{\alpha/2,n-2} \sqrt{MSE \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right]}, \hat{\mu}_{y|x_0} + t_{\alpha/2,n-2} \sqrt{MSE \left[\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right]}\right].$$

Note that the width of the prediction interval $E(y|x_0)$ is a function of x_0 . The interval width is minimum for $x_0 = \bar{x}$ and widens as $|x_0 - \bar{x}|$ increases. This is also expected as the best estimates of y to be made at x -values lie near the center of the data and the precision of estimation to deteriorate as we move to the boundary of the x -space.

Case 2: Prediction of actual value

If x_0 is the value of the explanatory variable, then the actual value predictor for y is

$$\hat{y}_0 = b_0 + b_1 x_0.$$

The true value of y in the prediction period is given by $y_0 = \beta_0 + \beta_1 x_0 + \varepsilon_0$ where ε_0 indicates the value that would be drawn from the distribution of random error in the prediction period. Note that the form of predictor is the same as of average value predictor, but its predictive error and other properties are different. This is the **dual nature of predictor**.

Predictive bias:

The predictive error of \hat{y}_0 is given by

$$\begin{aligned}\hat{y}_0 - y_0 &= b_0 + b_1 x_0 - (\beta_0 + \beta_1 x_0 + \varepsilon_0) \\ &= (b_0 - \beta_0) + (b_1 - \beta_1)x_0 - \varepsilon.\end{aligned}$$

Thus, we find that

$$\begin{aligned}E(\hat{y}_0 - y_0) &= E(b_0 - \beta_0) + E(b_1 - \beta_1)x_0 - E(\varepsilon_0) \\ &= 0 + 0 + 0 = 0\end{aligned}$$

which implies that \hat{y}_0 is an unbiased predictor of y_0 .

Predictive variance

Because the future observation y_0 is independent of \hat{y}_0 , the predictive variance of \hat{y}_0 is

$$\begin{aligned}
 PV(\hat{y}_0) &= E(\hat{y}_0 - y_0)^2 \\
 &= E[(b_0 - \beta_0) + (x_0 - \bar{x})(b_1 - \beta_1) + (b_1 - \beta_1)\bar{x} - \varepsilon_0]^2 \\
 &= Var(b_0) + (x_0 - \bar{x})^2 Var(b_1) + \bar{x}^2 Var(b_1) + Var(\varepsilon_0) + 2(x_0 - \bar{x})Cov(b_0, b_1) + 2\bar{x}Cov(b_0, b_1) + 2(x_0 - \bar{x})Var(b_1) \\
 &\quad [\text{rest of the terms are 0 assuming the independence of } \varepsilon_0 \text{ with } \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n] \\
 &= Var(b_0) + [(x_0 - \bar{x})^2 + \bar{x}^2 + 2(x_0 - \bar{x})]Var(b_1) + Var(\varepsilon) + 2[(x_0 - \bar{x}) + 2\bar{x}]Cov(b_0, b_1) \\
 &= Var(b_0) + x_0^2 Var(b_1) + Var(\varepsilon_0) + 2x_0 Cov(b_0, b_1) \\
 &= \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{s_{xx}} \right] + x_0^2 \frac{\sigma^2}{s_{xx}} + \sigma^2 - 2x_0 \frac{\bar{x}\sigma^2}{s_{xx}} \\
 &= \sigma^2 \left[1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right].
 \end{aligned}$$

Estimate of predictive variance

The estimate of predictive variance can be obtained by replacing σ^2 by its estimate $\hat{\sigma}^2 = MSE$ as

$$\begin{aligned}
 \widehat{PV}(\hat{y}_0) &= \hat{\sigma}^2 \left[1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right] \\
 &= MSE \left[1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right].
 \end{aligned}$$

Prediction interval:

If σ^2 is known, then the distribution of

$$\frac{\hat{y}_0 - y_0}{\sqrt{PV(\hat{y}_0)}}$$

is $N(0,1)$. So the $100(1-\alpha)\%$ prediction interval is obtained as

$$P \left[-z_{\alpha/2} \leq \frac{\hat{y}_0 - y_0}{\sqrt{PV(\hat{y}_0)}} \leq z_{\alpha/2} \right] = 1 - \alpha$$

which gives the prediction interval for y_0 as

$$\left[\hat{y}_0 - z_{\alpha/2} \sqrt{\sigma^2 \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right)}, \hat{y}_0 + z_{\alpha/2} \sqrt{\sigma^2 \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right)} \right].$$

When σ^2 is unknown, then

$$\frac{\hat{y}_0 - y_0}{\sqrt{\widehat{PV}(\hat{y}_0)}}$$

follows a t -distribution with $(n-2)$ degrees of freedom. The $100(1-\alpha)\%$ prediction interval for \hat{y}_0 in this case is obtained as

$$P\left[-t_{\alpha/2,n-2} \leq \frac{\hat{y}_0 - y_0}{\sqrt{\widehat{PV}(\hat{y}_0)}} \leq t_{\alpha/2,n-2}\right] = 1-\alpha$$

which gives the prediction interval

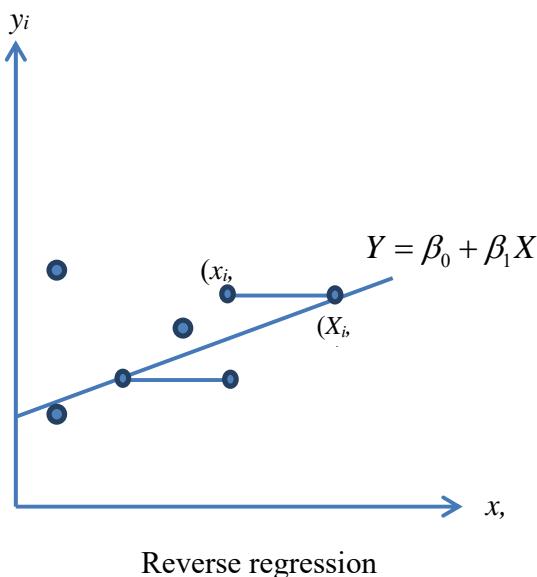
$$\left[\hat{y}_0 - t_{\alpha/2,n-2} \sqrt{MSE \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right)}, \hat{y}_0 + t_{\alpha/2,n-2} \sqrt{MSE \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{s_{xx}} \right)} \right].$$

The prediction interval is of minimum width at $x_0 = \bar{x}$ and widens as $|x_0 - \bar{x}|$ increases.

The prediction interval for \hat{y}_0 is wider than the prediction interval for $\hat{\mu}_{y/x_0}$ because the prediction interval for \hat{y}_0 depends on both the error from the fitted model as well as the error associated with the future observations.

Reverse regression method

The reverse (or inverse) regression approach minimizes the sum of squares of horizontal distances between the observed data points and the line in the following scatter diagram to obtain the estimates of regression parameters.



The reverse regression has been advocated in the analysis of gender (or race) discrimination in salaries. For example, if y denotes salary and x denotes qualifications, and we are interested in determining if there is gender discrimination in salaries, we can ask:

“Whether men and women with the same qualifications (value of x) are getting the same salaries (value of y). This question is answered by the direct regression.”

Alternatively, we can ask:

“Whether men and women with the same salaries (value of y) have the same qualifications (value of x). This question is answered by the reverse regression, i.e., regression of x on y . ”

The regression equation in case of reverse regression can be written as

$$x_i = \beta_0^* + \beta_1^* y_i + \delta_i \quad (i=1,2,\dots,n)$$

where δ_i ’s are the associated random error components and satisfy the assumptions as in the case of the usual simple linear regression model. The reverse regression estimates $\hat{\beta}_{OR}$ of β_0^* and $\hat{\beta}_{1R}$ of β_1^* for the model are obtained by interchanging the x and y in the direct regression estimators of β_0 and β_1 . The estimates are obtained as

$$\hat{\beta}_{OR} = \bar{x} - \hat{\beta}_{1R} \bar{y}$$

and

$$\hat{\beta}_{1R} = \frac{s_{yy}}{s_{xy}}$$

for β_0 and β_1 respectively. The residual sum of squares in this case is

$$SS_{res}^* = s_{xx} - \frac{s_{xy}^2}{s_{yy}}.$$

Note that

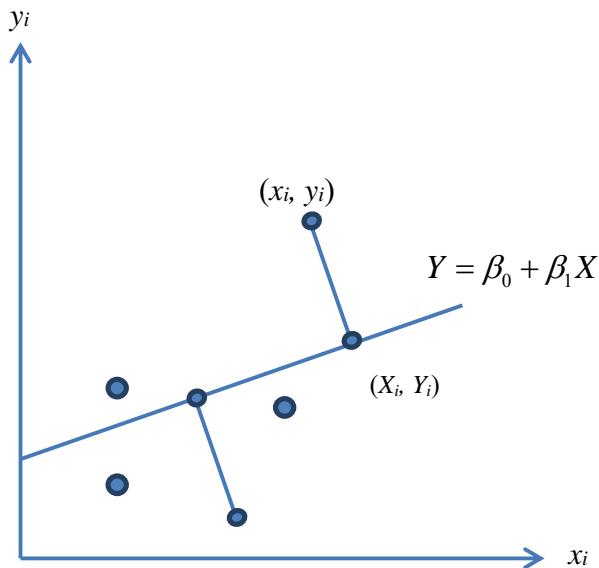
$$\hat{\beta}_{1R} b_1 = \frac{s_{xy}^2}{s_{xx} s_{yy}} = r_{xy}^2$$

where b_1 is the direct regression estimator of the slope parameter and r_{xy} is the correlation coefficient between x and y . Hence if r_{xy}^2 is close to 1, the two regression lines will be close to each other.

An important application of the reverse regression method is in solving the calibration problem.

Orthogonal regression method (or major axis regression method)

The direct and reverse regression methods of estimation assume that the errors in the observations are either in x -direction or y -direction. In other words, the errors can be either in the dependent variable or independent variable. There can be situations when uncertainties are involved in dependent and independent variables both. In such situations, the orthogonal regression is more appropriate. In order to take care of errors in both the directions, the least-squares principle in orthogonal regression minimizes the squared perpendicular distance between the observed data points and the line in the following scatter diagram to obtain the estimates of regression coefficients. This is also known as the **major axis regression method**. The estimates obtained are called **orthogonal regression estimates** or **major axis regression estimates** of regression coefficients.



Orthogonal or major axis regression

If we assume that the regression line to be fitted is $Y_i = \beta_0 + \beta_1 X_i$, then it is expected that all the observations (x_i, y_i) , $i = 1, 2, \dots, n$ lie on this line. But these points deviate from the line, and in such a case, the squared perpendicular distance of observed data (x_i, y_i) ($i = 1, 2, \dots, n$) from the line is given by

$$d_i^2 = (X_i - x_i)^2 + (Y_i - y_i)^2$$

where (X_i, Y_i) denotes the i^{th} pair of observation without any error which lies on the line.

The objective is to minimize the sum of squared perpendicular distances given by $\sum_{i=1}^n d_i^2$ to obtain the estimates of β_0 and β_1 . The observations (x_i, y_i) ($i=1, 2, \dots, n$) are expected to lie on the line

$$Y_i = \beta_0 + \beta_1 X_i,$$

so let

$$E_i = Y_i - \beta_0 - \beta_1 X_i = 0.$$

The regression coefficients are obtained by minimizing $\sum_{i=1}^n d_i^2$ under the constraints E_i 's using the Lagrangian's multiplier method. The Lagrangian function is

$$L_0 = \sum_{i=1}^n d_i^2 - 2 \sum_{i=1}^n \lambda_i E_i$$

where $\lambda_1, \dots, \lambda_n$ are the Lagrangian multipliers. The set of equations are obtained by setting

$$\frac{\partial L_0}{\partial X_i} = 0, \frac{\partial L_0}{\partial Y_i} = 0, \frac{\partial L_0}{\partial \beta_0} = 0 \text{ and } \frac{\partial L_0}{\partial \beta_1} = 0 \quad (i=1, 2, \dots, n).$$

Thus we find

$$\frac{\partial L_0}{\partial X_i} = (X_i - x_i) + \lambda_i \beta_1 = 0$$

$$\frac{\partial L_0}{\partial Y_i} = (Y_i - y_i) - \lambda_i = 0$$

$$\frac{\partial L_0}{\partial \beta_0} = \sum_{i=1}^n \lambda_i = 0$$

$$\frac{\partial L_0}{\partial \beta_1} = \sum_{i=1}^n \lambda_i X_i = 0.$$

Since

$$X_i = x_i - \lambda_i \beta_1$$

$$Y_i = y_i + \lambda_i,$$

so substituting these values is ε_i , we obtain

$$\begin{aligned} E_i &= (y_i + \lambda_i) - \beta_0 - \beta_1(x_i - \lambda_i \beta_1) = 0 \\ \Rightarrow \lambda_i &= \frac{\beta_0 + \beta_1 x_i - y_i}{1 + \beta_1^2}. \end{aligned}$$

Also using this λ_i in the equation $\sum_{i=1}^n \lambda_i = 0$, we get

$$\frac{\sum_{i=1}^n (\beta_0 + \beta_1 x_i - y_i)}{1 + \beta_1^2} = 0$$

and using $(X_i - x_i) + \lambda_i \beta_1 = 0$ and $\sum_{i=1}^n \lambda_i X_i = 0$, we get

$$\sum_{i=1}^n \lambda_i (x_i - \lambda_i \beta_1) = 0.$$

Substituting λ_i in this equation, we get

$$\frac{\sum_{i=1}^n (\beta_0 x_i + \beta_1 x_i^2 - y_i x_i)}{(1 + \beta_1^2)} - \frac{\beta_1 (\beta_0 + \beta_1 x_i - y_i)^2}{(1 + \beta_1^2)^2} = 0. \quad (1)$$

Using λ_i in the equation and using the equation $\sum_{i=1}^n \lambda_i = 0$, we solve

$$\frac{\sum_{i=1}^n (\beta_0 + \beta_1 x_i - y_i)}{1 + \beta_1^2} = 0.$$

The solution provides an orthogonal regression estimate of β_0 as

$$\hat{\beta}_{0OR} = \bar{y} - \hat{\beta}_{1OR} \bar{x}$$

where $\hat{\beta}_{1OR}$ is an orthogonal regression estimate of β_1 .

Now, substituting $\hat{\beta}_{0OR}$ in equation (1), we get

$$\sum_{i=1}^n (1 + \beta_1^2) [\bar{y} x_i - \beta_1 \bar{x} x_i + \beta_1 x_i^2 - x_i y_i] - \beta_1 \sum_{i=1}^n (\bar{y} - \beta_1 \bar{x} + \beta_1 x_i - y_i)^2 = 0$$

or

$$(1 + \beta_1^2) \sum_{i=1}^n x_i [y_i - \bar{y} - \beta_1 (x_i - \bar{x})] + \beta_1 \sum_{i=1}^n [-(y_i - \bar{y}) + \beta_1 (x_i - \bar{x})]^2 = 0$$

or

$$(1 + \beta_1^2) \sum_{i=1}^n (u_i + \bar{x})(v_i - \beta_1 u_i) + \beta_1 \sum_{i=1}^n (-v_i + \beta_1 u_i)^2 = 0$$

where

$$u_i = x_i - \bar{x},$$

$$v_i = y_i - \bar{y}.$$

Since $\sum_{i=1}^n u_i = \sum_{i=1}^n v_i = 0$, so

$$\sum_{i=1}^n [\beta_1^2 u_i v_i + \beta_1 (u_i^2 - v_i^2) - u_i v_i] = 0$$

or

$$\beta_1^2 s_{xy} + \beta_1 (s_{xx} - s_{yy}) - s_{xy} = 0.$$

Solving this quadratic equation provides the orthogonal regression estimate of β_1 as

$$\hat{\beta}_{1OR} = \frac{(s_{yy} - s_{xx}) + sign(s_{xy}) \sqrt{(s_{xx} - s_{yy})^2 + 4s_{xy}^2}}{2s_{xy}}$$

where $sign(s_{xy})$ denotes the sign of s_{xy} which can be positive or negative. So

$$sign(s_{xy}) = \begin{cases} 1 & \text{if } s_{xy} > 0 \\ -1 & \text{if } s_{xy} < 0. \end{cases}$$

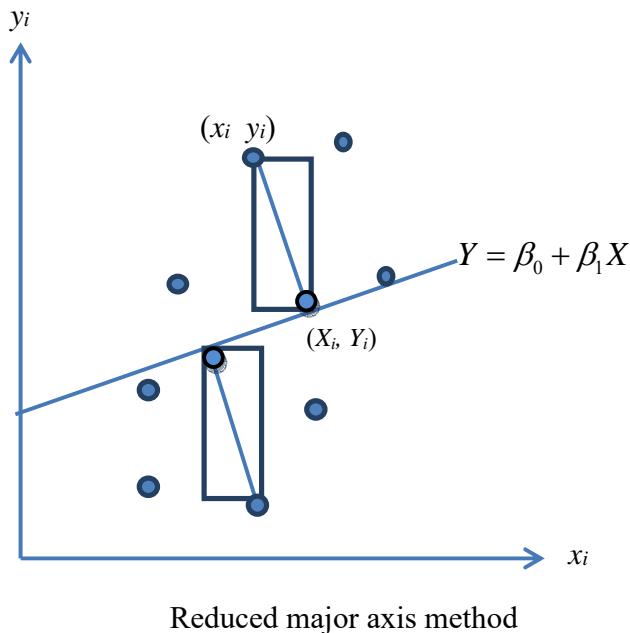
Notice that this gives two solutions for $\hat{\beta}_{1OR}$. We choose the solution which minimizes $\sum_{i=1}^n d_i^2$. The other

solution maximizes $\sum_{i=1}^n d_i^2$ and is in the direction perpendicular to the optimal solution. The optimal solution

can be chosen with the sign of s_{xy} .

Reduced major axis regression method:

The direct, reverse and orthogonal methods of estimation minimize the errors in a particular direction which is usually the distance between the observed data points and the line in the scatter diagram. Alternatively, one can consider the area extended by the data points in a certain neighbourhood and instead of distances, the area of rectangles defined between the corresponding observed data point and the nearest point on the line in the following scatter diagram can also be minimized. Such an approach is more appropriate when the uncertainties are present in the study and explanatory variables both. This approach is termed as reduced major axis regression.



Suppose the regression line is $Y_i = \beta_0 + \beta_1 X_i$ on which all the observed points are expected to lie. Suppose the points (x_i, y_i) , $i = 1, 2, \dots, n$ are observed which lie away from the line. The area of rectangle extended between the i^{th} observed data point and the line is

$$A_i = (X_i - x_i)(Y_i - y_i) \quad (i = 1, 2, \dots, n)$$

where (X_i, Y_i) denotes the i^{th} pair of observation without any error which lies on the line.

The total area extended by n data points is

$$\sum_{i=1}^n A_i = \sum_{i=1}^n (X_i - x_i)(Y_i - y_i).$$

All observed data points (x_i, y_i) , $(i = 1, 2, \dots, n)$ are expected to lie on the line

$$Y_i = \beta_0 + \beta_1 X_i$$

and let

$$E_i^* = Y_i - \beta_0 - \beta_1 X_i = 0.$$

So now the objective is to minimize the sum of areas under the constraints E_i^* to obtain the reduced major axis estimates of regression coefficients. Using the Lagrangian multiplier method, the Lagrangian function is

$$\begin{aligned} L_R &= \sum_{i=1}^n A_i - \sum_{i=1}^n \mu_i E_i^* \\ &= \sum_{i=1}^n (X_i - x_i)(Y_i - y_i) - \sum_{i=1}^n \mu_i E_i^* \end{aligned}$$

where μ_1, \dots, μ_n are the Lagrangian multipliers. The set of equations are obtained by setting

$$\frac{\partial L_R}{\partial X_i} = 0, \frac{\partial L_R}{\partial Y_i} = 0, \frac{\partial L_R}{\partial \beta_0} = 0, \frac{\partial L_R}{\partial \beta_1} = 0 \quad (i = 1, 2, \dots, n).$$

Thus

$$\frac{\partial L_R}{\partial X_i} = (Y_i - y_i) + \beta_1 \mu_i = 0$$

$$\frac{\partial L_R}{\partial Y_i} = (X_i - x_i) - \mu_i = 0$$

$$\frac{\partial L_R}{\partial \beta_0} = \sum_{i=1}^n \mu_i = 0$$

$$\frac{\partial L_R}{\partial \beta_1} = \sum_{i=1}^n \mu_i X_i = 0.$$

Now

$$X_i = x_i + \mu_i$$

$$Y_i = y_i - \beta_1 \mu_i$$

$$\beta_0 + \beta_1 X_i = y_i - \beta_1 \mu_i$$

$$\beta_0 + \beta_1 (x_i + \mu_i) = y_i - \beta_1 \mu_i$$

$$\Rightarrow \mu_i = \frac{y_i - \beta_0 - \beta_1 x_i}{2\beta_1}.$$

Substituting μ_i in $\sum_{i=1}^n \mu_i = 0$, the reduced major axis regression estimate of β_0 is obtained as

$$\hat{\beta}_{0RM} = \bar{y} - \hat{\beta}_{1RM} \bar{x}$$

where $\hat{\beta}_{1RM}$ is the reduced major axis regression estimate of β_1 . Using $X_i = x_i + \mu_i$, μ_i and $\hat{\beta}_{0RM}$ in

$$\sum_{i=1}^n \mu_i X_i = 0, \text{ we get}$$

$$\sum_{i=1}^n \left(\frac{y_i - \bar{y} + \beta_1 \bar{x} - \beta_1 x_i}{2\beta_1} \right) \left(x_i - \frac{y_i - \bar{y} + \beta_1 \bar{x} - \beta_1 x_i}{2\beta_1} \right) = 0.$$

Let $u_i = x_i - \bar{x}$ and $v_i = y_i - \bar{y}$, then this equation can be re-expressed as

$$\sum_{i=1}^n (v_i - \beta_1 u_i)(v_i + \beta_1 u_i + 2\beta_1 \bar{x}) = 0.$$

Using $\sum_{i=1}^n u_i = \sum_{i=1}^n u_i = 0$, we get

$$\sum_{i=1}^n v_i^2 - \beta_1^2 \sum_{i=1}^n u_i^2 = 0.$$

Solving this equation, the reduced major axis regression estimate of β_1 is obtained as

$$\hat{\beta}_{1RM} = sign(s_{xy}) \sqrt{\frac{s_{yy}}{s_{xx}}}$$

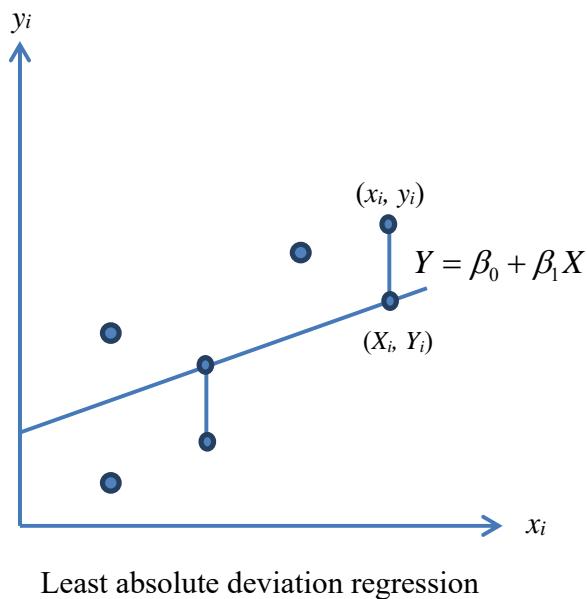
where $sign(s_{xy}) = \begin{cases} 1 & \text{if } s_{xy} > 0 \\ -1 & \text{if } s_{xy} < 0. \end{cases}$

We choose the regression estimator which has same sign as of s_{xy} .

Least absolute deviation regression method

The least-squares principle advocates the minimization of the sum of squared errors. The idea of squaring the errors is useful in place of simple errors because random errors can be positive as well as negative. So consequently their sum can be close to zero indicating that there is no error in the model and which can be misleading. Instead of the sum of random errors, the sum of absolute random errors can be considered which avoids the problem due to positive and negative random errors.

In the method of least squares, the estimates of the parameters β_0 and β_1 in the model $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$. ($i = 1, 2, \dots, n$) are chosen such that the sum of squares of deviations $\sum_{i=1}^n \varepsilon_i^2$ is minimum. In the method of least absolute deviation (LAD) regression, the parameters β_0 and β_1 are estimated such that the sum of absolute deviations $\sum_{i=1}^n |\varepsilon_i|$ is minimum. It minimizes the absolute vertical sum of errors as in the following scatter diagram:



The LAD estimates $\hat{\beta}_{0L}$ and $\hat{\beta}_{1L}$ are the estimates of β_0 and β_1 , respectively which minimize

$$LAD(\beta_0, \beta_1) = \sum_{i=1}^n |y_i - \beta_0 - \beta_1 x_i|$$

for the given observations (x_i, y_i) ($i = 1, 2, \dots, n$).

Conceptually, LAD procedure is more straightforward than OLS procedure because $|e|$ (absolute residuals) is a more straightforward measure of the size of the residual than e^2 (squared residuals). The LAD regression estimates of β_0 and β_1 are not available in closed form. Instead, they can be obtained numerically based on algorithms. Moreover, this creates the problems of non-uniqueness and degeneracy in the estimates. The concept of non-uniqueness relates to that more than one best line pass through a data point. The degeneracy concept describes that the best line through a data point also passes through more than one other data points. The non-uniqueness and degeneracy concepts are used in algorithms to judge the

quality of the estimates. The algorithm for finding the estimators generally proceeds in steps. At each step, the best line is found that passes through a given data point. The best line always passes through another data point, and this data point is used in the next step. When there is non-uniqueness, then there is more than one best line. When there is degeneracy, then the best line passes through more than one other data point. When either of the problems is present, then there is more than one choice for the data point to be used in the next step and the algorithm may go around in circles or make a wrong choice of the LAD regression line. The exact tests of hypothesis and confidence intervals for the LAD regression estimates can not be derived analytically. Instead, they are derived analogously to the tests of hypothesis and confidence intervals related to ordinary least squares estimates.

Estimation of parameters when X is stochastic

In a usual linear regression model, the study variable is supposed to be random and explanatory variables are assumed to be fixed. In practice, there may be situations in which the explanatory variable also becomes random.

Suppose both dependent and independent variables are stochastic in the simple linear regression model

$$y = \beta_0 + \beta_1 X + \varepsilon$$

where ε is the associated random error component. The observations (x_i, y_i) , $i = 1, 2, \dots, n$ are assumed to be jointly distributed. Then the statistical inferences can be drawn in such cases which are conditional on X .

Assume the joint distribution of X and y to be bivariate normal $N(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \rho)$ where μ_x and μ_y are the means of X and y ; σ_x^2 and σ_y^2 are the variances of X and y ; and ρ is the correlation coefficient between X and y . Then the conditional distribution of y given $X = x$ is the univariate normal conditional mean

$$E(y | X = x) = \mu_{y|x} = \beta_0 + \beta_1 x$$

and the conditional variance of y given $X = x$ is

$$\text{Var}(y | X = x) = \sigma_{y|x}^2 = \sigma_y^2(1 - \rho^2)$$

where

$$\beta_0 = \mu_y - \mu_x \beta_1$$

and

$$\beta_1 = \frac{\sigma_y}{\sigma_x} \rho.$$

When both X and y are stochastic, then the problem of estimation of parameters can be reformulated as follows. Consider a conditional random variable $y|X=x$ having a normal distribution with mean as conditional mean $\mu_{y|x}$ and variance as conditional variance $Var(y|X=x)=\sigma_{y|x}^2$. Obtain n independently distributed observation $y_i|x_i, i=1,2,\dots,n$ from $N(\mu_{y|x}, \sigma_{y|x}^2)$ with nonstochastic X . Now the method of maximum likelihood can be used to estimate the parameters which yield the estimates of β_0 and β_1 as earlier in the case of nonstochastic X as

$$\tilde{b} = \bar{y} - \tilde{b}_1 \bar{x}$$

and

$$\tilde{b}_1 = \frac{s_{xy}}{s_{xx}},$$

respectively.

Moreover, the correlation coefficient

$$\rho = \frac{E(y - \mu_y)(X - \mu_x)}{\sigma_y \sigma_x}$$

can be estimated by the sample correlation coefficient

$$\begin{aligned}\hat{\rho} &= \frac{\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \\ &= \frac{s_{xy}}{\sqrt{s_{xx}} \sqrt{s_{yy}}} \\ &= \tilde{b}_1 \sqrt{\frac{s_{xx}}{s_{yy}}}.\end{aligned}$$

Thus

$$\begin{aligned}\hat{\rho}^2 &= \tilde{b}_1^2 \frac{s_{xx}}{s_{yy}} \\ &= \tilde{b}_1 \frac{s_{xy}}{s_{yy}} \\ &= \frac{s_{yy} - \sum_{i=1}^n \hat{\epsilon}_i^2}{s_{yy}} \\ &= R^2\end{aligned}$$

which is same as the coefficient of determination. Thus R^2 has the same expression as in the case when X is fixed. Thus R^2 again measures the goodness of the fitted model even when X is stochastic.

Support Vector Machine

Contents to be covered

- Introduction to SVM
- Concept of maximum margin hyperplane
- Linear SVM
 1. Calculation of MMH
 2. Learning a linear SVM
 3. Classifying a test sample using linear SVM
 4. Classifying multi-class data
- Non-linear SVM
 1. Concept of non-linear data
 2. Soft-margin SVM
 3. Kernel Trick

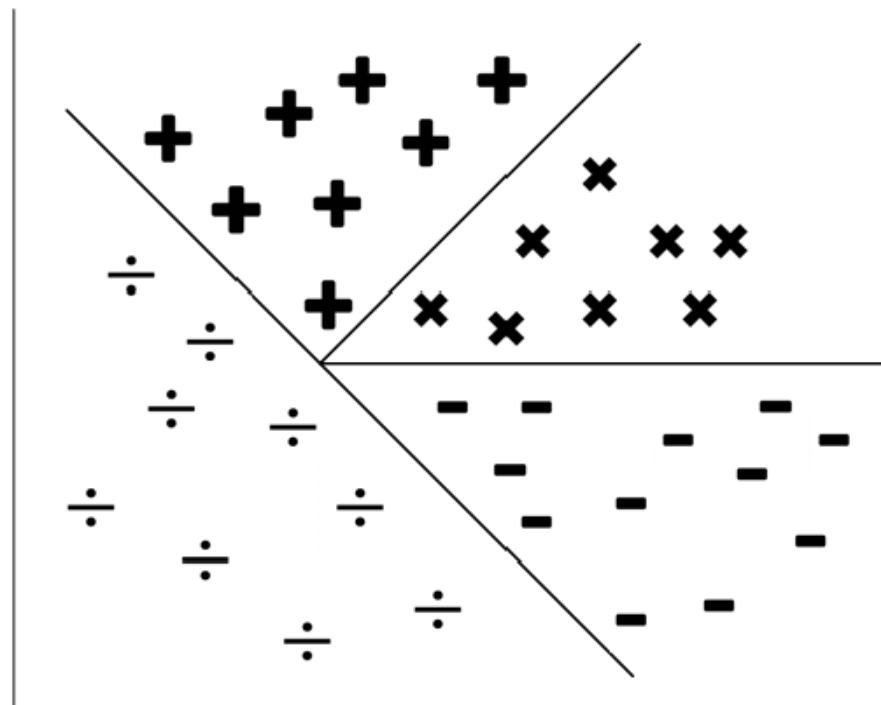
Introduction to SVM

Introduction

- A classification that has received considerable attention is support vector machine and popularly abbreviated as SVM.
- This technique has its roots in statistical learning theory (Vladimir Vapnik, 1992).
- As a task of classification, it searches for optimal hyperplane(i.e., decision boundary, see Fig. 1 in the next slide) separating the tuples of one class from another.
- SVM works well with higher dimensional data and thus avoids **dimensionality problem**.
- Although the SVM based classification (i.e., **training time**) is extremely slow, the result, is however highly accurate. Further, testing an unknown data is very fast.
- SVM is less prone to over fitting than other methods. It also facilitates compact model for classification.

Introduction

Figure 1: Decision boundary in SVM.



In this lecture, we shall discuss the following.

- Maximum margin hyperplane : a key concept in SVM.
- Linear SVM : a classification technique when training data are linearly separable.
- Non-linear SVM : a classification technique when training data are linearly non-separable.

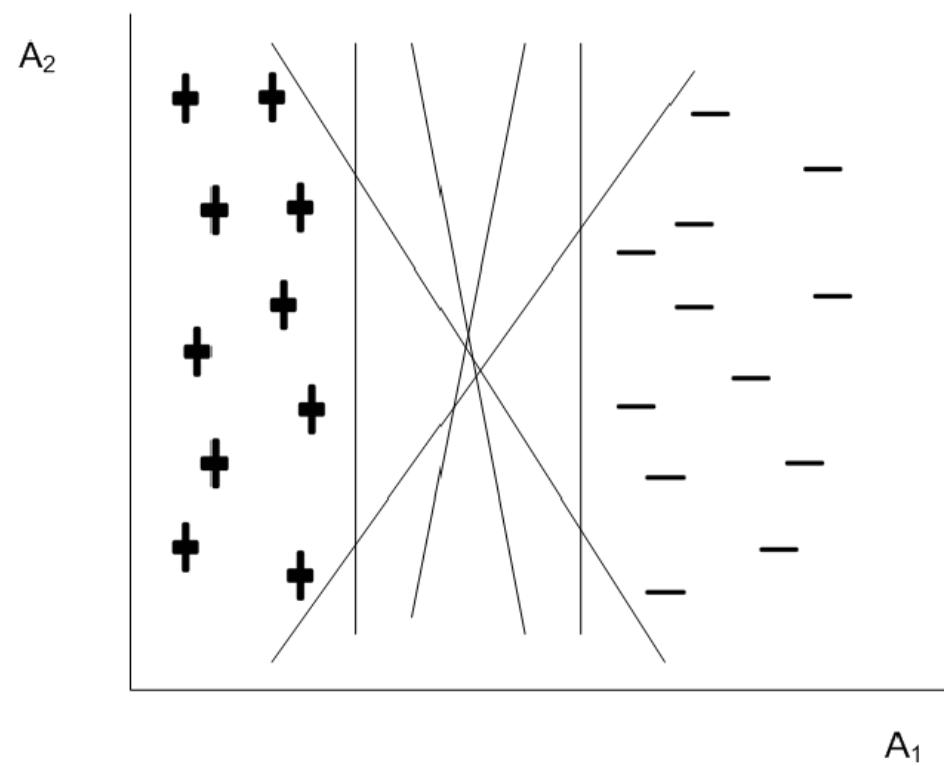
Maximum Margin Hyperplane

Maximum Margin Hyperplane

In our subsequent discussion, we shall assume a simplistic situation that given a training data $D = \{t_1, t_2, \dots, t_n\}$ with a set of n tuples, which belong to two classes either + or - and each tuple is described by two attributes say A_1, A_2 .

Maximum Margin Hyperplane

Figure 2: A 2D data linearly separable by hyperplanes



Maximum Margin Hyperplane contd...

- Figure 2 shows a plot of data in 2-D. Another simplistic assumption here is that the data is linearly separable, that is, we can find a hyperplane (in this case, it is a straight line) such that all +'s reside on one side whereas all -'s reside on other side of the hyperplane.
- From Fig. 2, it can be seen that there are an infinite number of separating lines that can be drawn. Therefore, the following two questions arise:
 1. Whether all hyperplanes are equivalent so far the classification of data is concerned?
 2. If not, which hyperplane is the best?

Maximum Margin Hyperplane contd...

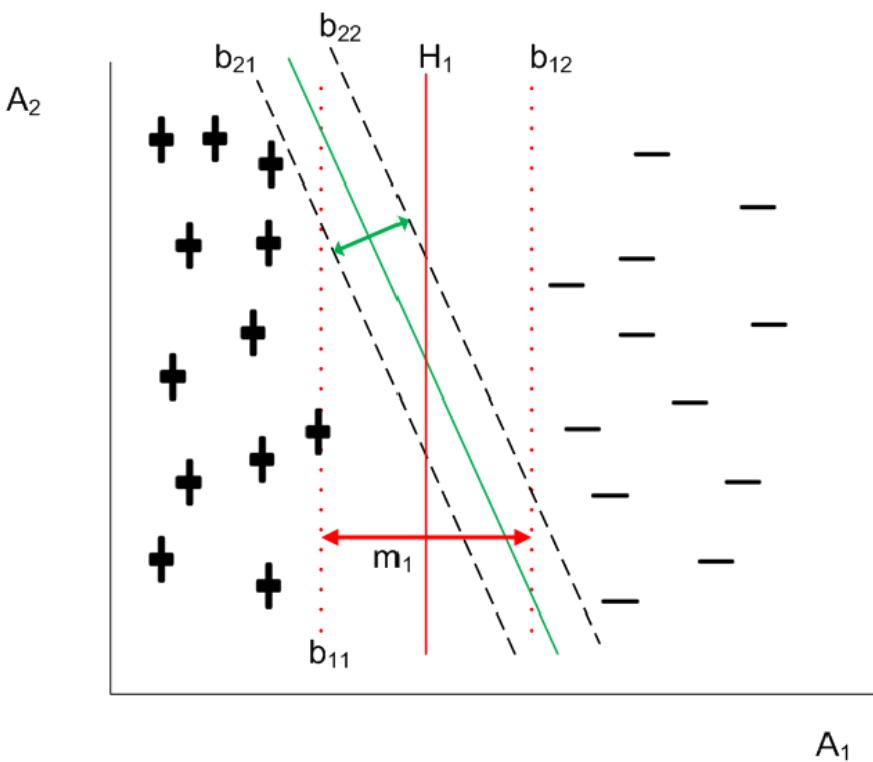
- We may note that so far the classification error is concerned (with training data), all of them are with zero error.
- However, there is no guarantee that all hyperplanes perform equally well on unseen (i.e., test) data.

Maximum Margin Hyperplane contd...

- Thus, for a good classifier it must choose one of the infinite number of hyperplanes, so that it performs better not only on training data but as well as test data.
- To illustrate how the different choices of hyperplane influence the classification error, consider any arbitrary two hyperplanes H₁ and H₂ as shown in Fig. 3.

Maximum Margin Hyperplane contd...

Figure 3: Hyperplanes with decision boundaries and their margins.



Maximum Margin Hyperplane contd...

- In Fig. 3, two hyperplanes H1 and H2 have their own boundaries called decision boundaries(denoted as b₁₁ and b₁₂ for H1 and b₂₁ and b₂₂ for H2).
- A decision boundary is a boundary which is parallel to hyperplane and touches the closest class in one side of the hyperplane.
- The distance between the two decision boundaries of a hyperplane is called the margin. So, if data is classified using Hyperplane H1, then it is with larger margin than using Hyperplane H2.
- The margin of hyperplane implies the error in classifier. In other words, the larger the margin, lower is the classification error.

Maximum Margin Hyperplane contd...

- Intuitively, the classifier that contains hyperplane with a small margin are more susceptible to model over fitting and tend to classify with weak confidence on unseen data.
- Thus during the training or learning phase, the approach would be to search for the hyperplane with maximum margin.
- Such a hyperplane is called **maximum margin hyperplane** and abbreviated as **MMH**.
- We may note the shortest distance from a hyperplane to one of its decision boundary is equal to the shortest distance from the hyperplane to the decision boundary at its other side.
- Alternatively, hyperplane is at the middle of its decision boundaries.

Linear SVM

Linear SVM

- A SVM which is used to classify data which are linearly separable is called linear SVM.
- In other words, a linear SVM searches for a hyperplane with the maximum margin.
- This is why a linear SVM is often termed as a maximal margin classifier (MMC).

Finding MMH for a Linear SVM

In the following, we shall discuss the mathematics to find the MMH given a training set.

In our discussion, we shall consider a binary classification problem consisting of n training data.

Each tuple is denoted by (X_i, Y_i) where $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ corresponds to the attribute set for the i^{th} tuple (data in m -dimensional space) and $Y_i \in \{+, -\}$ denotes its class label.

Note that choice of which class should be labeled as + or - is arbitrary.

Finding MMH for a Linear SVM

Thus, given $\{(X_i, Y_i)\}_{i=1}^n$, we are to obtain a hyperplane which separates all $X_i|_{i=1}^n$ into two sides of it (of course with maximum gap).

Before, going to a general equation of a plane in n -dimension, let us consider first, a hyperplane in 2-D plane.

Equation of a hyperplane in 2-D

Let us consider a 2-D training tuple with attributes A_1 and A_2 as $X = (x_1, x_2)$, where x_1 and x_2 are values of attributes A_1 and A_2 , respectively for X .

Equation of a plane in 2-D space can be written as

$$w_0 + w_1 x_1 + w_2 x_2 = 0 \quad [\text{e.g., } ax + by + c = 0]$$

where w_0 , w_1 , and w_2 are some constants defining the slope and intercept of the line.

Any point lying above such a hyperplane satisfies

$$w_0 + w_1 x_1 + w_2 x_2 > 0 \tag{1}$$

Equation of a hyperplane in 2-D

Similarly, any point lying below the hyperplane satisfies

$$w_0 + w_1 x_1 + w_2 x_2 < 0 \quad (2)$$

An SVM hyperplane is an n -dimensional generalization of a straight line in 2-D.

It can be visualized as a plane surface in 3-D, but it is not easy to visualize when dimensionality is greater than 3!

Equation of a hyperplane

In fact, Euclidean equation of a hyperplane in R^m is

$$w_1x_1 + w_2x_2 + \dots + w_mx_m = b \quad (3)$$

where w_i 's are the real numbers and b is a real constant (called the intercept, which can be positive or negative).

Finding a hyperplane

In matrix form, a hyperplane thus can be represented as

$$W \cdot X + b = 0 \quad (4)$$

where $W = [w_1, w_2, \dots, w_m]$ and $X = [x_1, x_2, \dots, x_m]$ and b is a real constant.

Here, W and b are parameters of the classifier model to be evaluated given a training set D .

Finding a hyperplane

Let us consider a two-dimensional training set consisting two classes + and - as shown in Fig. 4.

Suppose, b_1 and b_2 are two decision boundaries above and below a hyperplane, respectively.

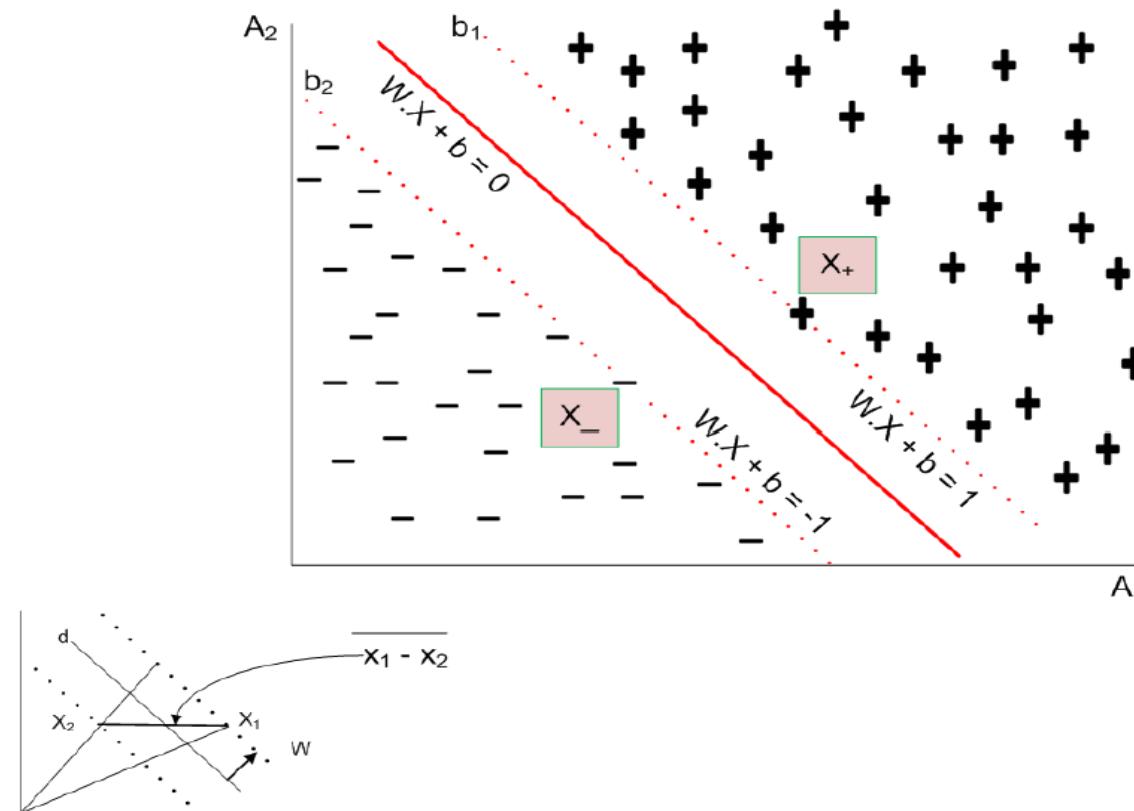
Consider any two points X_+ and X_- as shown in Fig. 4.

For X_+ located above the decision boundary, the equation can be written as

$$W.X_+ + b = K \quad \text{where } K > 0 \quad (5)$$

Finding a hyperplane

Figure 4: Computation of the MMH



Finding a hyperplane

Similarly, for any point X_- located below the decision boundary, the equation is

$$W \cdot X_- + b = K' \quad \text{where } K' < 0 \quad (6)$$

Thus, if we label all +'s as class label + and all -'s are class label -, then we can predict the class label Y for any test data X as

$$(Y) = \begin{cases} + & \text{if } W \cdot X + b > 0 \\ - & \text{if } W \cdot X + b < 0 \end{cases}$$

Hyperplane and Classification

Note that $W \cdot X + b = 0$, the equation representing hyperplane can be interpreted as follows.

- Here, W represents the orientation and b is the intercept of the hyperplane from the origin.
- If both W and b are scaled (up or down) by dividing a non zero constant, we get the same hyperplane.
- This means there can be infinite number of solutions using various scaling factors, all of them geometrical representing the same hyperplane.

Calculating Margin of a Hyperplane

To avoid such a confusion, we can make W and b unique by adding a constraint that $W'.X + b' = \pm 1$ for data points on boundary of each class.

It may be noted that $W'.X + b' = \pm 1$ represents two hyperplane parallel to each other.

For clarity in notation, we write this as $W.X + b = \pm 1$.

Having this understating, now we are in the position to calculate the margin of a hyperplane.

Calculating Margin of a Hyperplane

Suppose, x_1 and x_2 (refer Figure 3) are two points on the decision boundaries b_1 and b_2 , respectively. Thus,

$$W \cdot x_1 + b = 1 \quad (7)$$

$$W \cdot x_2 + b = -1 \quad (8)$$

or

$$W \cdot (x_1 - x_2) = 2 \quad (9)$$

This represents a dot (.) product of two vectors W and $x_1 - x_2$. Thus taking magnitude of these vectors, the equation obtained is

$$d = \frac{2}{\|W\|} \quad (10)$$

where $\|W\| = \sqrt{w_1^2 + w_2^2 + \dots + w_m^2}$ in an m -dimension space.

Calculating Margin of a Hyperplane

We calculate the margin more mathematically, as in the following.

- Consider two parallel hyperplanes H_1 and H_2 as shown in Fig. 5.
Let the equations of hyperplanes be

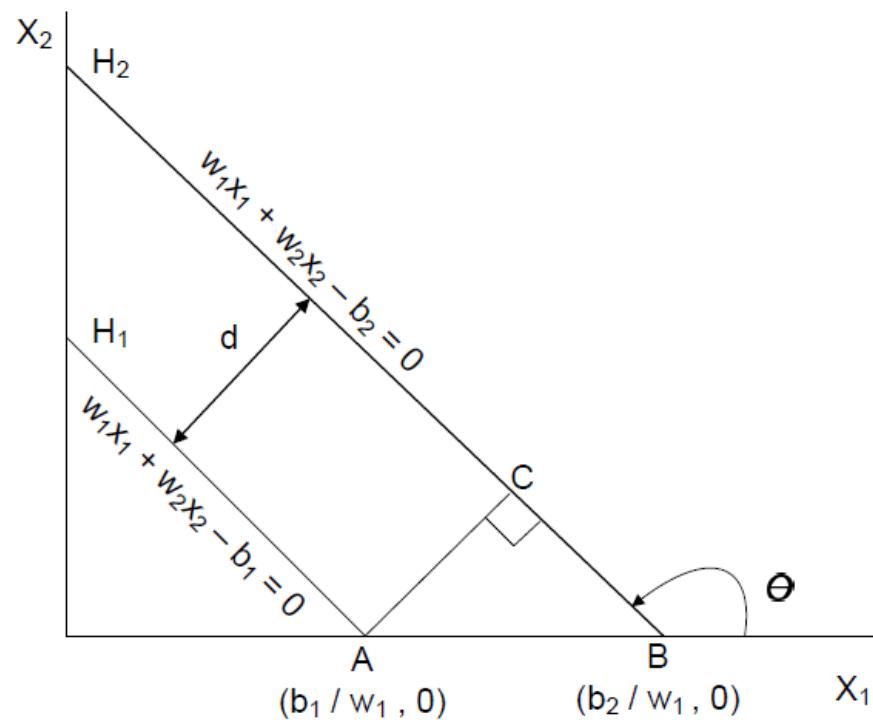
$$H_1 : w_1x_1 + w_2x_2 - b_1 = 0 \quad (11)$$

$$H_2 : w_1x_1 + w_2x_2 - b_2 = 0 \quad (12)$$

- To draw a perpendicular distance d between H_1 and H_2 , we draw a right-angled triangle ABC as shown in Fig.5.

Calculating Margin of a Hyperplane

Figure 5: Detail of margin calculation.



Calculating Margin of a Hyperplane

Being parallel, the slope of H_1 (and H_2) is $\tan\theta = -\frac{w_1}{w_2}$.

In triangle ABC, AB is the hypotenuse and AC is the perpendicular distance between H_1 and H_2 .

Thus, $\sin(180 - \theta) = \frac{AC}{AB}$ or $AC = AB \cdot \sin\theta$.

$$AB = \frac{b_2}{w_1} - \frac{b_1}{w_1} = \frac{|b_2 - b_1|}{w_1}, \sin\theta = \frac{w_1}{\sqrt{w_1^2 + w_2^2}}$$

(Since, $\tan\theta = -\frac{w_1}{w_2}$).

$$\text{Hence, } AC = \frac{|b_2 - b_1|}{\sqrt{w_1^2 + w_2^2}}.$$

!

Calculating Margin of a Hyperplane

This can be generalized to find the distance between two parallel margins of any hyperplane in n -dimensional space as

$$d = \frac{|b_2 - b_1|}{\sqrt{w_1^2 + w_2^2 + \dots + w_n^2}} \approx \frac{|b_2 - b_1|}{\| W \|}$$

where, $\| W \| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2}$.

In SVM literature, this margin is famously written as $\mu(W, b)$.

$mew(W, b)$

Calculating Margin of a Hyperplane

The training phase of SVM involves estimating the parameters W and b for a hyperplane from a given training data.

The parameters must be chosen in such a way that the following two inequalities are satisfied.

$$W \cdot x_i + b \geq 1 \quad \text{if } y_i = 1 \quad (13)$$

$$W \cdot x_i + b \leq -1 \quad \text{if } y_i = -1 \quad (14)$$

These conditions impose the requirements that all training tuples from class $Y = +$ must be located on or above the hyperplane $W \cdot x + b = 1$, while those instances from class $Y = -$ must be located on or below the hyperplane $W \cdot x + b = -1$ (also see Fig. 4).

Learning for a Linear SVM

Both the inequalities can be summarized as

$$y_i(W \cdot x_i + b) \geq 1 \quad \forall i = 1, 2, \dots, n \quad (15)$$

Note that any tuples that lie on the hyperplanes H_1 and H_2 are called support vectors.

Essentially, the support vectors are the most difficult tuples to classify and give the most information regarding classification.

In the following, we discuss the approach of finding MMH and the support vectors.

The above problem is turned out to be an optimization problem, that is, to maximize $\mu(W, b) = \frac{2}{\|w\|}$.

maximize

Searching for MMH

Maximizing the margin is, however, equivalent to minimizing the
following objective function

$$\mu(W, b) = \frac{\|W\|}{2} \quad (16)$$

In nutshell, the learning task in SVM, can be formulated as the
following constrained optimization problem.

$$\begin{aligned} & \text{minimize} && \mu(W, b) \\ & \text{subject to} && y_i(W \cdot x_i + b) \geq 1, \quad i = 1, 2, 3, \dots, n \end{aligned} \quad (17)$$

Searching for MMH

[convex optimization problem](#)

The above stated constrained optimization problem is popularly known as [convex optimization problem](#), where objective function is quadratic and constraints are linear in the parameters W and b .

The well known technique to solve a convex optimization problem is the standard [Lagrange Multiplier method](#).

[Lagrange Multiplier Method](#)

First, we shall learn the Lagrange Multiplier method, then come back to the solving of our own SVM problem.

Lagrange Multiplier Method

- The Lagrange multiplier method follows two different steps depending on type of constraints.

- 1 Equality constraint optimization problem:** In this case, the problem is of the form:

equality constraint optimization problem
inequality constraint optimization problem

$$\text{minimize } f(x_1, x_2, \dots, x_d)$$

$$\text{subject to } g_i(x) = 0, i = 1, 2, \dots, p$$

- 2 Inequality constraint optimization problem:** In this case, the problem is of the form:

$$\text{minimize } f(x_1, x_2, \dots, x_d)$$

$$\text{subject to } h_i(x) \leq 0, i = 1, 2, \dots, p$$

Lagrange Multiplier Method

Equality constraint optimization problem solving

- The following steps are involved in this case:

- Define the Lagrangian as follows:

$$(X, \lambda) = f(X) + \sum_{i=1}^p \lambda_i \cdot g_i(x) \quad (18)$$

where λ'_i 's are dummy variables called Lagrangian multipliers.

- Set the first order derivatives of the Lagrangian with respect to x and the Lagrangian multipliers λ'_i 's to zero's. That is

$$\begin{aligned}\frac{\delta L}{\delta x_i} &= 0, i = 1, 2, \dots, d \\ \frac{\delta L}{\delta \lambda_i} &= 0, i = 1, 2, \dots, p\end{aligned}$$

- Solve the $(d + p)$ equations to find the optimal value of $X = [x_1, x_2, \dots, x_d]$ and λ'_i 's.

Lagrange Multiplier Method

Example: Equality constraint optimization problem

Suppose, minimize $f(x, y) = x + 2y$

subject to $x^2 + y^2 - 4 = 0$

① Lagrangian $L(x, y, \lambda) = x + 2y + \lambda(x^2 + y^2 - 4)$

② $\frac{\partial L}{\partial x} = 1 + 2\lambda x = 0$

$$\frac{\partial L}{\partial y} = 1 + 2\lambda y = 0$$

$$\frac{\partial L}{\partial \lambda} = x^2 + y^2 - 4 = 0$$

③ Solving the above three equations for x , y and λ , we get $x = \mp \frac{2}{\sqrt{5}}$,
 $y = \mp \frac{4}{\sqrt{5}}$ and $\lambda = \pm \frac{\sqrt{5}}{4}$

Lagrange Multiplier Method

Example : Equality constraint optimization problem

- When $\lambda = \frac{\sqrt{5}}{4}$,

$$x = -\frac{2}{\sqrt{5}},$$

$$y = -\frac{4}{\sqrt{5}},$$

we get $f(x, y, \lambda) = -\frac{10}{\sqrt{5}}$

- Similarly, when $\lambda = -\frac{\sqrt{5}}{4}$,

$$x = \frac{2}{\sqrt{5}},$$

$$y = \frac{4}{\sqrt{5}},$$

we get $f(x, y, \lambda) = \frac{10}{\sqrt{5}}$

- Thus, the function $f(x, y)$ has its minimum value at

$$x = -\frac{2}{\sqrt{5}}, y = -\frac{4}{\sqrt{5}}$$

Lagrange Multiplier Method

Inequality constraint optimization problem solving

- The method for solving this problem is quite similar to the Lagrange multiplier method described above.
- It starts with the Lagrangian

$$L = f(x) + \sum_{i=1}^p \lambda_i \cdot h_i(x) \quad (19)$$

- In addition to this, it introduces additional constraints, called **Karush-Kuhn-Tucker (KKT) constraints**, which are stated in the next slide.

Karush kuhn tucker (KKT) constraints

Lagrange Multiplier Method

Inequality constraint optimization problem solving

$$\frac{\delta L}{\delta x_i} = 0, i = 1, 2, \dots, d$$

$$\lambda_i \geq 0, i = 1, 2, \dots, p$$

$$h_i(x) \leq 0, i = 1, 2, \dots, p$$

$$\lambda_i \cdot h_i(x) = 0, i = 1, 2, \dots, p$$

Solving the above equations, we can find the optimal value of $f(x)$.

Lagrange Multiplier Method

Example: Inequality constraint optimization problem

Consider the following problem.

Minimize $f(x, y) = (x - 1)^2 + (y - 3)^2$

subject to $x + y \leq 2$,

$y \geq x$

- The Lagrangian for this problem is

$$L = (x - 1)^2 + (y - 3)^2 + \lambda_1(x + y - 2) + \lambda_2(x - y).$$

subject to the KKT constraints, which are as follows:

Lagrange Multiplier Method

Example: Inequality constraint optimization problem

$$\frac{\delta L}{\delta x} = 2(x - 1) + \lambda_1 + \lambda_2 = 0$$

$$\frac{\delta L}{\delta y} = 2(y - 3) + \lambda_1 - \lambda_2 = 0$$

$$\lambda_1(x + y - 2) = 0$$

$$\lambda_2(x - y) = 0$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0$$

$$(x + y) \leq 2, y \geq x$$

Lagrange Multiplier Method

Example: Inequality constraint optimization problem

To solve KKT constraints, we have to check the following tests:

- Case 1: $\lambda_1 = 0, \lambda_2 = 0$

$$2(x - 1) = 0 \mid 2(y - 3) = 0 \Rightarrow x = 1, y = 3$$

since, $x + y = 4$, it violates $x + y \leq 2$; is not a feasible solution.

- Case 2: $\lambda_1 = 0, \lambda_2 \neq 0$ $2(x - y) = 0 \mid$

$$2(x - 1) + \lambda_2 = 0 \mid$$

$$2(y - 3) - \lambda_2 = 0$$

$$\Rightarrow x = 2, y = 2 \text{ and } \lambda_2 = -2$$

since, $x + y \leq 4$, it violates $\lambda_2 \geq 0$; is not a feasible solution.

Lagrange Multiplier Method

Example: Inequality constraint optimization problem

- Case 3: $\lambda_1 \neq 0, \lambda_2 = 0$ $2(x + y) = 2$

$$2(x - 1) + \lambda_1 = 0$$

$$2(y - 3) + \lambda_1 = 0$$

$\Rightarrow x = 0, y = 2$ and $\lambda_1 = 2$; this is a feasible solution.

- Case 4: $\lambda_1 \neq 0, \lambda_2 \neq 0$ $2(x + y) = 2$

$$2(x - y) = 0$$

$$2(x - 1) + \lambda_1 + \lambda_2 = 0$$

$$2(y - 3) + \lambda_1 - \lambda_2 = 0$$

$\Rightarrow x = 1, y = 1$ and $\lambda_1 = 2, \lambda_2 = -2$

This is not a feasible solution.

LMM to Solve Linear SVM

- The optimization problem for the linear SVM is inequality constraint optimization problem.
- The Lagrangian multiplier for this optimization problem can be written as

$$L = \frac{\|W\|^2}{2} - \sum_{i=1}^n \lambda_i(y_i(W \cdot x_i + b) - 1) \quad (20)$$

where the parameters λ_i 's are the Lagrangian multipliers, and $W = [w_1, w_2, \dots, w_m]$ and b are the model parameters.

LMM to Solve Linear SVM

- The KKT constraints are:

$$\frac{\delta L}{\delta w} = 0 \Rightarrow W = \sum_{i=1}^n \lambda_i \cdot y_i \cdot x_i$$

$$\frac{\delta L}{\delta b} = 0 \Rightarrow \sum_{i=1}^n \lambda_i \cdot y_i = 0$$

$$\lambda \geq 0, i = 1, 2, \dots, n$$

KKT

$$\lambda_i [y_i (W \cdot x_i + b) - 1] = 0, i = 1, 2, \dots, n$$

$$y_i (W \cdot x_i + b) \geq 1, i = 1, 2, \dots, n$$

- Solving KKT constraints are computationally expensive and can be solved using a typical linear/ quadratic programming technique (or any other numerical technique).

LMM to Solve Linear SVM

- We first solve the above set of equations to find all the feasible solutions.
- Then, we can determine optimum value of $\mu(W, b)$.

Note:

- ① Lagrangian multiplier λ_i must be zero unless the training instance x_i satisfies the equation $y_i(W \cdot x_i + b) = 1$. Thus, the training tuples with $\lambda_i > 0$ lie on the hyperplane margins and hence are support vectors.
- ② The training instances that do not lie on the hyperplane margin have $\lambda_i = 0$.

Classifying a test sample using linear SVM

- For a given training data, using SVM principle, we obtain MMH in the form of W , b and λ_i 's. This is the machine (i.e., the SVM).
- Now, let us see how this MMH can be used to classify a test tuple say X . This can be done as follows.

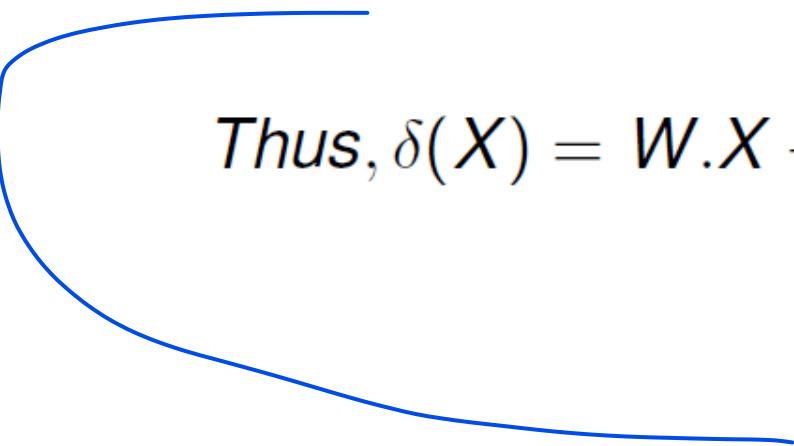
$$\delta(X) = W \cdot X + b = \sum_{i=1}^n \lambda_i \cdot y_i \cdot x_i \cdot X + b \quad (21)$$

Note that

$$W = \sum_{i=1}^n \lambda_i \cdot y_i \cdot x_i$$

Classifying a test sample using linear SVM

- This is famously called as “Representer Theorem” which states that the solution W always be represented as a linear combination of training data.


$$\text{Thus, } \delta(X) = W.X + b = \sum_{i=1}^n \lambda_i.y_i.x_i.X + b$$

Classifying a test sample using linear SVM

- The above involves a dot product of $x_i \cdot X$, where x_i is a support vector (this is so because $(\lambda_i) = 0$ for all training tuples except the support vectors), we can check the sign of $\delta(X)$.
- If it is positive, then X falls on or above the MMH and so the SVM predicts that X belongs to class label +. On the other hand, if the sign is negative, then X falls on or below MMH and the class prediction is -.

Note:

- ① Once the SVM is trained with training data, the complexity of the classifier is characterized by the number of support vectors.
- ② Dimensionality of data is not an issue in SVM unlike in other classifier.

complexity of svm classifier

dimensionality not an issue

Illustration: Linear SVM

- Consider the case of a binary classification starting with a training data of 8 tuples as shown in Table 1.
- Using quadratic programming, we can solve the KKT constraints to obtain the Lagrange multipliers λ_i for each training tuple, which is shown in Table 1.
- Note that only the first two tuples are support vectors in this case.
- Let $W = (w_1, w_2)$ and b denote the parameter to be determined now. We can solve for w_1 and w_2 as follows:

$$y = Wx + b, W = \text{sigma}(\lambda_i \cdot y_i \cdot x_{ij})$$

$$w_1 = \sum_i \lambda_i \cdot y_i \cdot x_{i1} = 65.52 \times 1 \times 0.38 + 65.52 \times -1 \times 0.49 = -6.64 \quad \rightarrow \quad \text{!!! summation over multiple training datas}$$

(22)

$$w_2 = \sum_i \lambda_i \cdot y_i \cdot x_{i2} = 65.52 \times 1 \times 0.47 + 65.52 \times -1 \times 0.61 = -9.32$$

(23)

Illustration: Linear SVM

Table 1: Training Data

A_1	A_2	y	λ_i
0.38	0.47	+	65.52
0.49	0.61	-	65.52
0.92	0.41	-	0
0.74	0.89	-	0
0.18	0.58	+	0
0.41	0.35	+	0
0.93	0.81	-	0
0.21	0.10	+	0

Illustration: Linear SVM

Figure 6: Linear SVM example.

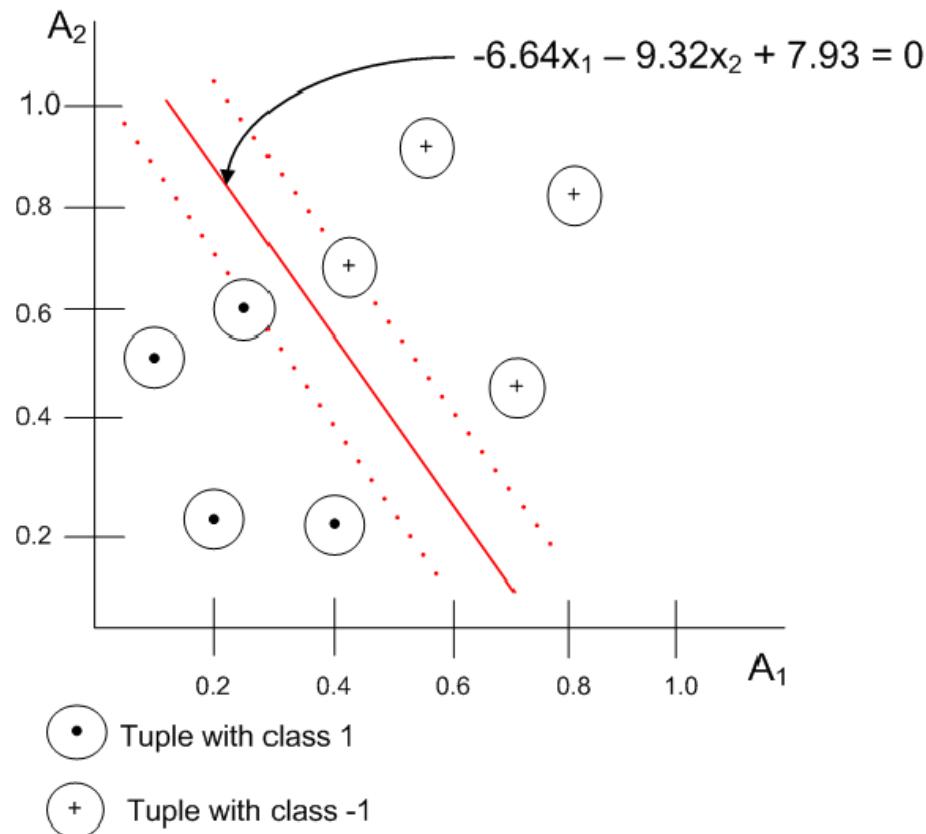


Illustration: Linear SVM

- The parameter b can be calculated for each support vector as follows

$$b_1 = 1 - W \cdot x_1 // \text{for support vector } x_1$$

$$= 1 - (-6.64) \times 0.38 - (-9.32) \times 0.47 // \text{using dot product}$$

$$= 7.93$$

$$b_2 = 1 - W \cdot x_2 // \text{for support vector } x_2$$

$$= 1 - (-6.64) \times 0.48 - (-9.32) \times 0.611 // \text{using dot product}$$

$$= 7.93$$

- Averaging these values of b_1 and b_2 , we get $b = 7.93$.

Illustration: Linear SVM

- Thus, the MMH is $-6.64x_1 - 9.32x_2 + 7.93 = 0$ (also see Fig. 6).

- Suppose, test data is $X = (0.5, 0.5)$. Therefore,

$$\delta(X) = W \cdot X + b$$

$$= -6.64 \times 0.5 - 9.32 \times 0.5 + 7.93$$

$$= -0.05$$

$$= -ve$$

- This implies that the test data falls on or below the MMH and SVM classifies that X belongs to class label -.

Classification of Multiple-class Data

- In the discussion of linear SVM, we have limited to binary classification (i.e., classification with two classes only).
- Note that the discussed linear SVM can handle any n -dimension, $n \geq 2$. Now, we are to discuss a more generalized linear SVM to classify n -dimensional data belong to two or more classes.
- There are two possibilities: all classes are pairwise linearly separable, or classes are overlapping, that is, not linearly separable.
- If the classes are pair wise linearly separable, then we can extend the principle of linear SVM to each pair. There are two strategies:
 - ① One versus one (OVO) strategy
 - ② One versus all (OVA) strategy

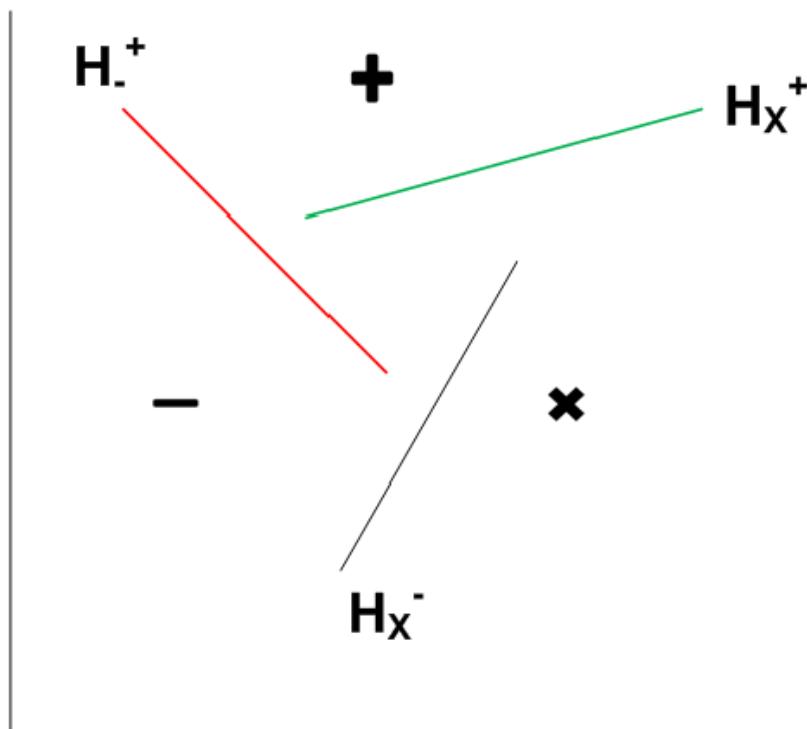
one vvs one (ovo)
one vs all (ova)

Classification of Multiple-class Data OVO Strategy

- In OVO strategy, we are to find MMHs for each pair of classes.
- Thus, if there are n classes, then ${}^n C_2$ pairs and hence so many classifiers possible (of course, some of which may be redundant).
- Also, see Fig. 7 for 3 classes (namely +, - and \times). Here, H_y^x denotes MMH between class labels x and y .
- Similarly, you can think for 4 classes (namely +, -, \times , \div) and more.

Classification of Multiple-class Data OVO Strategy

Figure 7: 3-pairwise linearly separable classes.



Classification of Multiple-class Data OVO Strategy

- With OVO strategy, we test each of the classifier in turn and obtain $\delta_i^j(X)$, that is, the count for MMH between i^{th} and j^{th} classes for test data X .
- If there is a class i , for which $\delta_i^j(X)$ for all j (and $j \neq i$), gives same sign, then unambiguously, we can say that X is in class i .

Classification of Multiple-class Data OVA Strategy

- OVO strategy is not useful for data with a large number of classes, as the computational complexity increases exponentially with the number of classes.
- As an alternative to OVO strategy, OVA(one versus all) strategy has been proposed.
- In this approach, we choose any class say C_i and consider that all tuples of other classes belong to a single class.

Classification of Multiple-class Data OVA Strategy

- This is, therefore, transformed into a binary classification problem and using the linear SVM discussed above, we can find the hyperplane. Let the hyperplane between C_i and remaining classes be MMH_i .
- The process is repeated for each $C_i \in [C_1, C_2, \dots, C_k]$ and getting MMH_i .
- Thus, with OVA strategies we get k classifiers.

Classification of Multiple-class Data OVA Strategy

- The unseen data X is then tested with each classifier so obtained.
- Let $\delta_j(X)$ be the test result with MMH_j , which has the maximum magnitude of test values.
- That is

$$\delta_j(X) = \max_{\forall i} \{\delta_i(X)\} \quad (24)$$

- Thus, X is classified into class C_j .

Classification of Multiple-class Data OVA Strategy

Note:

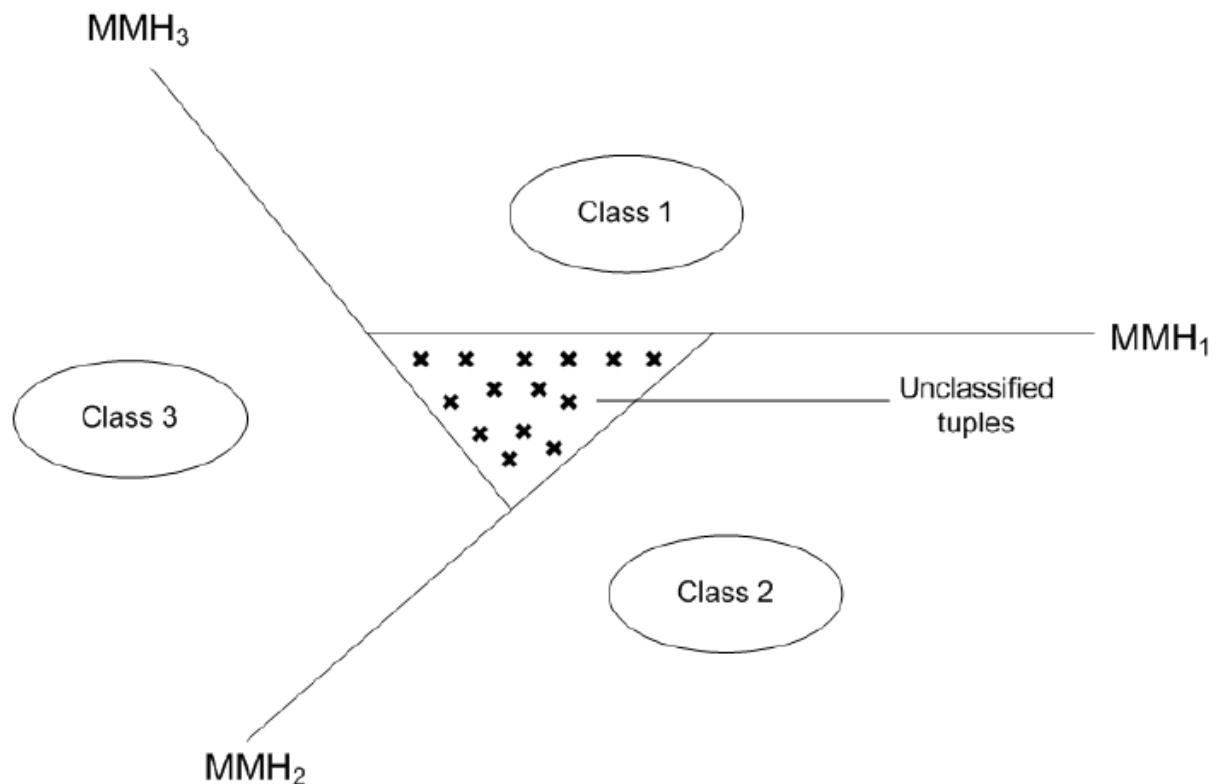
- The linear SVM that is used to classify multi-class data fails, if all classes are not linearly separable.
- If one class is linearly separable to remaining other classes and test data belongs to that particular class, then only it classifies accurately.

Classification of Multiple-class Data OVA Strategy

- Further, it is possible to have some tuples which cannot be classified none of the linear SVMs (also see Fig.8).
- There are some tuples which cannot be classified unambiguously by neither of the hyperplanes.
- All these tuples may be due to noise, errors or data are not linearly separable.

Classification of Multiple-class Data OVA Strategy

Figure 8: Unclassifiable region in OVA strategy.



Non-Linear SVM

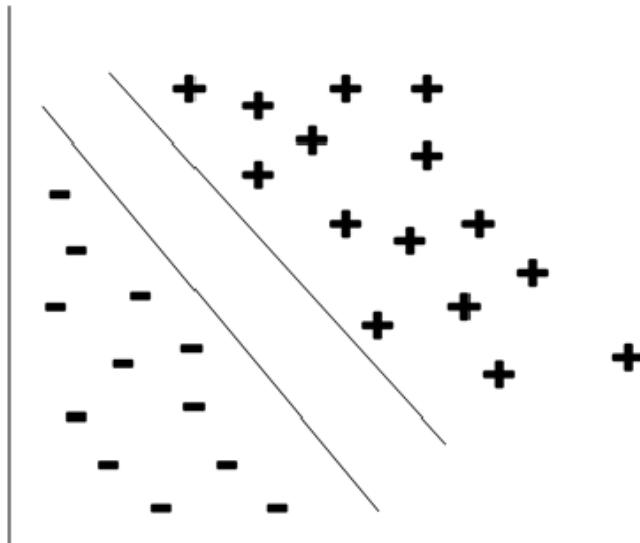
Non-Linear SVM

SVM classification for non separable data

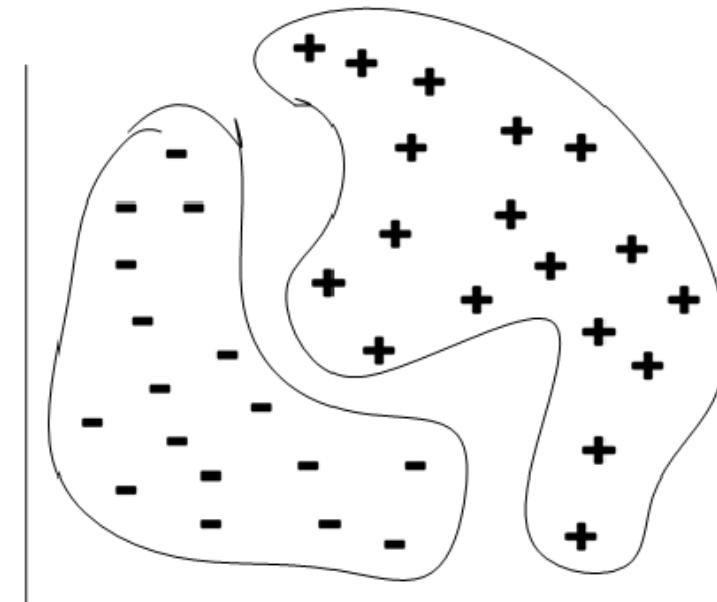
- Figure 9 shows a 2-D views of data when they are linearly separable and not separable.
- In general, if data are linearly separable, then there is a hyperplane otherwise no hyperplane.

Linear and Non-Linear Separable Data

Figure 9: Two types of training data.



(a) Linearly separable



(b) Linearly non-separable

SVM classification for non separable data

- Such a linearly not separable data can be classified using two approaches.
 - 1 Linear SVM with soft margin
 - 2 Non-linear SVM
- In the following, we discuss the extension of linear SVM to classify linearly not separable data.
- We discuss non-linear SVM in detail later.

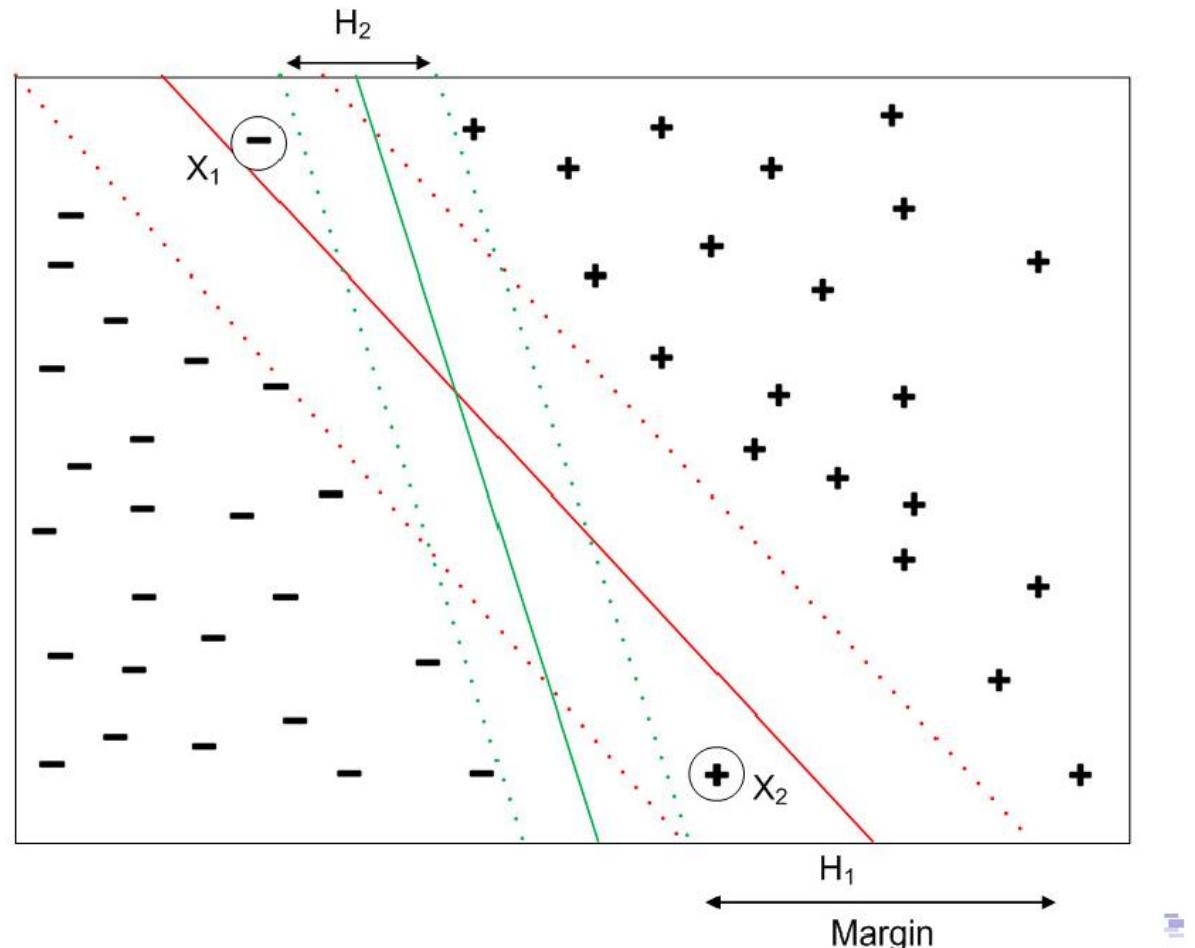
linear svm with soft margin
non linear svm

Linear SVM for Linearly Not Separable Data

- If the number of training data instances violating linear separability is less, then we can use linear SVM classifier to classify them.
- The rational behind this approach can be better understood from Fig. 10.

Linear SVM for Linearly Not Separable Data

Figure 10: Problem with linear SVM for linearly not separable data.



Linear SVM for Linearly Not Separable Data

- Suppose, X_1 and X_2 are two instances.
- We see that the hyperplane H_1 classifies wrongly both X_1 and X_2 .
- Also, we may note that with X_1 and X_2 , we could draw another hyperplane namely H_2 , which could classify all training data correctly.
- However, H_1 is more preferable than H_2 as H_1 has higher margin compared to H_2 and thus H_1 is less susceptible to over fitting.

Linear SVM for Linearly Not Separable Data

- In other words, a linear SVM can be refitted to learn a hyperplane that is tolerable to a small number of non-separable training data.
- The approach of refitting is called **soft margin approach** (hence, the SVM is called **Soft Margin SVM**), where it introduces slack variables to the inseparable cases.
- More specifically, the soft margin SVM considers a linear SVM hyperplane (i.e., linear decision boundaries) even in situations where the classes are not linearly separable.
- The concept of **Soft Margin SVM** is presented in the following slides.

Soft Margin SVM

- Recall that for linear SVM, we are to determine a maximum margin hyperplane $W \cdot X + b = 0$ with the following optimization:

$$\underset{W}{\text{minimize}} \frac{\|W\|^2}{2} \quad (25)$$

subject to $y_i \cdot (W \cdot x_i + b) \geq 1, i = 1, 2, \dots, n$

- In soft margin SVM, we consider the similar optimization technique except that a relaxation of inequalities, so that it also satisfies the case of linearly not separable data.

Soft Margin SVM

- To do this, soft margin SVM introduces slack variable (ξ), a positive-value into the constraint of optimization problem.
- Thus, for soft margin we rewrite the optimization problem as follows.

slack variable

$$\underset{\mathbf{W}}{\text{minimize}} \frac{\|\mathbf{W}\|^2}{2}$$

subject to $(\mathbf{W} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \text{ if } y_i = +1$

$(\mathbf{W} \cdot \mathbf{x}_i + b) \leq -1 + \xi_i, \text{ if } y_i = -1$

where $\forall i, \xi_i \geq 0$.

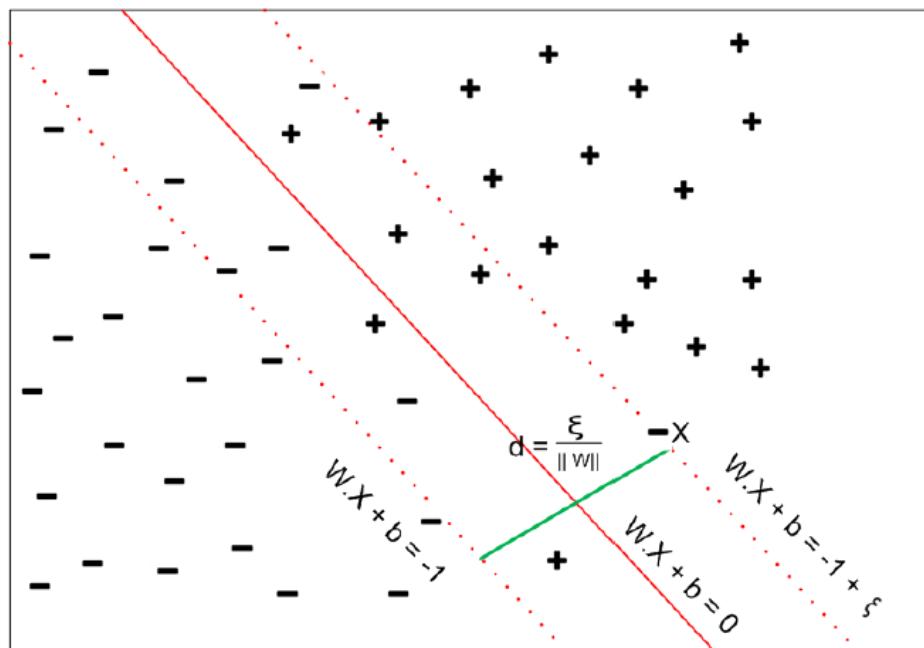
(26)

- Thus, in soft margin SVM, we are to calculate \mathbf{W} , b and ξ_i s as a solution to learn SVM.

Soft Margin SVM: Interpretation of ξ

- Let us find an interpretation of ξ , the slack variable in soft margin SVM. For this consider the data distribution shown in the Fig. 11.

Figure 11: Interpretation of slack variable ξ .



Soft Margin SVM: Interpretation of ξ

- The data X is one of the instances that violates the constraints to be satisfied for linear SVM.
- Thus, $W \cdot X + b = -1 + \xi$ represents a hyperplane that is parallel to the decision boundaries for class - and passes through X .
- It can be shown that the distance between the hyperplanes is $d = \frac{\xi}{\|W\|}$.
- In other words, ξ provides an estimate of the error of decision boundary on the training example X .

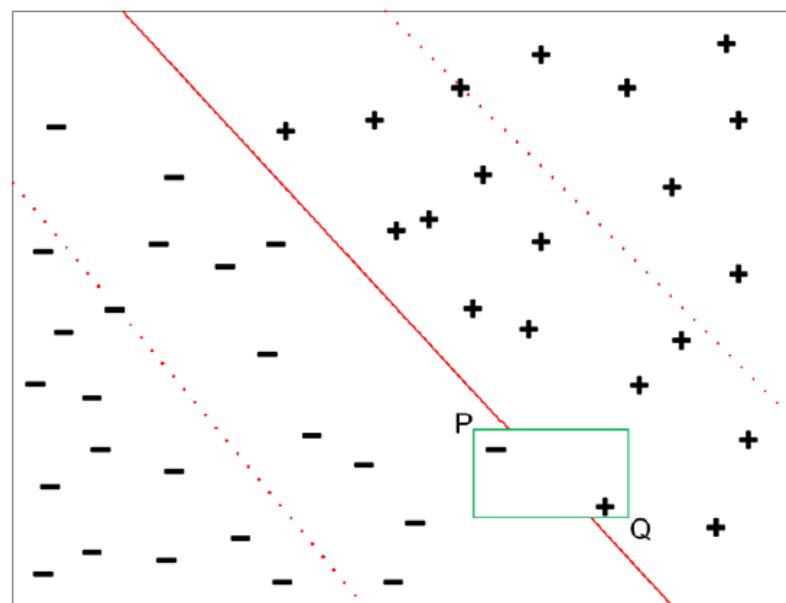
Soft Margin SVM: Interpretation of ξ

- In principle, Eqn. 25 and 26 can be chosen as the optimization problem to train a soft margin SVM.
- However, the soft margin SVM should impose a constraint on the number of such non linearly separable data it takes into account.
- This is so because a SVM may be trained with decision boundaries with very high margin thus, chances of misclassifying many of the training data.

Soft Margin SVM: Interpretation of ξ

- This is explained in Fig. 12. Here, if we increase margin further, then P and Q will be misclassified.
- Thus, there is a trade-off between the length of margin and training error.

Figure 12: MMH with wide margin and large training error.



Soft Margin SVM: Interpretation of ξ

penalty
training error
large margin

- To avoid this problem, it is therefore necessary to modify the objective function, so that penalizing for margins with a large gap, that is, large values of slack variables.
- The modified objective function can be written as

$$f(W) = \frac{\|W\|^2}{2} + c \cdot \sum_{i=1}^n (\xi_i)^\phi \quad (27)$$

where c and ϕ are user specified parameters representing the penalty of misclassifying the training data.

- Usually, $\phi = 1$. The larger value of c implies more penalty.

Solving for Soft Margin SVM

- We can follow the Lagrange multiplier method to solve the inequality constraint optimization problem, which can be reworked as follows:

$$L = \frac{\|W\|^2}{2} + c \cdot \sum_{i=1}^n \xi_i - \sum_{i=1}^n \lambda_i(y_i(W \cdot x_i + b) - 1 + \xi_i) - \sum_{i=1}^n \mu_i \cdot \xi_i \quad (28)$$

- Here, λ_i 's and μ_i 's are Lagrange multipliers.
- The inequality constraints are:

$$\xi_i \geq 0, \lambda_i \geq 0, \mu_i \geq 0.$$

$$\lambda_i \{y_i(W \cdot x_i + b) - 1 + \xi_i\} = 0$$

$$\mu_i \cdot \xi_i = 0$$

Solving for Soft Margin SVM

- The KKT constraints (in terms of first order derivative of L with respect to different parameters) are:

$$\frac{\delta L}{\delta w_j} = w_j - \sum_{i=1}^n \lambda_i \cdot y_i \cdot x_{ij} = 0 \quad \text{KKT}$$

$$w_j = \sum_{i=1}^n \lambda_i \cdot y_i \cdot x_{ij} = 0 \quad \forall i = 1, 2, \dots, n$$

$$\frac{\delta L}{\delta b} = - \sum_{i=1}^n \lambda_i \cdot y_i = 0$$

$$\sum_{i=1}^n \lambda_i \cdot y_i = 0$$

Solving for Soft Margin SVM

$$\frac{\delta L}{\delta \xi_i} = c - \lambda_i - \mu_i = 0$$

$$\lambda_i + \mu_i = c \text{ and } \mu_i \cdot \xi_i = 0 \quad \forall i = 1, 2, \dots, n. \quad (29)$$

- The above set of equations can be solved for values of $W = [w_1, w_2, \dots, w_m]$, b , $\lambda'_i s$, $\mu'_i s$ and $\xi'_i s$.
- **Note:**
 - ① $\lambda_i \neq 0$ for support vectors or has $\xi_i > 0$ and
 - ② $\mu_i = 0$ for those training data which are misclassified, that is, $\xi_i > 0$.

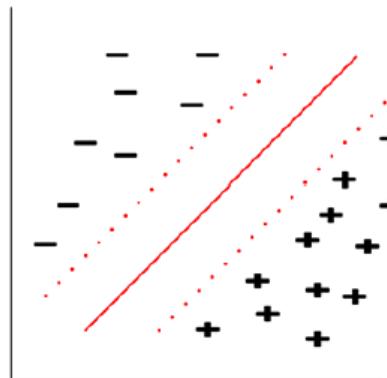
Non-Linear SVM

- Linear SVM undoubtedly better to classify data if it is trained by linearly separable data.
- Linear SVM also can be used for non-linearly separable data, provided that number of such instances is less.
- However, in real life applications, number of data overlapping is so high that soft margin SVM cannot cope to yield accurate classifier.
- As an alternative to this there is a need to compute a decision boundary, which is not linear (i.e., not a hyperplane rather hypersurface).

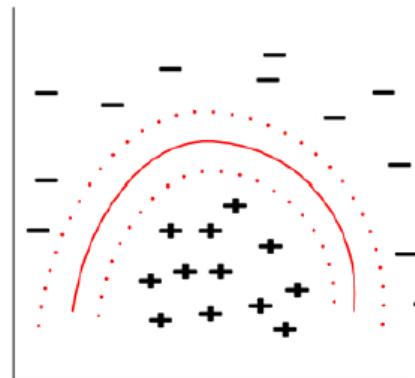
Non-Linear SVM

- For understanding this, see Figure 13.
- Note that a linear hyperplane is expressed as a linear equation in terms of n -dimensional component, whereas a non-linear hypersurface is a non-linear expression.

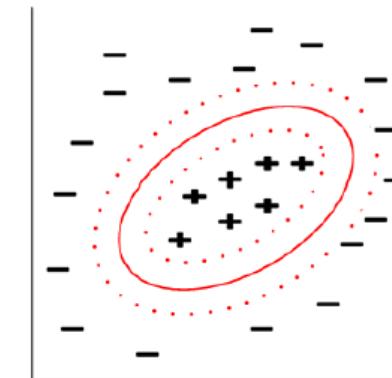
Figure 13: 2D view of few class separabilities.



(a) Linear hyperplane



(b) Nonlinear hyperplane



(c) Nonlinear hypersurface

Non-Linear SVM

- A hyperplane is expressed as

$$\text{linear : } w_1x_1 + w_2x_2 + w_3x_3 + c = 0 \quad (30)$$

- Whereas a non-linear hypersurface is expressed as.

$$\text{Nonlinear : } w_1x_1^2 + w_2x_2^2 + w_3x_1x_2 + w_4x_3^2 + w_5x_1x_3 + c = 0 \quad (31)$$

- The task therefore takes a turn to find a nonlinear decision boundaries, that is, nonlinear hypersurface in input space comprising with linearly not separable data.
- This task indeed neither hard nor so complex, and fortunately can be accomplished extending the formulation of linear SVM, we have already learned.

Non-Linear SVM

- This can be achieved in two major steps.
 - ① Transform the original (non-linear) input data into a higher dimensional space (as a linear representation of data).

Note that this is feasible because SVM's performance is decided by number of support vectors (i.e., \approx training data) not by the dimension of data.
 - ② Search for the linear decision boundaries to separate the transformed higher dimensional data.

The above can be done in the same line as we have done for linear SVM.

- In nutshell, to have a nonlinear SVM, the trick is to transform non-linear data into higher dimensional linear data.
- This transformation is popularly called **non linear mapping or attribute transformation**. The rest is same as the linear SVM.

non linear mapping
attribute transformation

Concept of Non-Linear Mapping

In order to understand the concept of non-linear transformation of original input data into a higher dimensional space, let us consider a non-linear second order polynomial in a 3-D input space.

$$X(x_1, x_2, x_3) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1^2 + w_5x_1x_2 + w_6x_1x_3 + c$$

The 3-D input vector $X(x_1, x_2, x_3)$ can be mapped into a 6-D space $Z(z_1, z_2, z_3, z_4, z_5, z_6)$ using the following mappings:

$$z_1 = \phi_1(x) = x_1$$

$$z_2 = \phi_2(x) = x_2$$

$$z_3 = \phi_3(x) = x_3$$

$$z_4 = \phi_4(x) = x_1^2$$

$$z_5 = \phi_5(x) = x_1 \cdot x_2$$

$$z_6 = \phi_6(x) = x_1 \cdot x_3$$

Concept of Non-Linear Mapping

- The transformed form of linear data in 6-D space will look like.

$$Z : w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6x_1z_6 + c$$

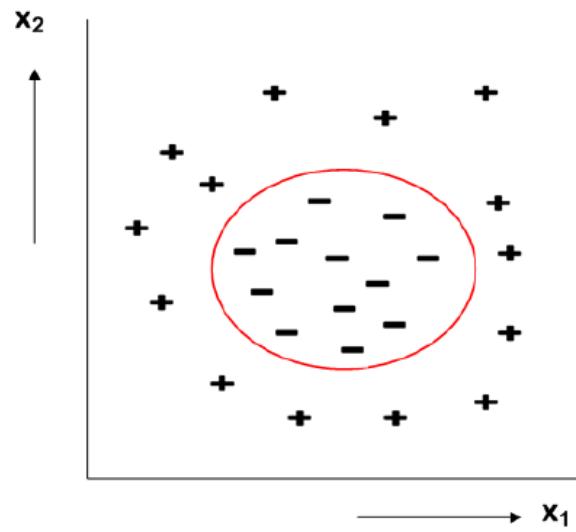
- Thus, if Z space has input data for its attributes x_1, x_2, x_3 (and hence Z 's values), then we can classify them using linear decision boundaries.

Concept of Non-Linear Mapping

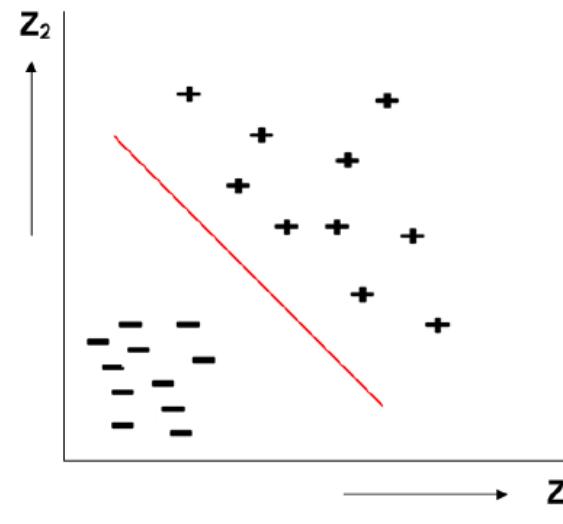
Example: Non-linear mapping to linear SVM

- The below figure shows an example of 2-D data set consisting of class label +1 (as +) and class label -1 (as -).

Figure 14: Non-linear mapping to Linear SVM.



X in 2D space



$X \rightarrow Z$ in 2D space

Concept of Non-Linear Mapping

Example: Non-linear mapping to linear SVM

- We see that all instances of class -1 can be separated from instances of class +1 by a circle, The following equation of the decision boundary can be thought of:

$$X(x_1, x_2) = +1 \text{ if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 2$$

$$X(x_1, x_2) = -1 \text{ otherwise} \quad (32)$$

Concept of Non-Linear Mapping

Example: Non-linear mapping to linear SVM

- The decision boundary can be written as:

$$X = \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} = 0.2$$

$$\text{or } x_1^2 - x_1 + x_2^2 - x_2 = 0.46$$

- A non-linear transformation in 2-D space is proposed as follows:

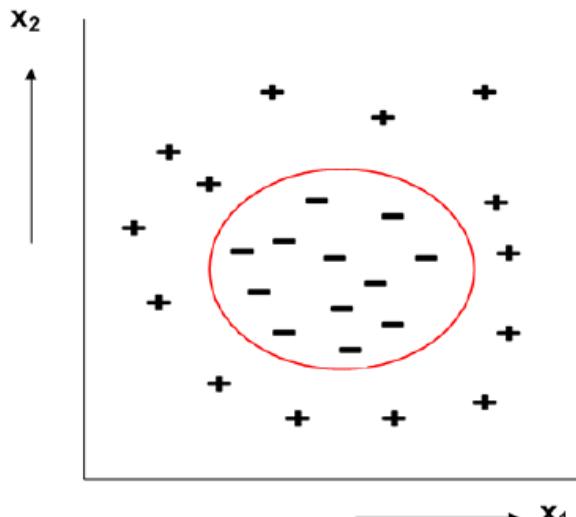
$$Z(z_1, z_2) : \phi_1(x) = x_1^2 - x_1, \phi_2(x) = x_2^2 - x_2 \quad (33)$$

Concept of Non-Linear Mapping

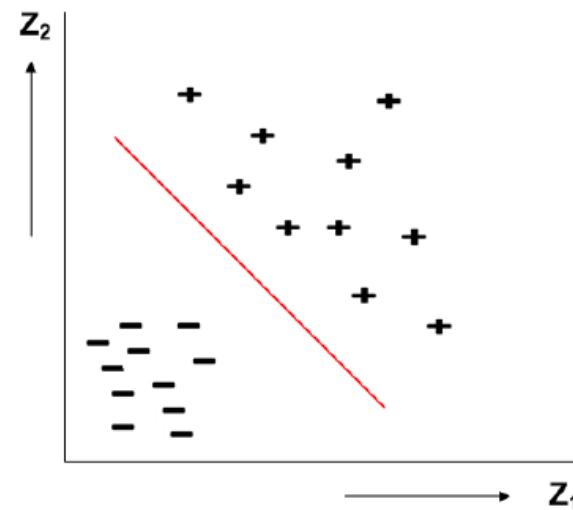
Example: Non-linear mapping to linear SVM

- The Z space when plotted will take view as shown in Fig. 15, where data are separable with linear boundary, namely
 $Z : z_1 + z_2 = -0.46$

Figure 15: Non-linear mapping to Linear SVM.



X in 2D space



X → Z in 2D space

Non-Linear to Linear Transformation: Issues

- The non linear mapping and hence a linear decision boundary concept looks pretty simple. But there are many potential problems to do so.

① **Mapping:** How to choose the non linear mapping to a higher dimensional space?

- In fact, the ϕ -transformation works fine for small example.
- But, it fails for realistically sized problems.

② **Cost of mapping:** For n -dimensional input instances there exist $N_H = \frac{(N+d-1)!}{d!(N-1)!}$ different monomials comprising a feature space of dimensionality N_H . Here, d is the maximum degree of monomial.

Non-Linear to Linear Transformation: Issues

• ...

- ③ Dimensionality problem: It may suffer from the curse of dimensionality problem often associated with a high dimensional data.
 - More specifically, in the calculation of $W.X$ or $X_i.X$ (in $\delta(X)$ see Eqn. 18), we need n multiplications and n additions (in their dot products) for each of the n -dimensional input instances and support vectors, respectively.
 - As the number of input instances as well as support vectors are enormously large, it is therefore, computationally expensive.
- ④ Computational cost: Solving the quadratic constrained optimization problem in the high dimensional feature space is too a computationally expensive task.

Non-Linear to Linear Transformation: Issues

- Fortunately, mathematicians have cleverly proposed an elegant solution to the above problems.
- Their solution consists of the following:
 - ① Dual formulation of optimization problem
 - ② Kernel trick
- In the next few slides, we shall learn about the above-mentioned two topics.

dual formulation of optimization problem
kernel trick

Dual Formulation of Optimization Problem

- We have already learned the Lagrangian formulation to find the maximum margin hyperplane as a linear SVM classifier.
- Such a formulation is called **primal form of the constraint optimization problem**.
primal form of the constraint optimization problem
- Primal form of Lagrangian optimization problem is reproduced further

$$\text{Minimize} \frac{\|W\|^2}{2}$$

Subject to $y_i(W.x_i + b) \geq 1, i = 1, 2, 3, \dots, n.$ (34)

Dual Formulation of Optimization Problem

- The primal form of the above mentioned inequality constraint optimization problem(according to Lagrange multiplier method) is given by

$$y_i(w \cdot x_i + b) \geq 1 \Rightarrow y_i(w \cdot x_i + b) - 1$$

$$L_p = \frac{\|W\|^2}{2} - \sum_{i=1}^n \lambda_i(y_i(W \cdot x_i + b) - 1) \quad (35)$$

where λ_i 's are called Lagrange multipliers.

- This L_p is called the **primal form of the Lagrangian optimization problem**.

Dual Formulation of Optimization Problem

- The dual form of the same problem can be derived as follows.
- To minimize the Lagrangian, we must take the derivative of L_p with respect to W , b and set them to zero.

$$\frac{\delta L_p}{\delta W} = 0 \Rightarrow W = \sum_{i=1}^n \lambda_i \cdot y_i \cdot x_i \quad (36)$$

$$\frac{\delta L_p}{\delta b} = 0 \Rightarrow \sum_{i=1}^n \lambda_i \cdot y_i = 0 \quad (37)$$

over ?

Dimensionality Reduction

Motivation

- Clustering

- One way to summarize a complex real-valued data point with a single categorical variable

- Dimensionality reduction

- Another way to simplify complex high-dimensional data
- Summarize data with a lower dimensional real valued vector

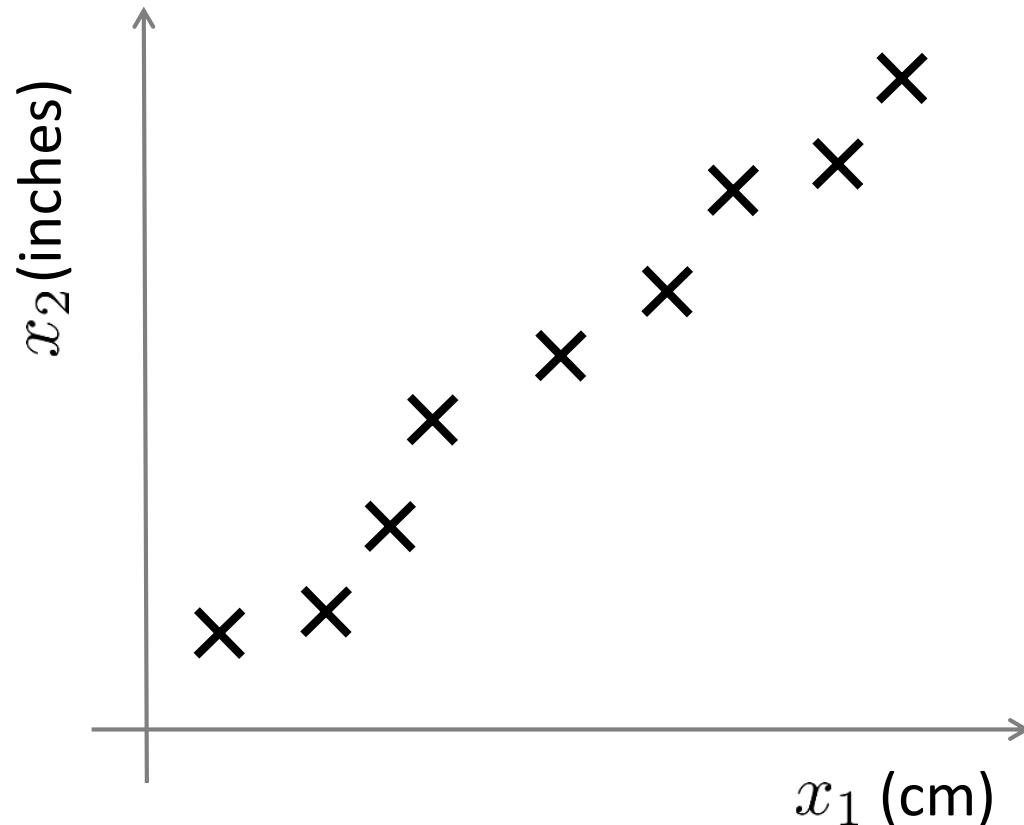
Motivation

- Clustering
 - One way to summarize a complex real-valued data point with a single categorical variable
- Dimensionality reduction
 - Another way to simplify complex high-dimensional data
 - Summarize data with a lower dimensional real valued vector



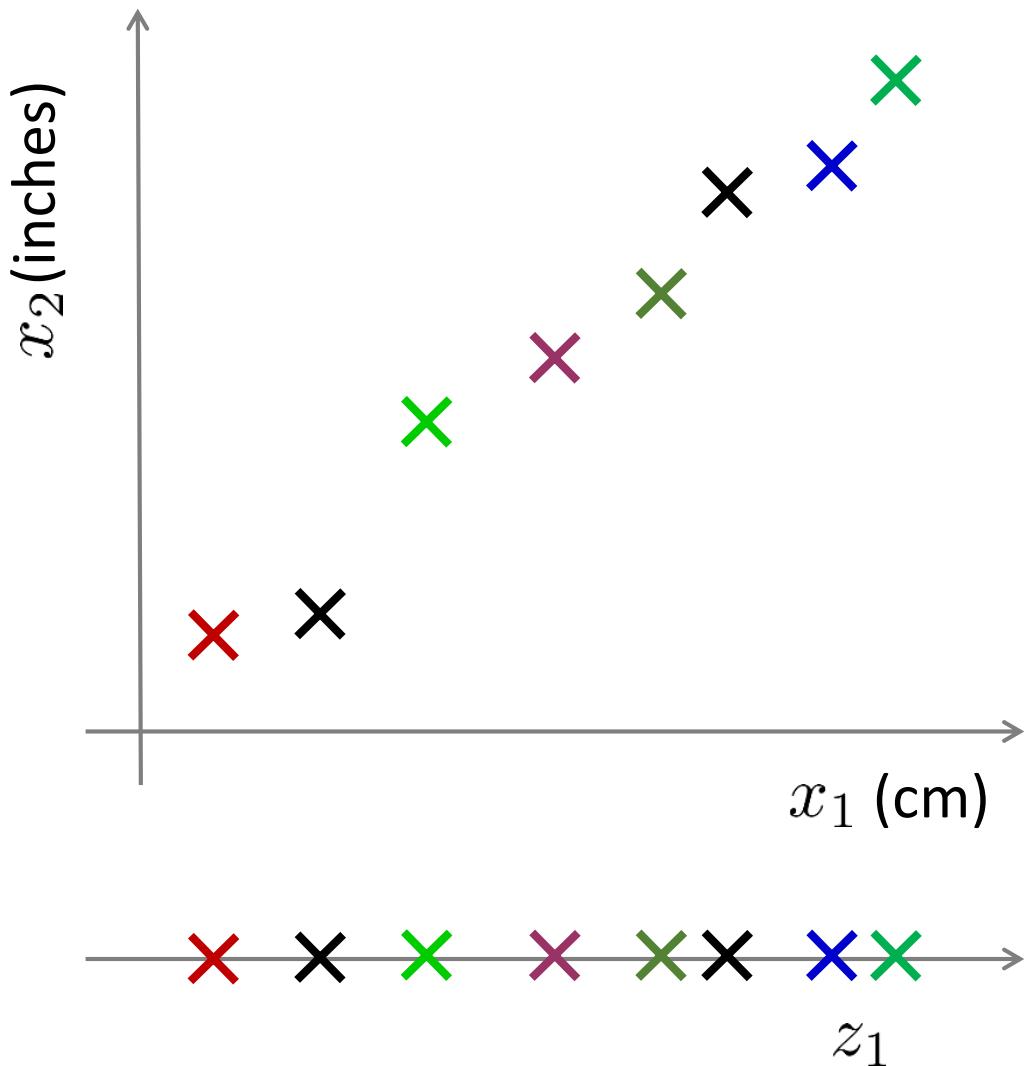
- Given data points in d dimensions
 - Convert them to data points in $r < d$ dimensions
 - With minimal loss of information

Data Compression



Reduce data from
2D to 1D

Data Compression



Reduce data from
2D to 1D

$$x^{(1)} \rightarrow z^{(1)}$$

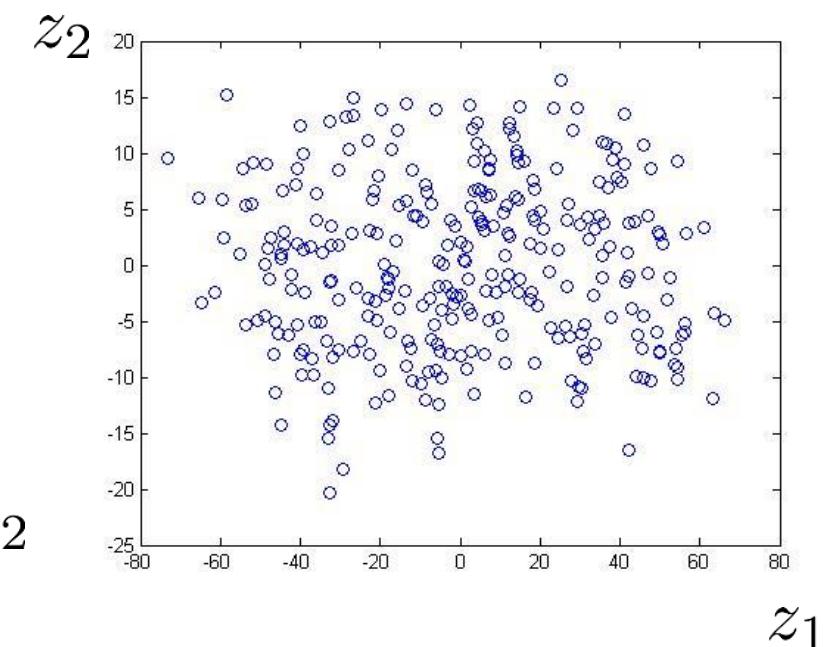
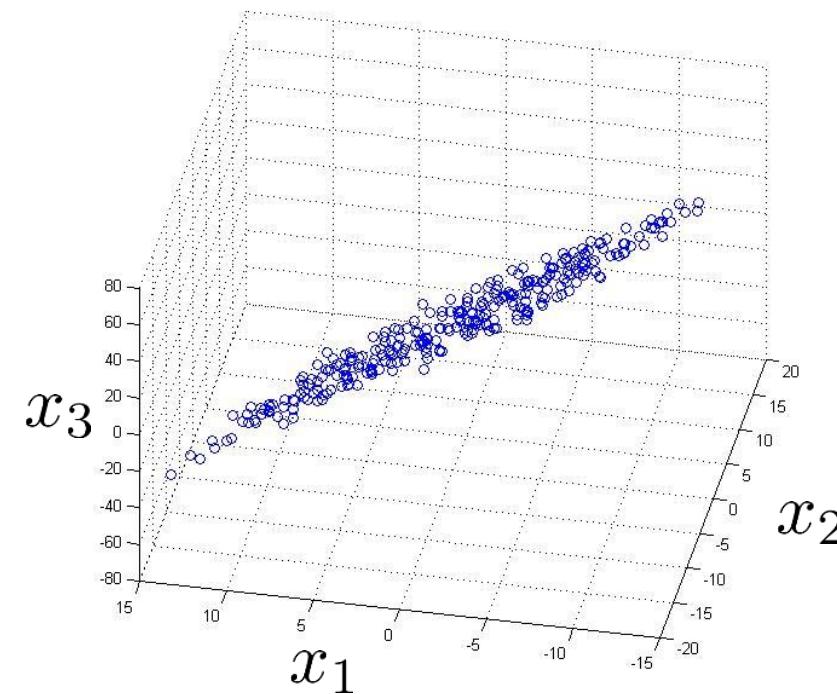
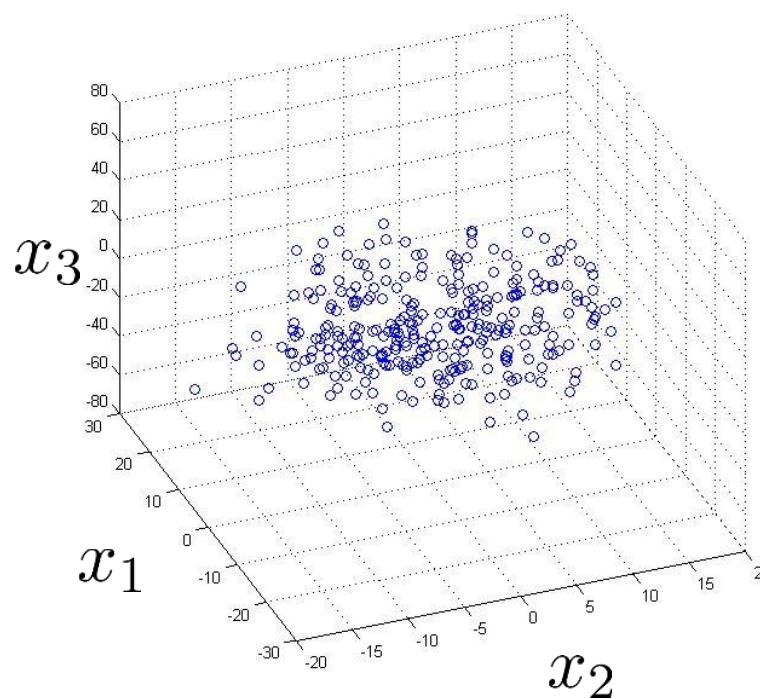
$$x^{(2)} \rightarrow z^{(2)}$$

⋮

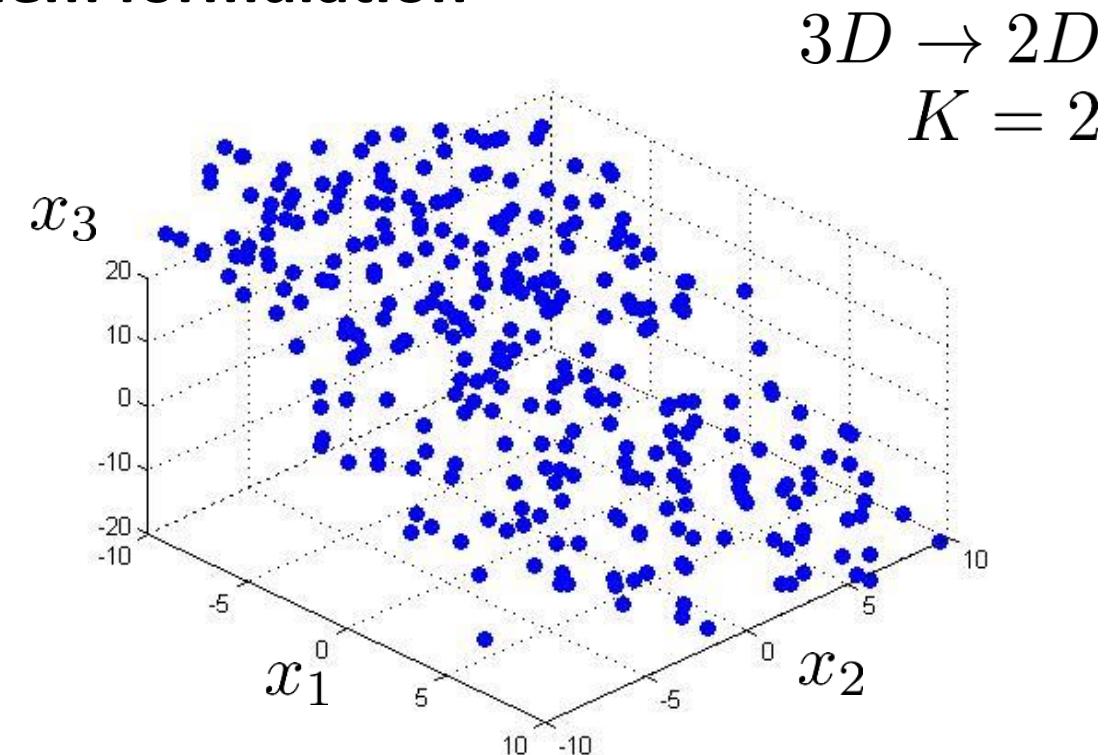
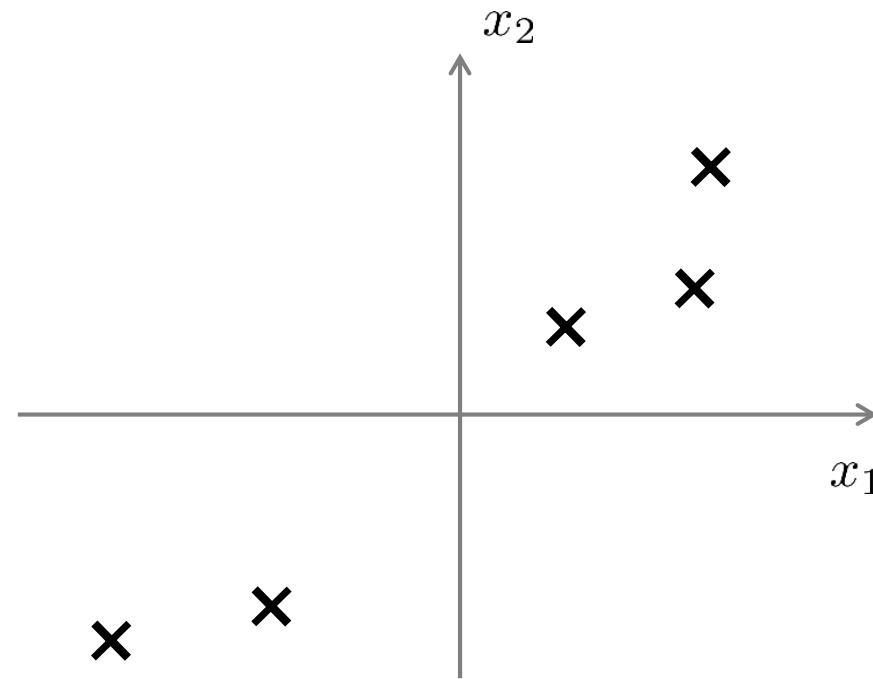
$$x^{(m)} \rightarrow z^{(m)}$$

Data Compression

Reduce data from 3D to 2D



Principal Component Analysis (PCA) problem formulation



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Principal Component Analysis

Goal: Find r -dim projection that best preserves variance

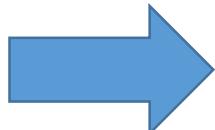
1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

where y is the new point, x is the old one,
and the rows of A are the eigenvectors

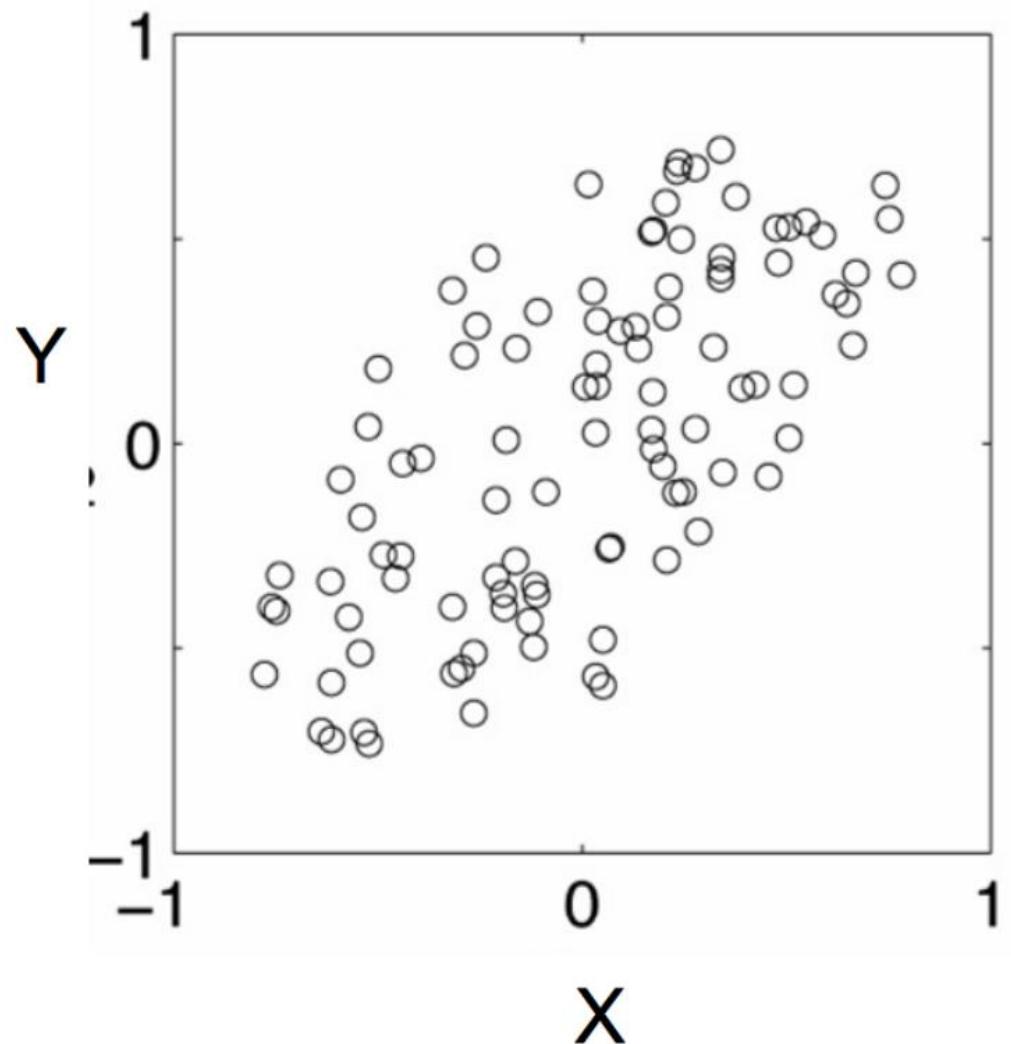
Covariance

- Variance and Covariance:
 - Measure of the “spread” of a set of points around their center of mass(mean)
- Variance:
 - Measure of the deviation from the mean for points in one dimension
- Covariance:
 - Measure of how much each of the dimensions vary from the mean with respect to each other



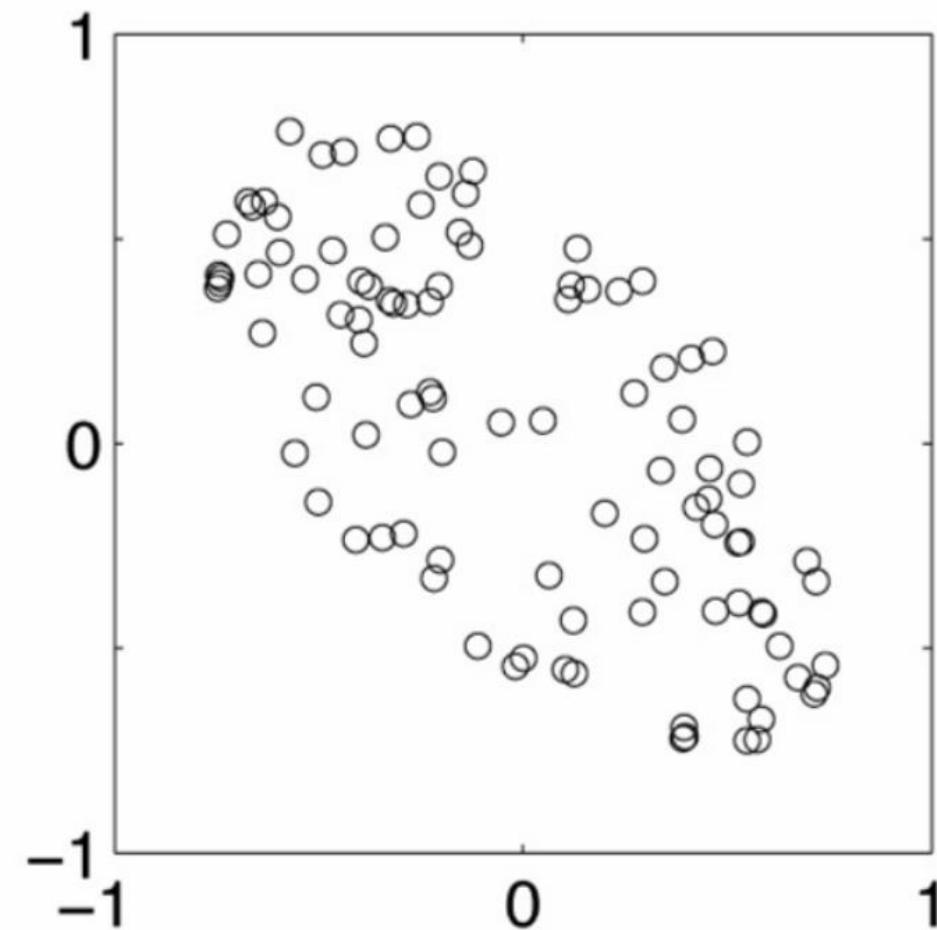
- Covariance is measured between two dimensions
- Covariance sees if there is a relation between two dimensions
- Covariance between one dimension is the variance

positive covariance



Positive: Both dimensions increase or decrease together

negative covariance



Negative: While one increase the other decrease

Covariance

- Used to find relationships between dimensions in high dimensional data sets

$$q_{jk} = \frac{1}{N} \sum_{i=1}^N (X_{ij} - E(X_j))(X_{ik} - E(X_k))$$

The Sample mean

maybe important?

Eigenvector and Eigenvalue

$$Ax = \lambda x$$

A: Square Matrix

λ : Eigenvector or characteristic vector

X: Eigenvalue or characteristic value



- *The zero vector can not be an eigenvector*
- *The value zero can be eigenvalue*

Eigenvector and Eigenvalue

$$Ax = \lambda x$$

A: Square Matrix

λ : Eigenvector or characteristic vector

X: Eigenvalue or characteristic value

Example

Show $x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is an eigenvector for $A = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix}$

$$\text{Solution: } Ax = \begin{bmatrix} 2 & -4 \\ 3 & -6 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\text{But for } \lambda = 0, \lambda x = 0 \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Thus, x is an eigenvector of A , and $\lambda = 0$ is an eigenvalue.

Eigenvector and Eigenvalue

$$Ax = \lambda x$$



$$Ax - \lambda x = 0$$

$$(A - \lambda I)x = 0$$

If we define a new matrix B:



$$B = A - \lambda I$$

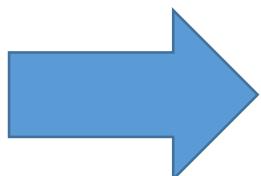
$$Bx = 0$$

If B has an inverse:



$$x = B^{-1}0 = 0$$

X BUT! an eigenvector cannot be zero!!



x will be an eigenvector of A if and only if B does not have an inverse, or equivalently $\det(B)=0$:

$$\det(A - \lambda I) = 0$$

Eigenvector and Eigenvalue

Example 1: Find the eigenvalues of

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$$

$$\begin{aligned} |\lambda I - A| &= \begin{vmatrix} \lambda - 2 & 12 \\ -1 & \lambda + 5 \end{vmatrix} = (\lambda - 2)(\lambda + 5) + 12 \\ &= \lambda^2 + 3\lambda + 2 = (\lambda + 1)(\lambda + 2) \end{aligned}$$

two eigenvalues: $-1, -2$

Note: The roots of the characteristic equation can be repeated. That is, $\lambda_1 = \lambda_2 = \dots = \lambda_k$.
If that happens, the eigenvalue is said to be of multiplicity k.

Example 2: Find the eigenvalues of

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$|\lambda I - A| = \begin{vmatrix} \lambda - 2 & -1 & 0 \\ 0 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 2 \end{vmatrix} = (\lambda - 2)^3 = 0$$

$\lambda = 2$ is an eigenvector of multiplicity 3.

Principal Component Analysis

Input:

$$\mathbf{x} \in \mathbb{R}^D: \mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

Set of basis vectors: $\mathbf{u}_1, \dots, \mathbf{u}_K$

Summarize a D dimensional vector X with K dimensional feature vector $h(\mathbf{x})$

$$h(\mathbf{x}) = \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{x} \\ \mathbf{u}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{u}_K \cdot \mathbf{x} \end{bmatrix}$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$$

Basis vectors are orthonormal

$$\mathbf{u}_i^T \mathbf{u}_j = 0$$

$$\|\mathbf{u}_j\| = 1$$

New data representation $h(\mathbf{x})$

$$z_j = \mathbf{u}_j \cdot \mathbf{x}$$

$$h(\mathbf{x}) = [z_1, \dots, z_K]^T$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$$

New data representation $h(\mathbf{x})$

$$h(\mathbf{x}) = \mathbf{U}^T \mathbf{x}$$

$$h(\mathbf{x}) = \mathbf{U}^T (\mathbf{x} - \mu_0)$$

Empirical mean of the data



$$\mu_0 = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$$

The space of all face images

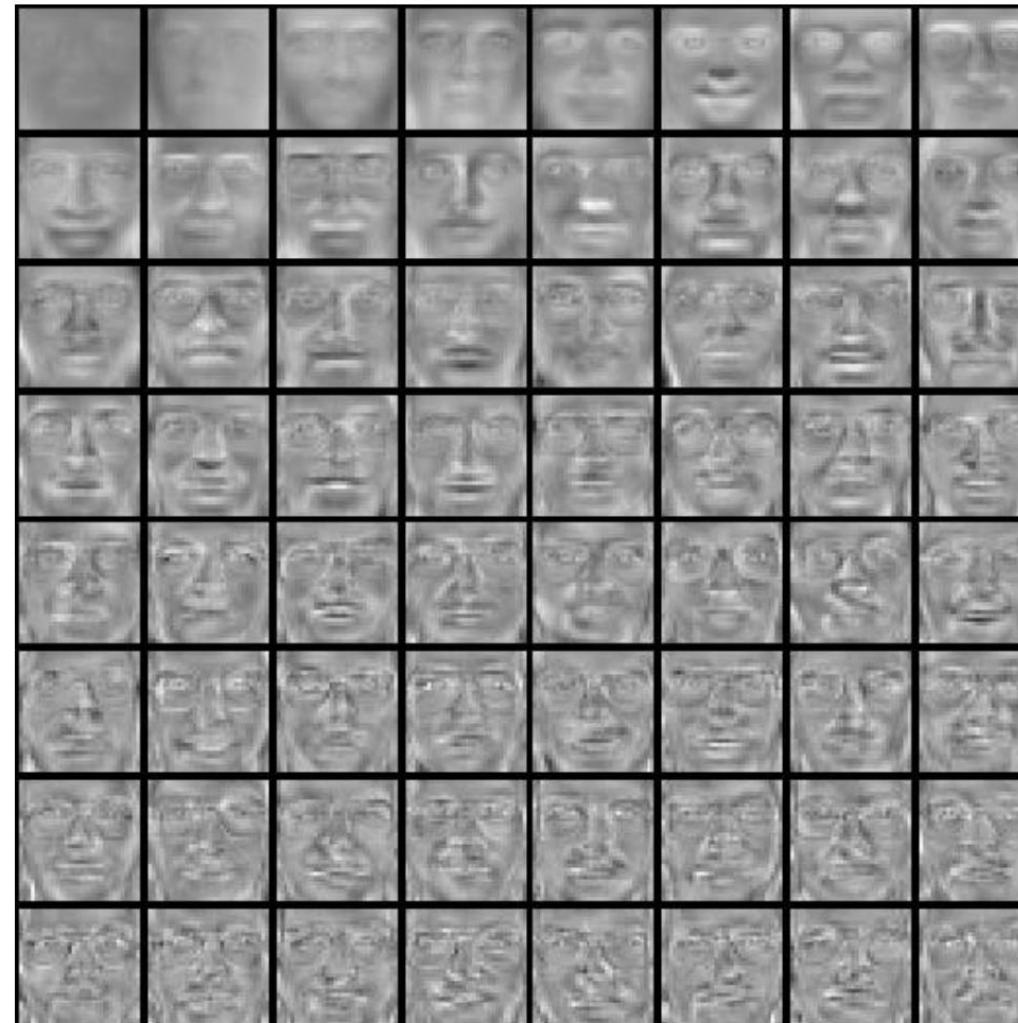
- When viewed as vectors of pixel values, face images are extremely high-dimensional
 - 100×100 image = 10,000 dimensions
 - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images



Eigenfaces example

Top eigenvectors: u_1, \dots, u_k

Mean: μ



Representation and reconstruction

- Face \mathbf{x} in “face space” coordinates:



$$\begin{aligned}\mathbf{x} &\rightarrow [\mathbf{u}_1^T(\mathbf{x} - \mu), \dots, \mathbf{u}_k^T(\mathbf{x} - \mu)] \\ &= w_1, \dots, w_k\end{aligned}$$

- Reconstruction:

$$\begin{aligned}\hat{\mathbf{x}} &= \mathbf{\mu} + w_1 \mathbf{u}_1 + w_2 \mathbf{u}_2 + w_3 \mathbf{u}_3 + w_4 \mathbf{u}_4 + \dots \\ &= \mathbf{\mu} + w_1 \mathbf{u}_1 + w_2 \mathbf{u}_2 + w_3 \mathbf{u}_3 + w_4 \mathbf{u}_4 + \dots\end{aligned}$$

The diagram shows the reconstruction of a face. On the left is the original face image. To its right is an equals sign followed by a mean face image. To the right of that is a plus sign. To the right of the plus sign is a vertical stack of seven smaller face images, each representing a basis function \mathbf{u}_i . Below the original face image is a superscripted \wedge over the letter x , indicating the reconstructed face.

Reconstruction

$P = 4$



$P = 200$

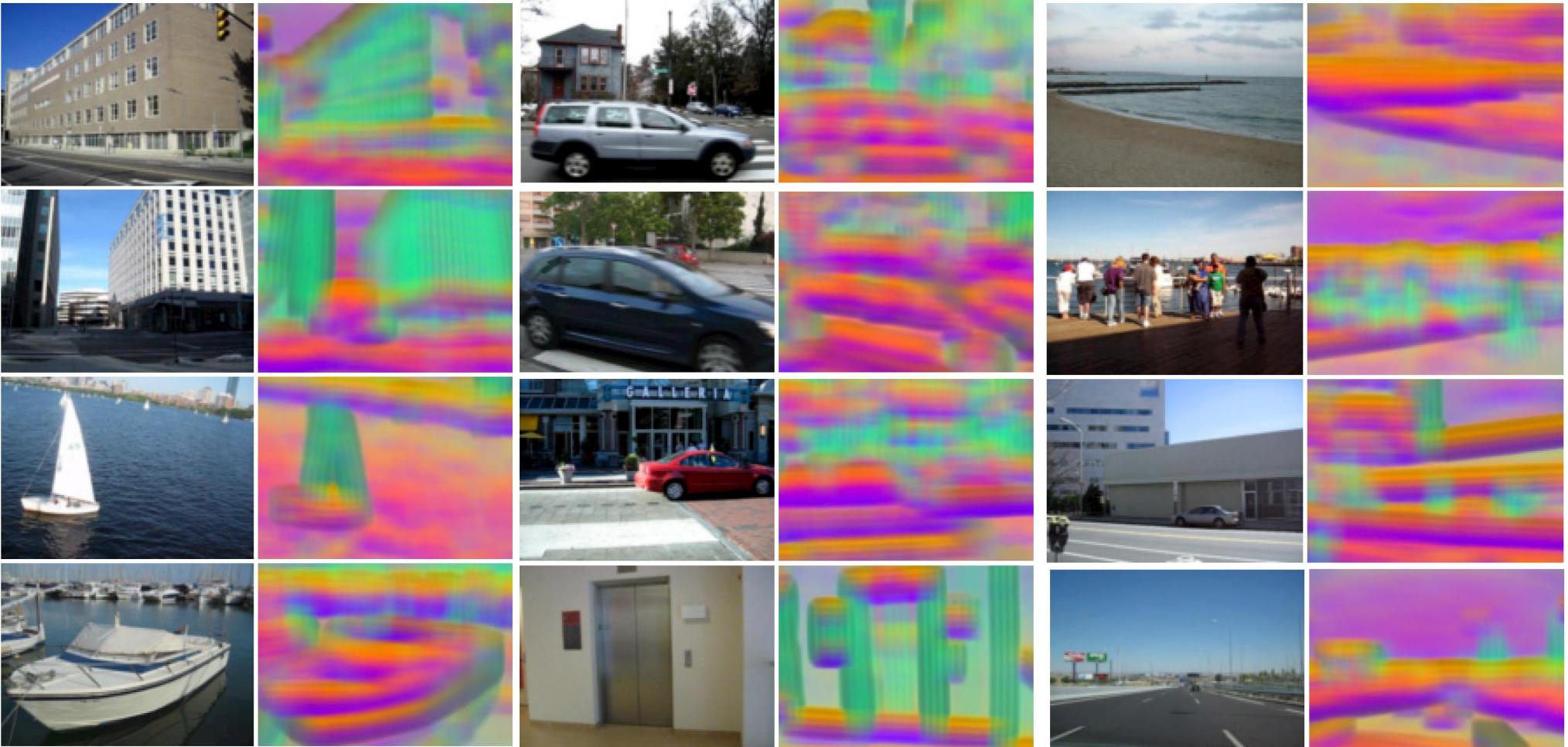


$P = 400$



After computing eigenfaces using 400 face images from ORL face database

SIFT feature visualization



- The top three principal components of SIFT descriptors from a set of images are computed
- Map these principal components to the principal components of the RGB space
- pixels with similar colors share similar structures

Application: Image compression



- Divide the original 372x492 image into patches:
 - Each patch is an instance that contains 12x12 pixels on a grid
 - View each as a 144-D vector

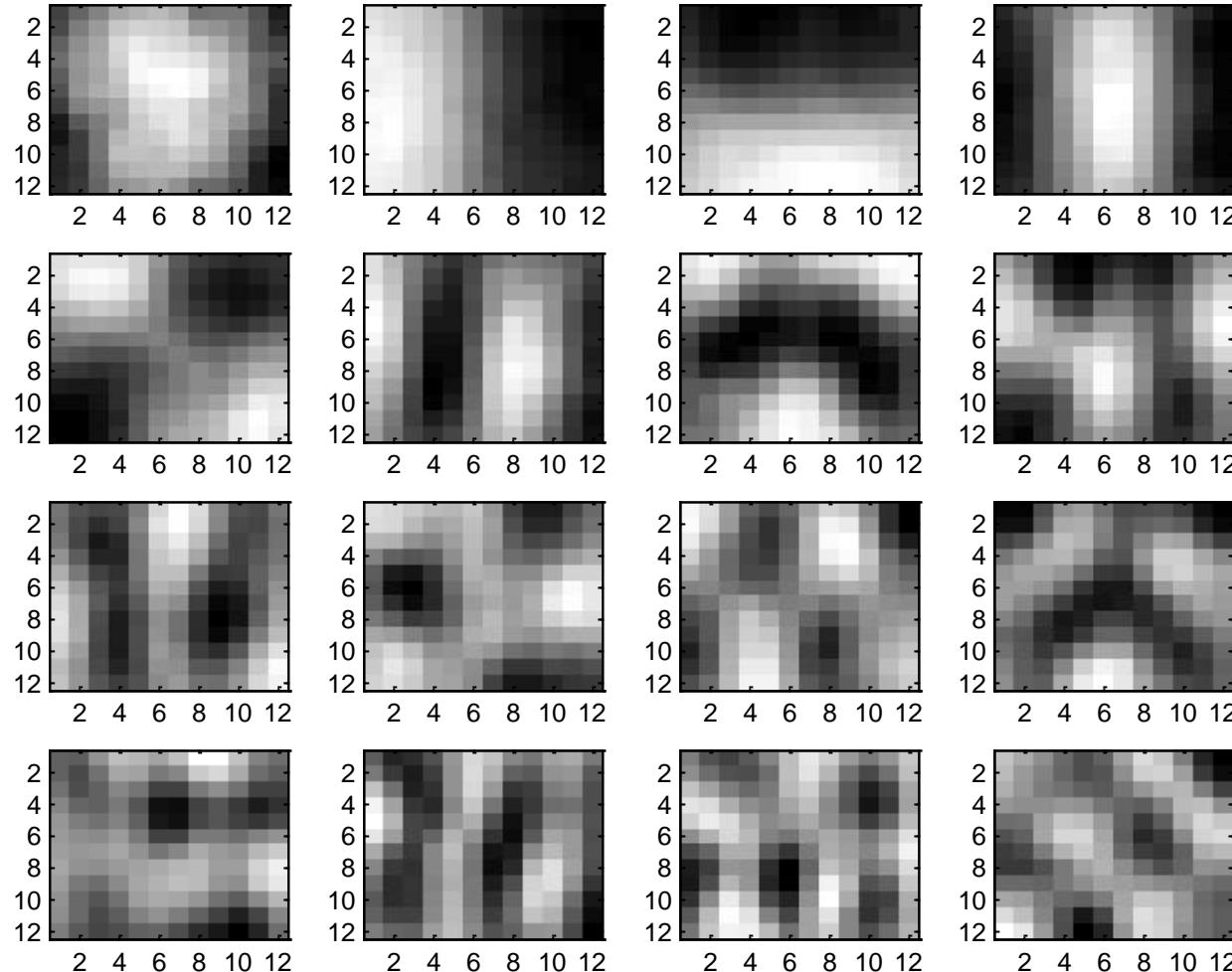
PCA compression: 144D → 60D



PCA compression: 144D → 16D



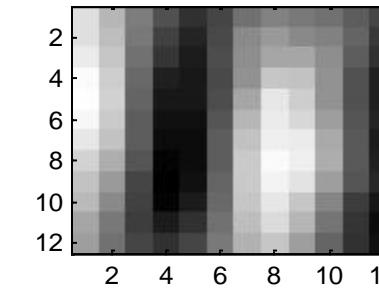
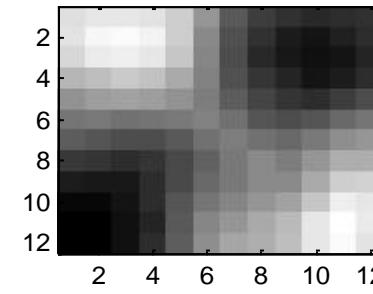
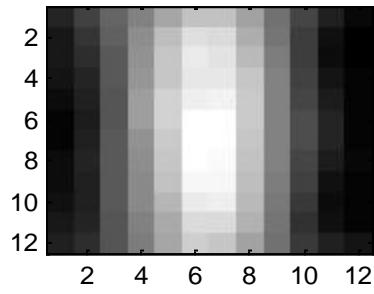
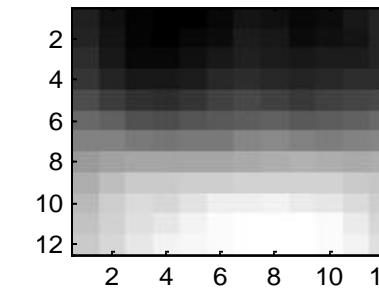
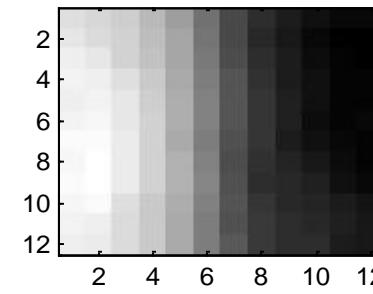
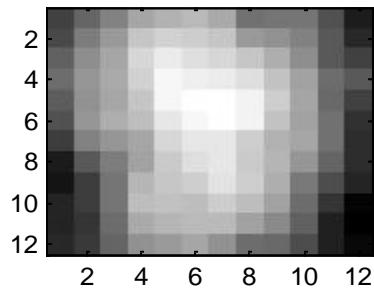
16 most important eigenvectors



PCA compression: 144D → 6D



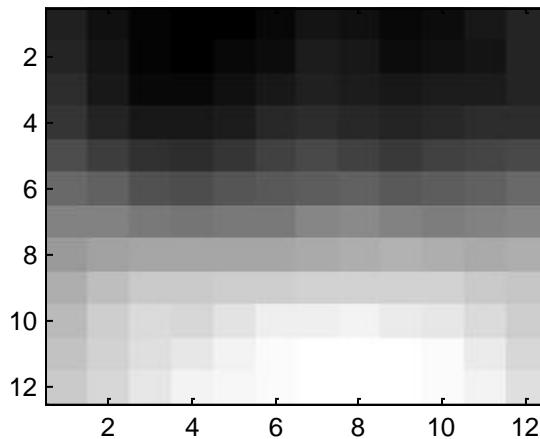
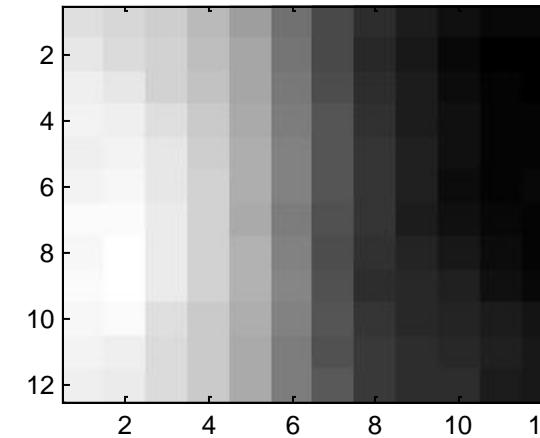
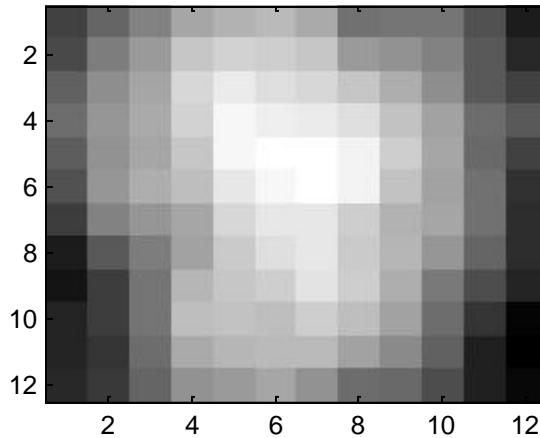
6 most important eigenvectors



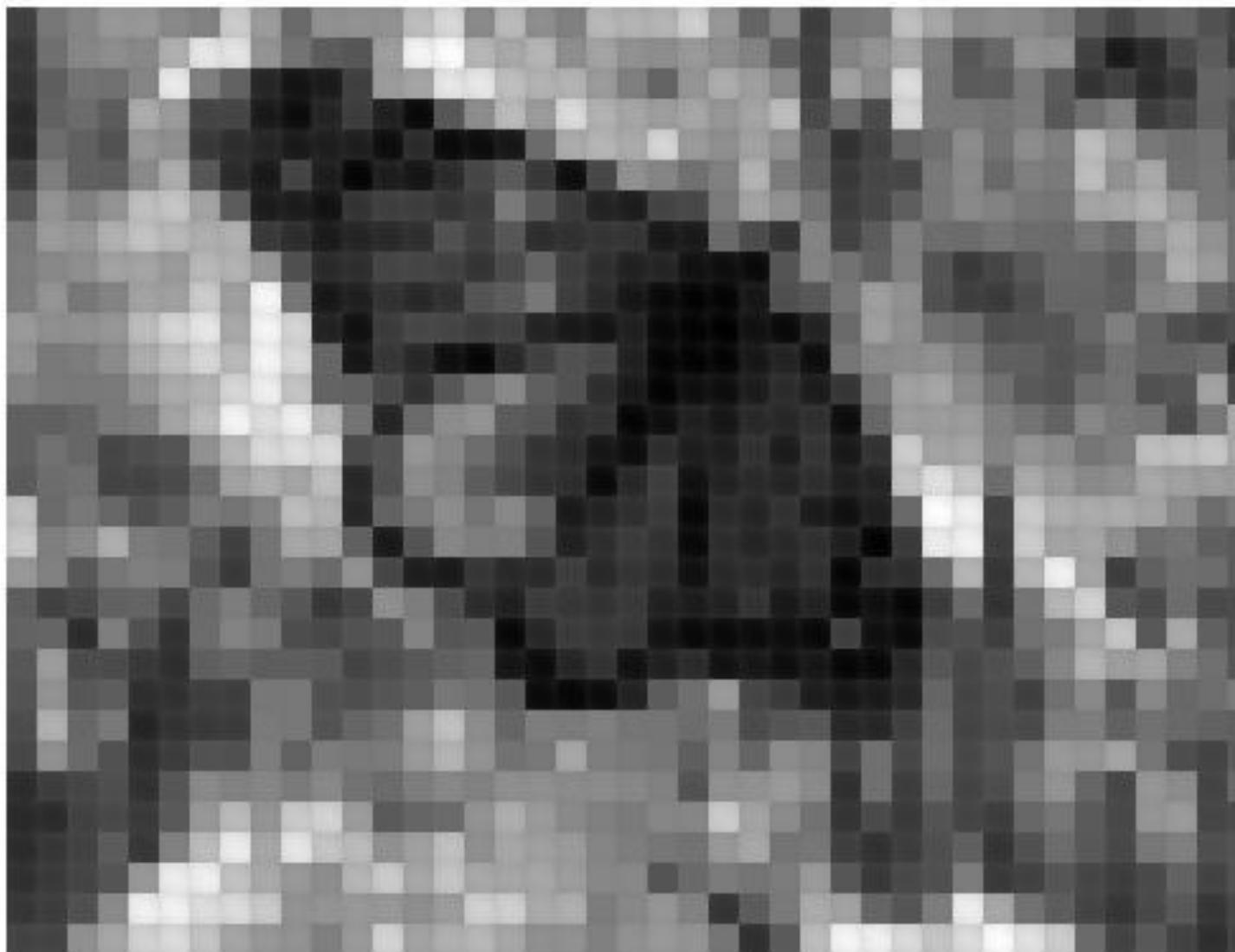
PCA compression: 144D \rightarrow 3D



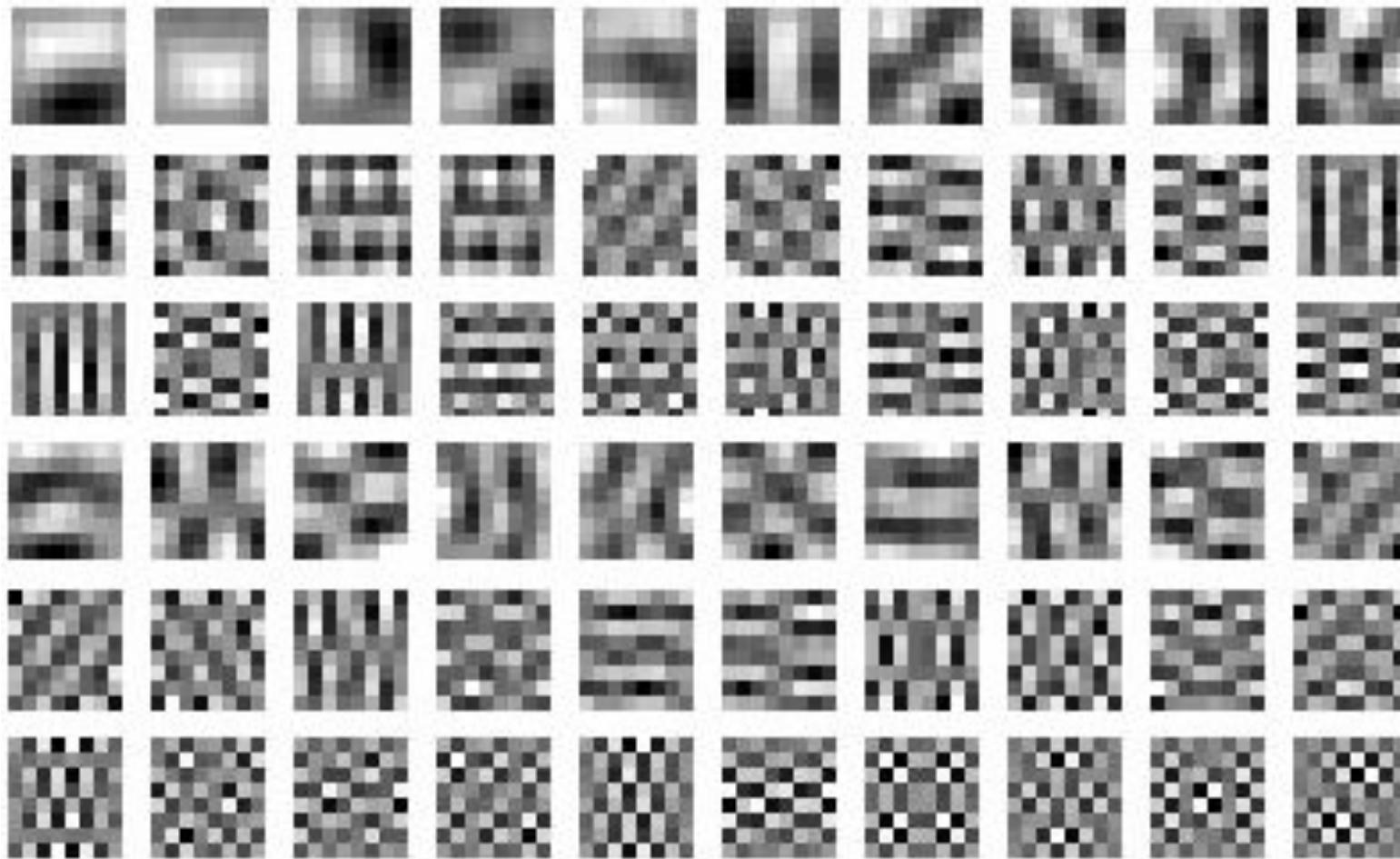
3 most important eigenvectors



PCA compression: 144D \rightarrow 1D

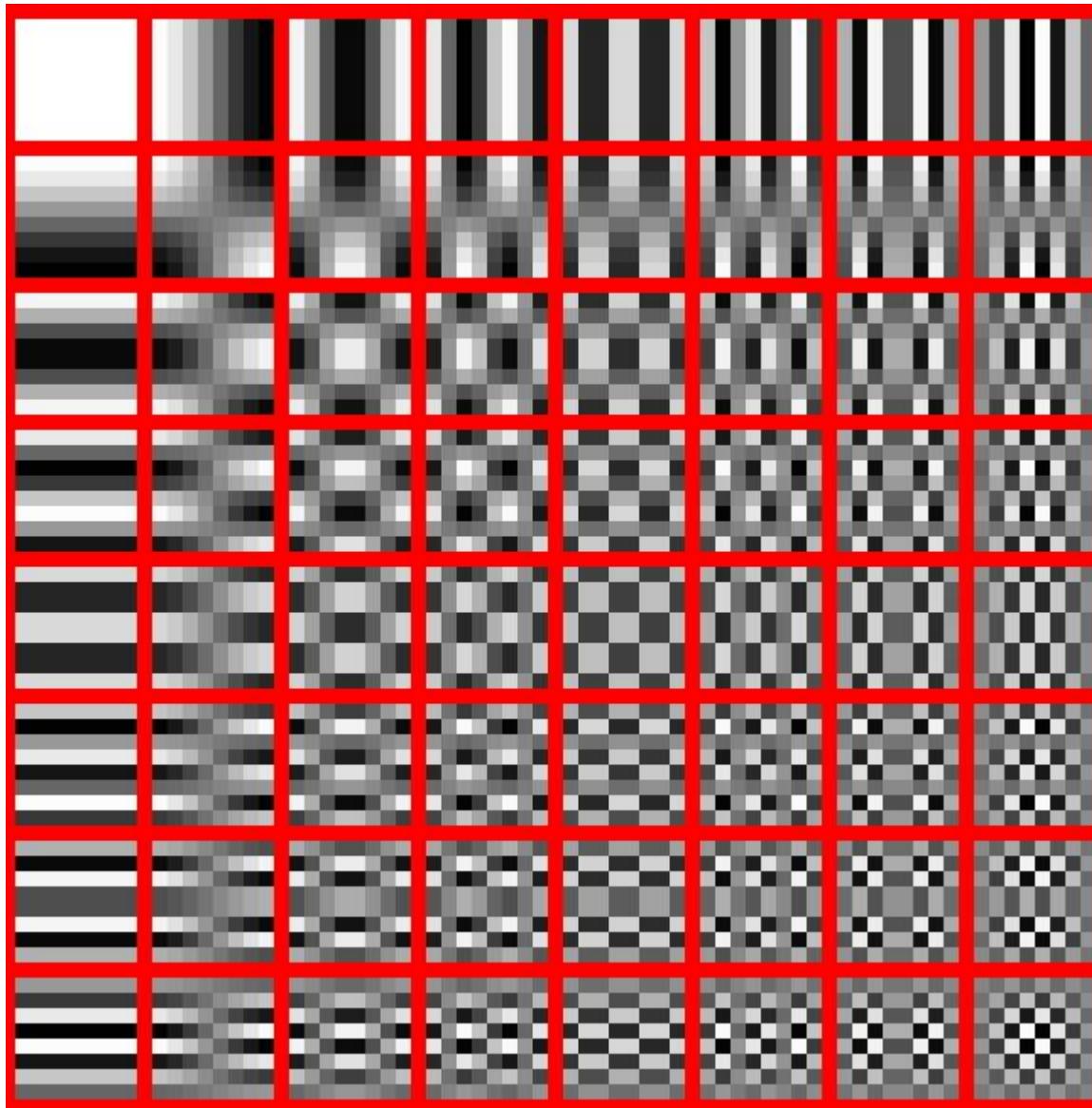


60 most important eigenvectors



Looks like the discrete cosine bases of JPG!...

2D Discrete Cosine Basis



http://en.wikipedia.org/wiki/Discrete_cosine_transform

Dimensionality reduction

- PCA (Principal Component Analysis):
 - Find projection that maximize the variance
- ICA (Independent Component Analysis):
 - Very similar to PCA except that it assumes non-Gaussian features
- Multidimensional Scaling:
 - Find projection that best preserves inter-point distances
- LDA(Linear Discriminant Analysis):
 - Maximizing the component axes for class-separation
- ...
- ...

Principal Components Analysis

some slides from

-Octavia Camps, PSU

[-www.cs.rit.edu/~rsg/BIIS_05lecture7.ppt](http://www.cs.rit.edu/~rsg/BIIS_05lecture7.ppt) tby R.S.Gaborski Professor

-hebb.mit.edu/courses/9.641/lectures/pca.ppt by Sebastian Seung.

Covariance

- Variance and Covariance are a measure of the "spread" of a set of points around their center of mass (mean)
- Variance - measure of the deviation from the mean for points in one dimension e.g. heights
- Covariance as a measure of how much each of the dimensions vary from the mean with respect to each other.
- Covariance is measured between 2 dimensions to see if there is a relationship between the 2 dimensions e.g. number of hours studied & marks obtained.
- The covariance between one dimension and itself is the variance

Covariance

$$\text{covariance } (X, Y) = \frac{\sum_{i=1}^n (\bar{X}_i - X)(\bar{Y}_i - Y)}{(n - 1)}$$

- So, if you had a 3-dimensional data set (x, y, z) , then you could measure the covariance between the x and y dimensions, the y and z dimensions, and the x and z dimensions. Measuring the covariance between x and x , or y and y , or z and z would give you the variance of the x , y and z dimensions respectively.

Covariance Matrix

- Representing Covariance between dimensions as a matrix e.g. for 3 dimensions:

$$C = \begin{bmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{bmatrix}$$

Variances

- Diagonal is the **variances** of x , y and z
- $\text{cov}(x,y) = \text{cov}(y,x)$ hence matrix is **symmetrical** about the diagonal
- N -dimensional data will result in **$N \times N$ covariance matrix**

Covariance

- What is the interpretation of covariance calculations?

e.g.: 2 dimensional data set

x: number of hours studied for a subject

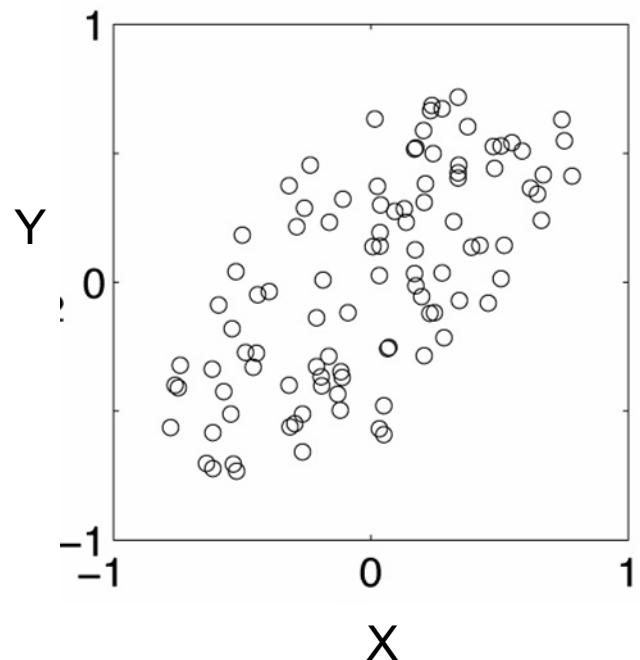
y: marks obtained in that subject

covariance value is say: 104.53

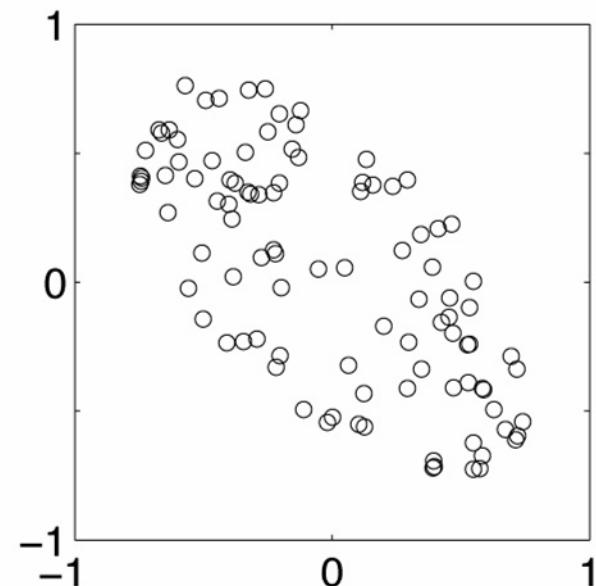
what does this value mean?

Covariance examples

positive covariance



negative covariance



Covariance

- Exact value is not as important as it's sign.
- A positive value of covariance indicates both dimensions increase or decrease together e.g. as the number of hours studied increases, the marks in that subject increase.
- A negative value indicates while one increases the other decreases, or vice-versa e.g. active social life at PSU vs performance in CS dept.
- If covariance is zero: the two dimensions are independent of each other e.g. heights of students vs the marks obtained in a subject

Covariance

- Why bother with calculating covariance when we could just plot the 2 values to see their relationship?

Covariance calculations are used to find relationships between dimensions in high dimensional data sets (usually greater than 3) where visualization is difficult.

PCA

- principal components analysis (PCA) is a technique that can be used to simplify a dataset
- It is a linear transformation that chooses a new coordinate system for the data set such that
 - greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component),
 - the second greatest variance on the second axis,
 - and so on.
- PCA can be used for reducing dimensionality by eliminating the later principal components.

PCA Toy Example

Consider the following 3D points

1	2	4	3	5	6
2	4	8	6	10	12
3	6	12	9	15	18

If each component is stored in a byte,
we need $18 = 3 \times 6$ bytes

PCA Toy Example

Looking closer, we can see that all the points are related geometrically: they are all the same point, scaled by a factor:

$$\begin{array}{c|c} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & = 1 * \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \\ \hline \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \end{array} \quad \begin{array}{c|c} \begin{matrix} 2 \\ 4 \\ 6 \end{matrix} & = 2 * \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \\ \hline \begin{matrix} 4 \\ 8 \\ 12 \end{matrix} & \end{array} \quad \begin{array}{c|c} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & = 4 * \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \\ \hline \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \end{array}$$
$$\begin{array}{c|c} \begin{matrix} 3 \\ 6 \\ 9 \end{matrix} & = 3 * \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \\ \hline \begin{matrix} 3 \\ 6 \\ 9 \end{matrix} & \end{array} \quad \begin{array}{c|c} \begin{matrix} 5 \\ 10 \\ 15 \end{matrix} & = 5 * \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \\ \hline \begin{matrix} 6 \\ 12 \\ 18 \end{matrix} & \end{array} \quad \begin{array}{c|c} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & = 6 * \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \\ \hline \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \end{array}$$

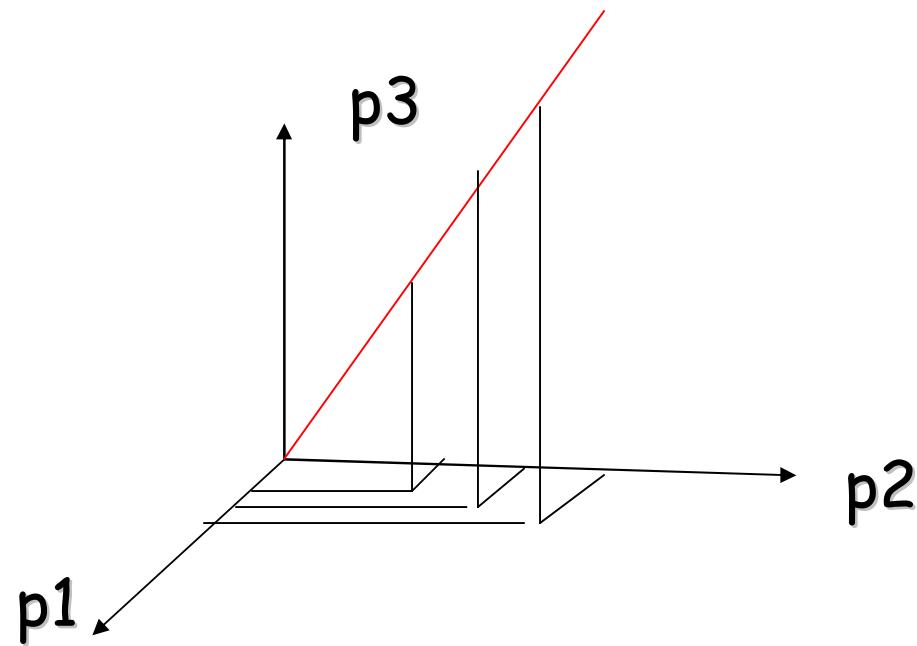
PCA Toy Example

$$\begin{array}{c|c|c|c|c|c} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & = 1 * & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 2 \\ 4 \\ 6 \end{matrix} & = 2 * & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 4 \\ 8 \\ 12 \end{matrix} & = 4 * & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \\ \hline \begin{matrix} 3 \\ 6 \\ 9 \end{matrix} & = 3 * & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 5 \\ 10 \\ 15 \end{matrix} & = 5 * & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 6 \\ 12 \\ 18 \end{matrix} & = 6 * & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \end{array}$$

They can be stored using only 9 bytes (50% savings!):
Store one point (3 bytes) + the multiplying constants (6 bytes)

Geometrical Interpretation:

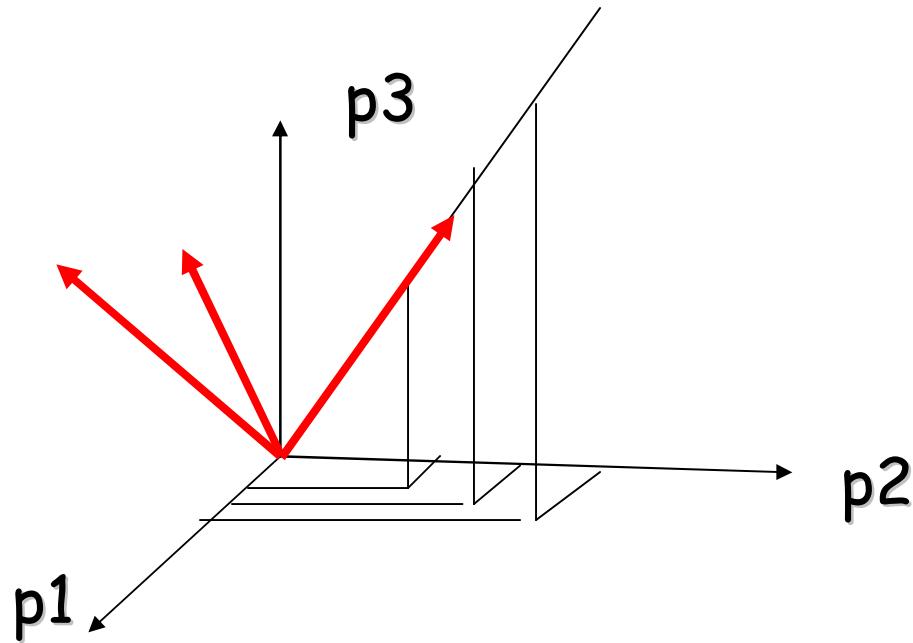
View each point in 3D space.



But in this example, all the points happen to belong to a line: a 1D subspace of the original 3D space.

Geometrical Interpretation:

Consider a new coordinate system where one of the axes is along the direction of the line:



In this coordinate system, every point has only one non-zero coordinate: we only need to store the direction of the line (a 3 bytes image) and the non-zero coordinate for each of the points (6 bytes).

Principal Component Analysis (PCA)

- Given a set of points, how do we know if they can be compressed like in the previous example?
 - The answer is to look into the correlation between the points
 - The tool for doing this is called PCA

PCA

- By finding the eigenvalues and eigenvectors of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset.
- This is the principal component.
- PCA is a useful statistical technique that has found application in:
 - fields such as face recognition and image compression
 - finding patterns in data of high dimension.

PCA Theorem

Let x_1, x_2, \dots, x_n be a set of $n N \times 1$ vectors and let \bar{x} be their average:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iN} \end{bmatrix} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^{i=n} \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iN} \end{bmatrix}$$

.

xbar is the average vector (Vector with each variable the mean from all training data)

PCA Theorem

Let X be the $N \times n$ matrix with columns

$x_1 - \bar{x}, x_2 - \bar{x}, \dots x_n - \bar{x}$:

$$X = \begin{bmatrix} x_1 - \bar{x} & x_2 - \bar{x} & \cdots & x_n - \bar{x} \end{bmatrix}$$

Note: subtracting the mean is equivalent to translating the coordinate system to the location of the mean.

PCA Theorem

Let $Q = X X^T$ be the $N \times N$ matrix:

$$Q = X X^T = \begin{bmatrix} \mathbf{x}_1 - \bar{\mathbf{x}} & \mathbf{x}_2 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}} \end{bmatrix} \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T \end{bmatrix}$$

Notes:

1. Q is square
2. Q is symmetric
3. Q is the covariance matrix [aka scatter matrix]
4. Q can be very large (in vision, N is often the number of pixels in an image!)

PCA Theorem

Theorem:

Each x_j can be written as:
$$x_j = \bar{x} + \sum_{i=1}^{i=n} g_{ji} e_i$$

where e_i are the n eigenvectors of Q with non-zero eigenvalues.

Notes:

1. The eigenvectors $e_1 e_2 \dots e_n$ span an eigenspace
2. $e_1 e_2 \dots e_n$ are $N \times 1$ orthonormal vectors (directions in N-Dimensional space)
3. The scalars g_{ji} are the coordinates of x_j in the space.

$$g_{ji} = (x_j - \bar{x}) \cdot e_i$$

Using PCA to Compress Data

- Expressing x in terms of $e_1 \dots e_n$ has not changed the size of the data
- However, if the points are highly correlated many of the coordinates of x will be zero or close to zero.

note: this means they lie in a lower-dimensional linear subspace

Using PCA to Compress Data

- Sort the eigenvectors e_i according to their eigenvalue:

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_n$$

- Assuming that $\lambda_i \approx 0$ if $i > k$

- Then

$$\mathbf{x}_j \approx \bar{\mathbf{x}} + \sum_{i=1}^{i=k} g_{ji} \mathbf{e}_i$$

PCA Example -STEP 1

<http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf>

- DATA:
- | x | y |
|-----|-----|
| 2.5 | 2.4 |
| 0.5 | 0.7 |
| 2.2 | 2.9 |
| 1.9 | 2.2 |
| 3.1 | 3.0 |
| 2.3 | 2.7 |
| 2 | 1.6 |
| 1 | 1.1 |
| 1.5 | 1.6 |
| 1.1 | 0.9 |

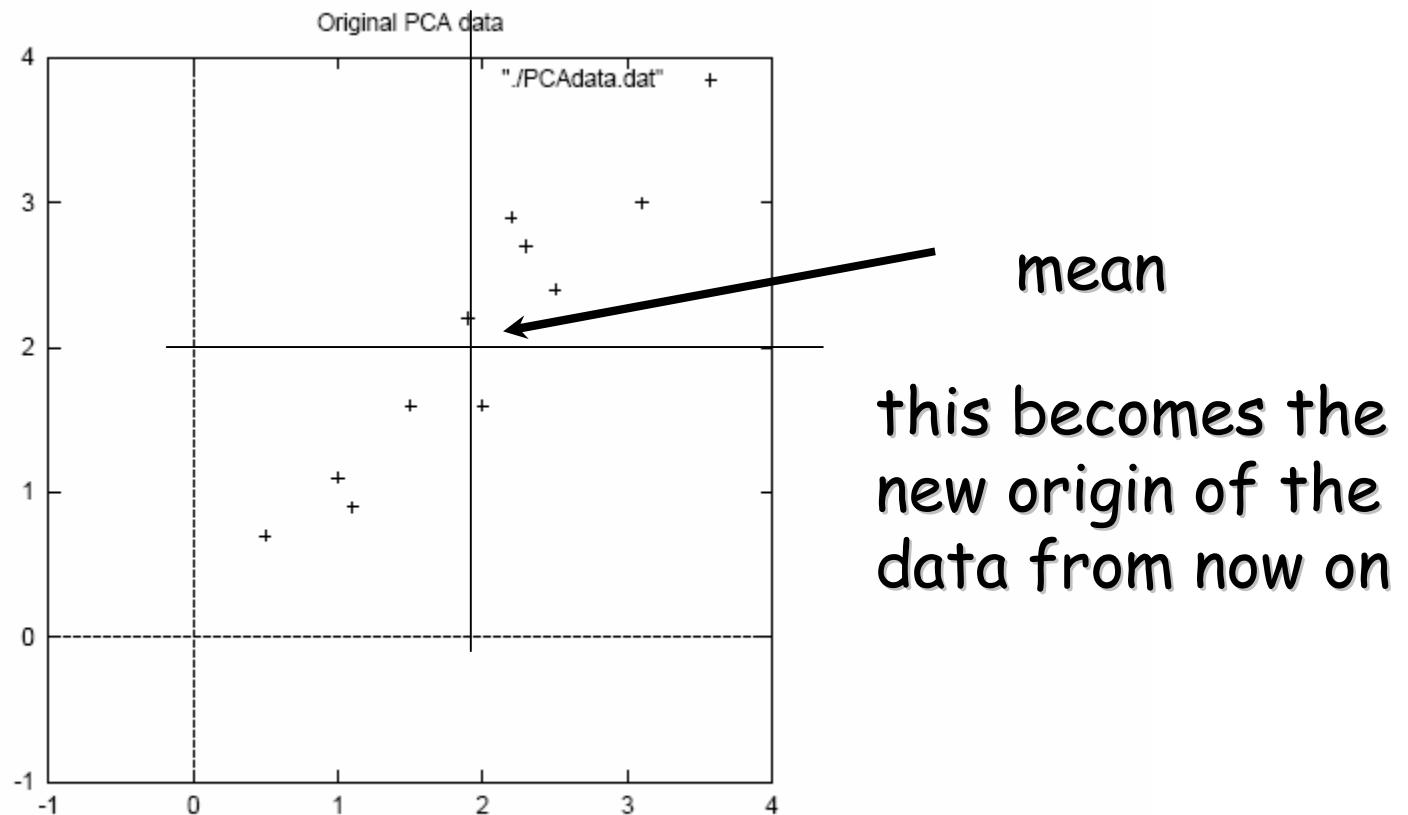


Figure 3.1: PCA example data, original data on the left, data with the means subtracted on the right, and a plot of the data

PCA Example -STEP 2

- Calculate the covariance matrix

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

- since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variable increase together.

PCA Example -STEP 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix

$$\text{eigenvalues} = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -735178656 \end{pmatrix}$$

PCA Example -STEP 3

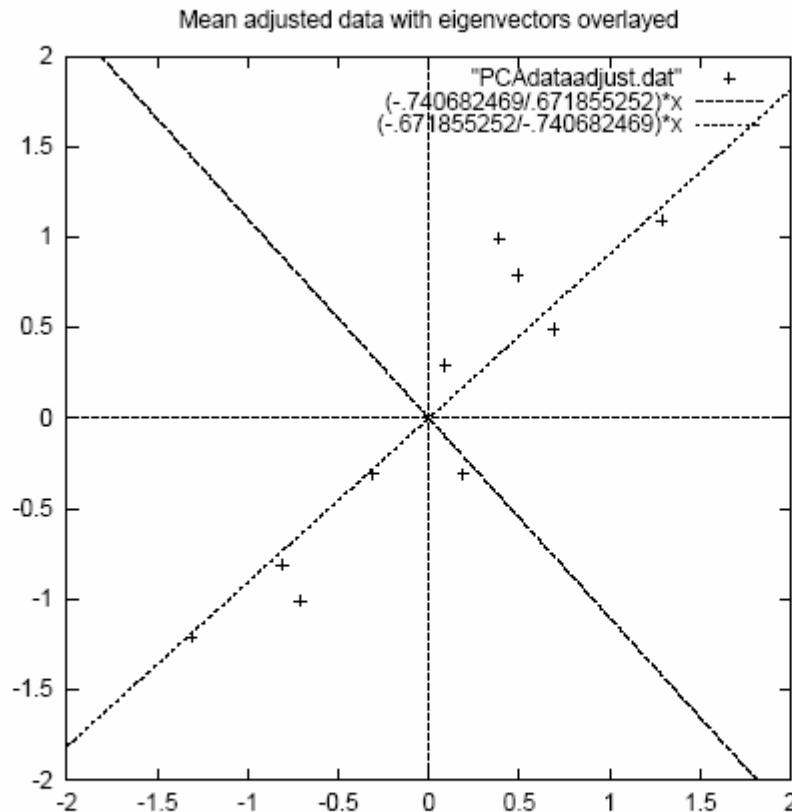


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

- eigenvectors are plotted as diagonal dotted lines on the plot.
- Note they are perpendicular to each other.
- Note one of the eigenvectors goes through the middle of the points, like drawing a line of best fit.
- The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

PCA Example -STEP 4

- Feature Vector

FeatureVector = (eig_1 eig_2 $eig_3 \dots eig_n$)

We can either form a feature vector with both of the eigenvectors:

$$\begin{bmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{bmatrix}$$

or, we can choose to leave out the smaller, less significant component and only have a single column:

$$\begin{bmatrix} -.677873399 \\ -.735178656 \end{bmatrix}$$

PCA Example -STEP 5

- Deriving new data coordinates

FinalData = RowFeatureVector \times RowZeroMeanData

RowFeatureVector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top

RowZeroMeanData is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

Note: this is essential Rotating the coordinate axes so higher-variance axes come first.

PCA Example -STEP 5

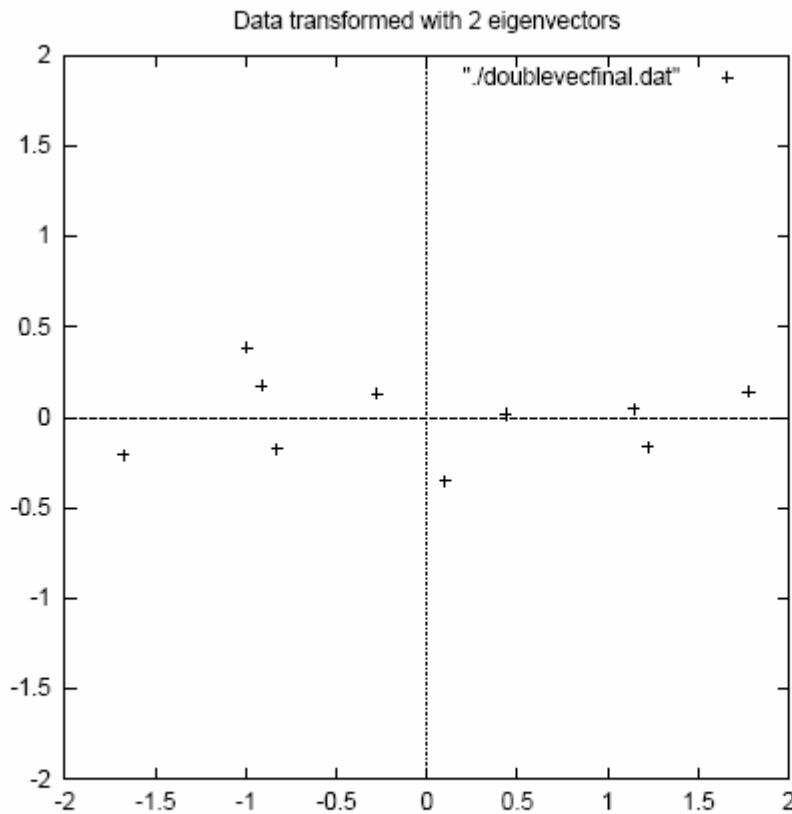
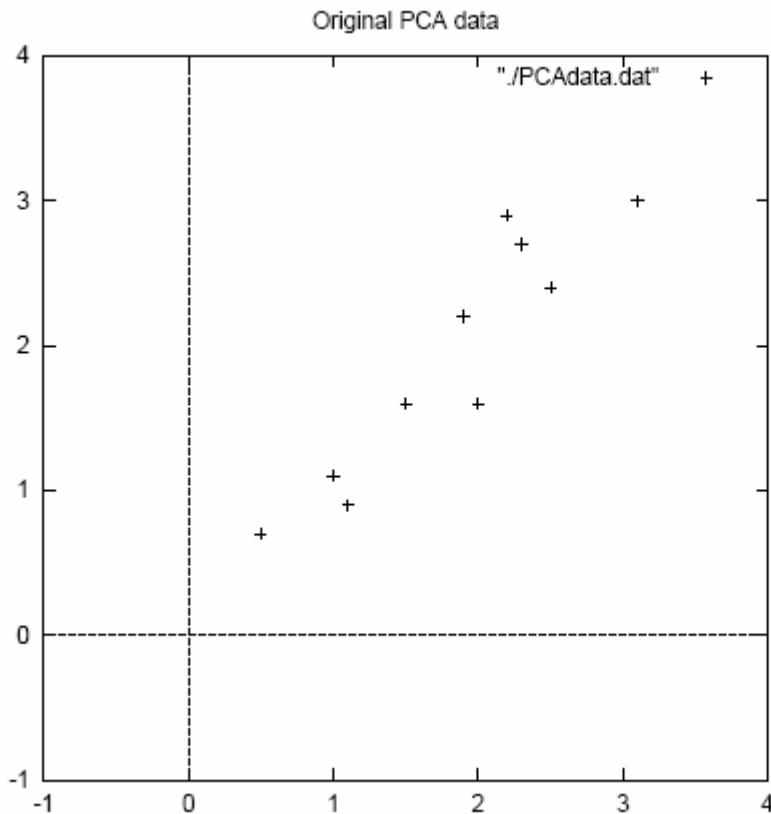


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

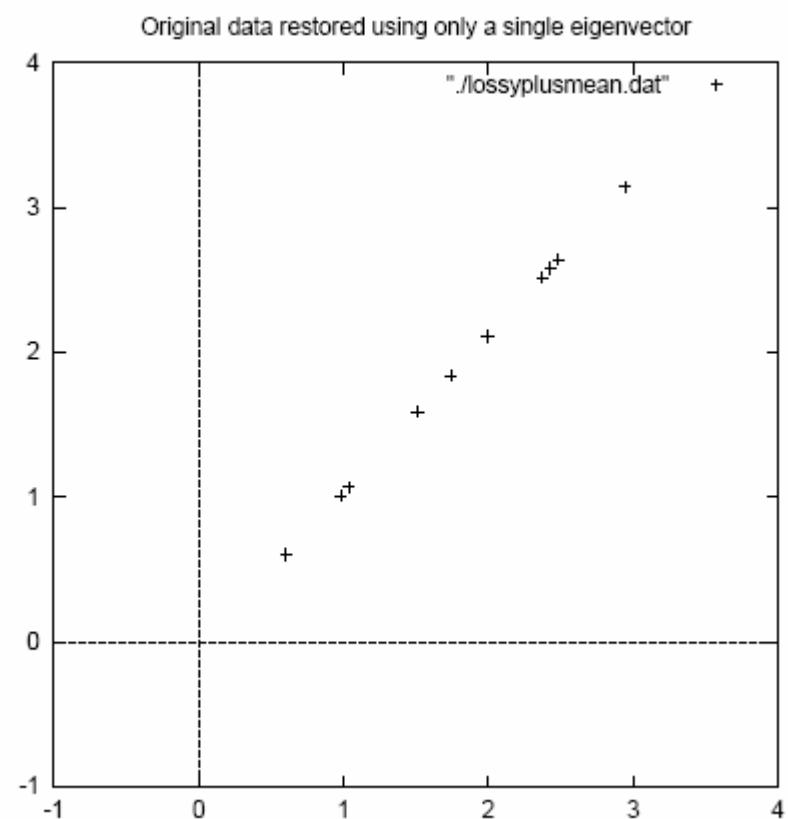
PCA Example : Approximation

- If we reduced the dimensionality, obviously, when reconstructing the data we would lose those dimensions we chose to discard. In our example let us assume that we considered only the x dimension...

PCA Example : Final Approximation



2D point cloud

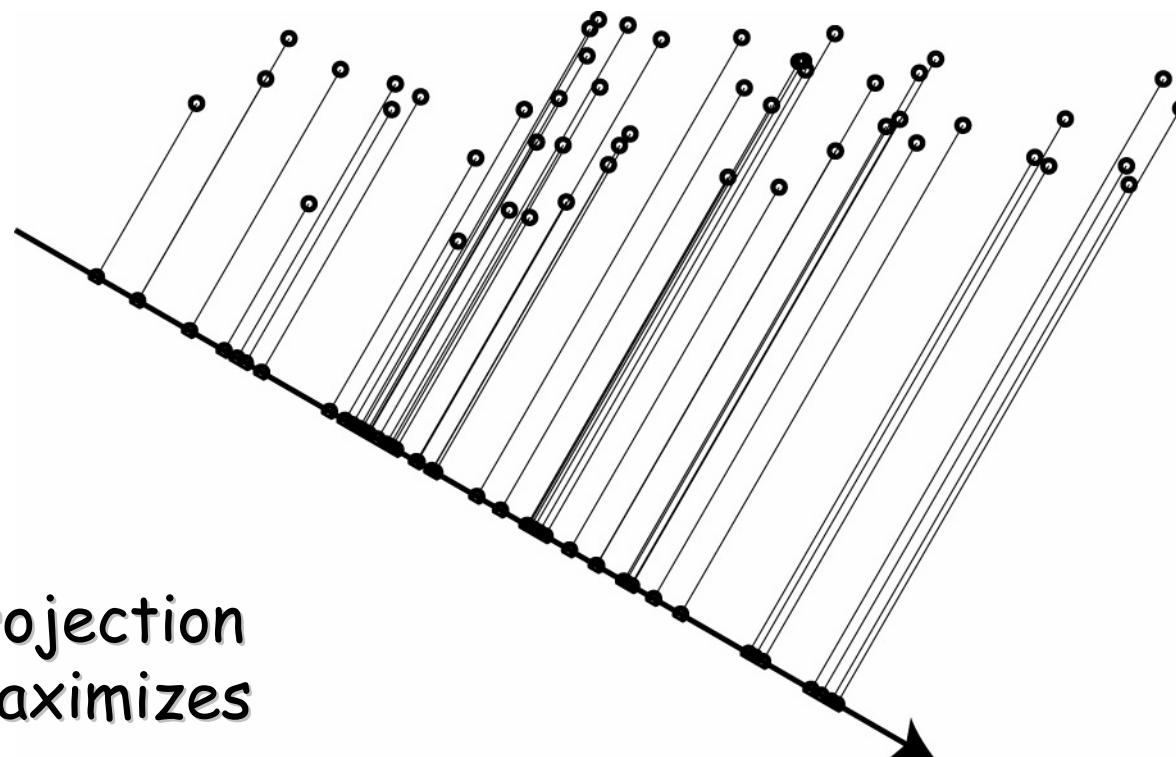


Approximation using
one eigenvector basis

Another way of thinking about Principal component

- direction of maximum variance in the input space
- happens to be same as the principal eigenvector of the covariance matrix

One-dimensional projection



find projection
that maximizes
variance

Covariance to variance

- From the covariance, the variance of any projection can be calculated.
- Let w be a unit vector

$$\begin{aligned}\langle (w^T x)^2 \rangle - \langle w^T x \rangle^2 &= w^T C w \\ &= \sum_{ij} w_i C_{ij} w_j\end{aligned}$$

Maximizing variance

- Principal eigenvector of C
 - the one with the largest eigenvalue.

$$w^* = \arg \max_{w:|w|=1} w^T C w$$

$$\begin{aligned}\lambda_{\max}(C) &= \max_{w:|w|=1} w^T C w \\ &= w^{*T} C w^*\end{aligned}$$

Implementing PCA

- Need to find “first” k eigenvectors of Q :

$$Q = XX^T = \begin{bmatrix} \mathbf{x}_1 - \bar{\mathbf{x}} & \mathbf{x}_2 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}} \end{bmatrix} \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T \end{bmatrix}$$

Q is $N \times N$ (Again, N could be the number of pixels in an image. For a 256×256 image, $N = 65536$!!)
Don't want to explicitly compute Q !!!!

Singular Value Decomposition (SVD)

Any $m \times n$ matrix X can be written as the product of 3 matrices:

$$X = UDV^T$$

Where:

- U is $m \times m$ and its columns are orthonormal vectors
- V is $n \times n$ and its columns are orthonormal vectors
- D is $m \times n$ diagonal and its diagonal elements are called the singular values of X , and are such that:

$$\sigma_1, \sigma_2, \dots, \sigma_n, 0$$

SVD Properties

$$X = UDV^T$$

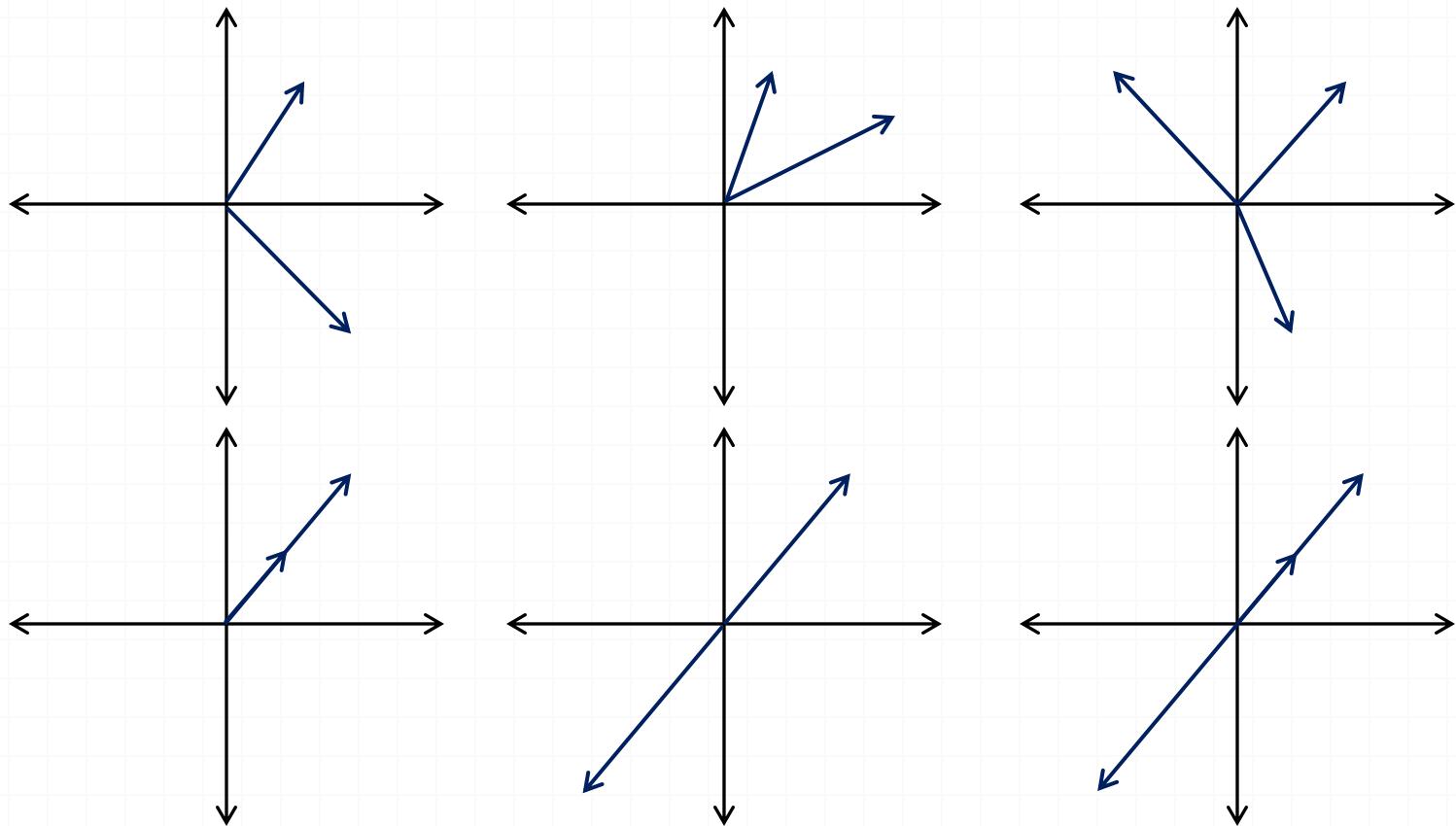
- The columns of U are the eigenvectors of XX^T
- The columns of V are the eigenvectors of X^TX
- The squares of the diagonal elements of D are the eigenvalues of XX^T and X^TX

Feature Extraction (PCA+LDA)

Mathematical Tools (Recall)

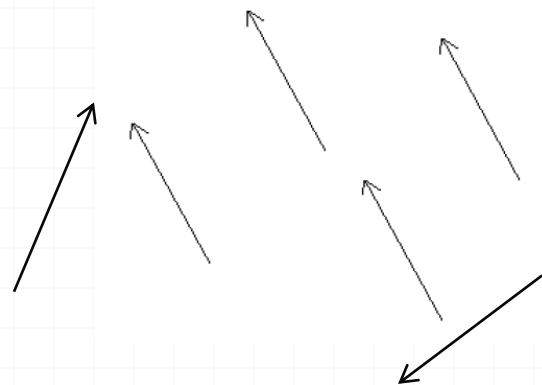
Linear Independence

- Different direction vectors are independent.
- Parallel vectors are dependent. (+ve/-ve directions are parallel)

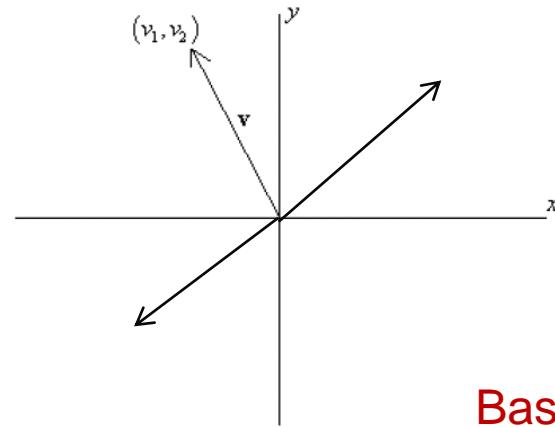


Basis & Orthonormal Bases

o Basis (or axes): frame of reference



vs



$$\begin{bmatrix} 2 \\ 3 \end{bmatrix} = 2 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad \begin{bmatrix} -9 \\ 6 \end{bmatrix} = -9 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 6 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

Basis
 $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$ is a set
of representative for \mathbb{R}^2

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \frac{9}{7} \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \frac{4}{7} \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix}; \quad \begin{bmatrix} -9 \\ 6 \end{bmatrix} = -3 \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix} + 3 \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \end{bmatrix} \right\}$$

is another
Set of representative for \mathbb{R}^2

Basis & Orthonormal Bases

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix} = 2 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix};$$

$$\begin{bmatrix} -9 \\ 6 \end{bmatrix} = -9 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 6 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -9 \\ 6 \end{bmatrix};$$

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \frac{9}{7} \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \frac{4}{7} \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 9/7 \\ 4/7 \end{bmatrix};$$

Linear
Transformation

$$\begin{bmatrix} -9 \\ 6 \end{bmatrix} = -3 \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix} + 3 \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} -3 \\ 3 \end{bmatrix};$$

Basis & Orthonormal Bases

Basis: a space is totally defined by a set of vectors – any vector is a *linear combination* of the basis

Ortho-Normal: *orthogonal* + *normal*

[Sneak peek: **Orthogonal**: dot product is zero
Normal: magnitude is one]

$$\vec{x} = [1 \quad 0 \quad 0]^T$$

$$\vec{y} = [0 \quad 1 \quad 0]^T$$

$$\vec{z} = [0 \quad 0 \quad 1]^T$$

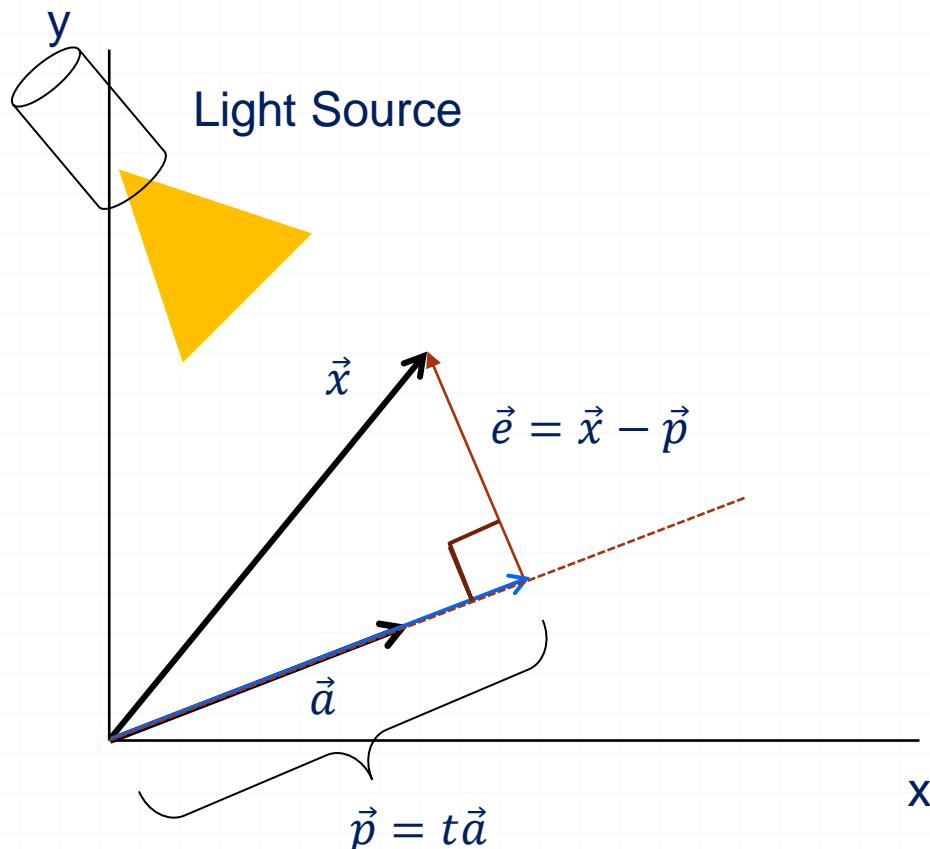
$$\vec{x} \cdot \vec{y} = 0$$

$$\vec{x} \cdot \vec{z} = 0$$

$$\vec{y} \cdot \vec{z} = 0$$

Projection: Using Inner Products

Projection of \vec{x} along the direction \vec{a}



$$\cos 90^\circ = \frac{\vec{a} \cdot \vec{e}}{\|\vec{a}\| \|\vec{e}\|}$$
$$\Rightarrow \vec{a} \cdot \vec{e} = 0$$
$$\Rightarrow \vec{a} \cdot (\vec{x} - \vec{p}) = 0$$
$$\Rightarrow \vec{a} \cdot (\vec{x} - t\vec{a}) = 0$$
$$\Rightarrow \vec{a} \cdot \vec{x} - t\vec{a} \cdot \vec{a} = 0$$
$$\Rightarrow t = \frac{\vec{a} \cdot \vec{x}}{\vec{a} \cdot \vec{a}}$$

Projection vector $\vec{p} = t\vec{a}$

$$\vec{p} = \vec{a} \frac{\vec{a}^T \vec{x}}{\vec{a}^T \vec{a}} = \frac{\vec{a} \vec{a}^T}{\vec{a}^T \vec{a}} \vec{x}$$

Projection Matrix

Eigenvalues & Eigenvectors

0 Eigenvectors (for a square $m \times m$ matrix \mathbf{S})

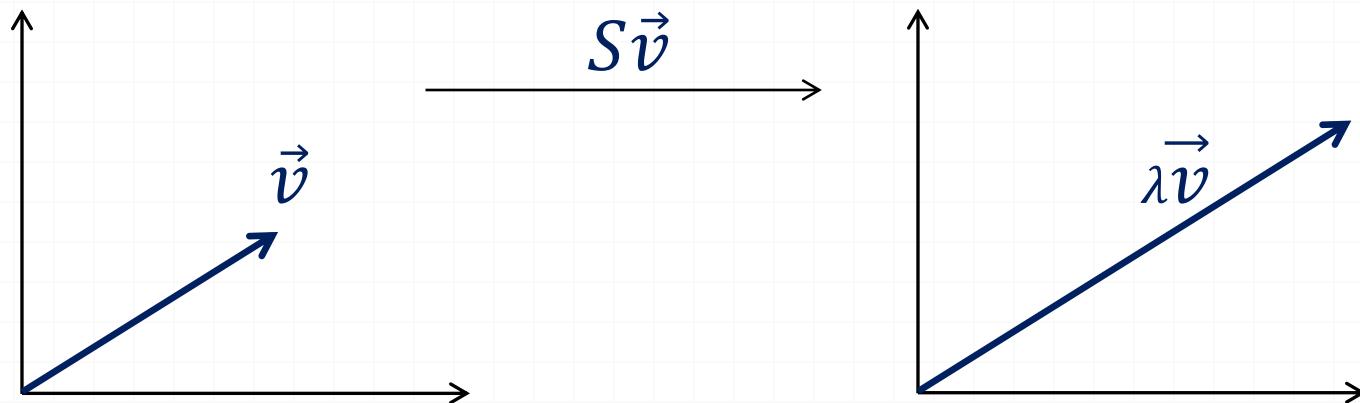
$$S\vec{v} = \lambda\vec{v}$$

(right) eigenvector $\vec{v} \in \mathbb{R}^m \neq \vec{0}$

eigenvalue $\lambda \in \mathbb{R}$

Example

$$\begin{pmatrix} 6 & -2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$



Eigenvalues & Eigenvectors (Properties)

0 The eigenvalues of real symmetric matrices are real.

$$A\vec{v} = \lambda\vec{v}$$

- Then we can conjugate to get $\bar{A}\bar{\vec{v}} = \bar{\lambda}\bar{\vec{v}}$
- If the entries of A are real, this becomes $A\bar{\vec{v}} = \bar{\lambda}\bar{\vec{v}}$
- This proves that complex eigenvalues of real valued matrices come in conjugate pairs
- Now transpose to get $\bar{\vec{v}}^t A^T = \bar{\vec{v}}^t \bar{\lambda}$. Because A is symmetric matrix we now have $\bar{\vec{v}}^t A = \bar{\vec{v}}^t \bar{\lambda}$
- Multiply both sides of this equation on the right with \vec{v} , i.e. $\bar{\vec{v}}^t A \vec{v} = \bar{\vec{v}}^t \bar{\lambda} \vec{v}$
- On the other hand multiply $A\vec{v} = \lambda\vec{v}$ on the left by $\bar{\vec{v}}^t$ to get $\bar{\vec{v}}^t A \vec{v} = \bar{\vec{v}}^t \lambda \vec{v}$
$$\Rightarrow \bar{\vec{v}}^t \bar{\lambda} \vec{v} = \bar{\vec{v}}^t \lambda \vec{v} \Rightarrow \bar{\lambda} = \lambda \Rightarrow \lambda \text{ is real}$$

Eigenvalues & Eigenvectors (Properties)

- o If A is an $n \times n$ symmetric matrix, then any two eigenvectors that come from distinct eigenvalues are orthogonal.

$$A\vec{v}_i = \lambda_i \vec{v}_i$$

- Left multiply with \vec{v}_j^T to $A\vec{v}_i = \lambda_i \vec{v}_i \Rightarrow \vec{v}_j^T A \vec{v}_i = \vec{v}_j^T \lambda_i \vec{v}_i$
- Similarly $\vec{v}_i^T A \vec{v}_j = \vec{v}_i^T \lambda_j \vec{v}_j$
- From the above two equations $(\lambda_j - \lambda_i) \vec{v}_i^T \vec{v}_j = 0$

$$\Rightarrow \vec{v}_i^T \vec{v}_j = 0$$

$\therefore \vec{v}_j$ and \vec{v}_i are perpendicular

Diagonalization

- o Stack up all the eigen vectors to get

$$AV = V\Lambda$$

Where $V = [\vec{v}_1 \quad \vec{v}_2 \quad \dots \quad \vec{v}_n]$; $\Lambda = \text{diag}(\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_n)$

- o If all eigenvectors are linearly independent , V is invertible.

$$A = V\Lambda V^{-1}$$

- o Suppose A is symmetric matrix, then $V^T = V^{-1}$

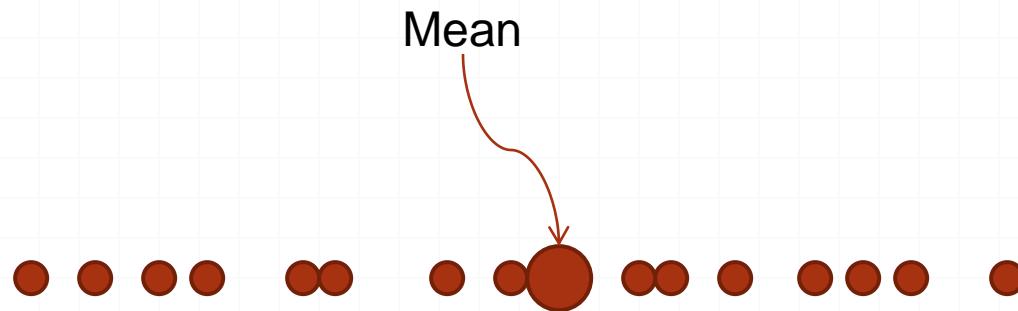
Therefore $A = V\Lambda V^T$

Variance

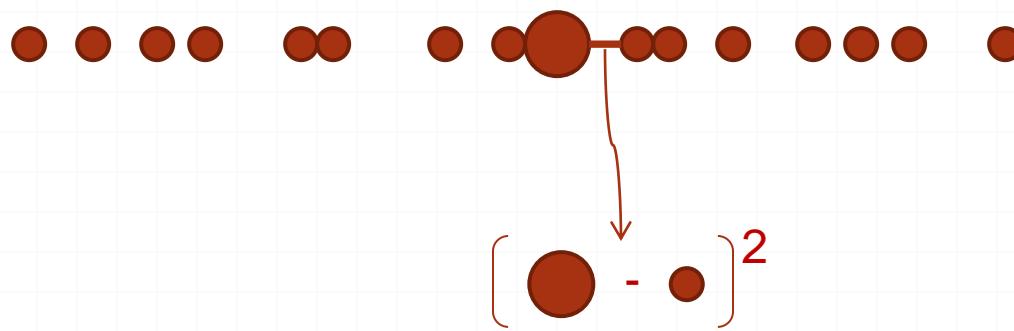
Variance is the average squared deviation from the mean of a set of data. It is used to find the **standard deviation**.



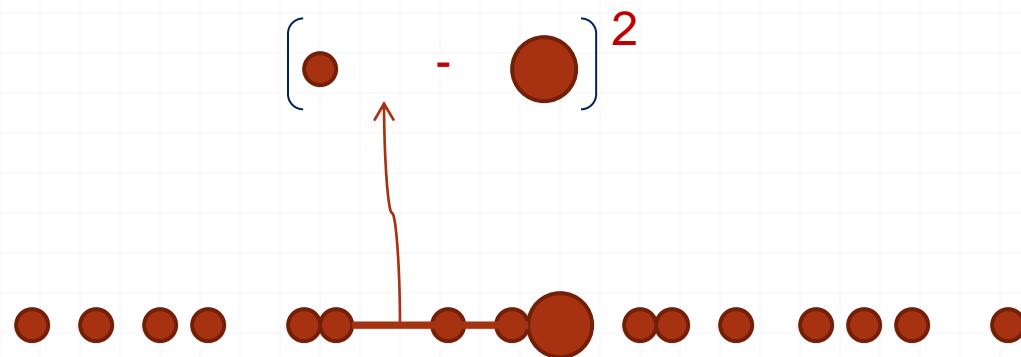
Variance



Variance



Variance



$$(\textcolor{brown}{\bullet} - \bullet)^2$$

Variance



$$\frac{1}{\text{No. of Data Points}} \left(\dots + (\bullet - \bar{\bullet})^2 + (\bar{\bullet} - \bullet)^2 + \dots \right)$$

Variance Formula

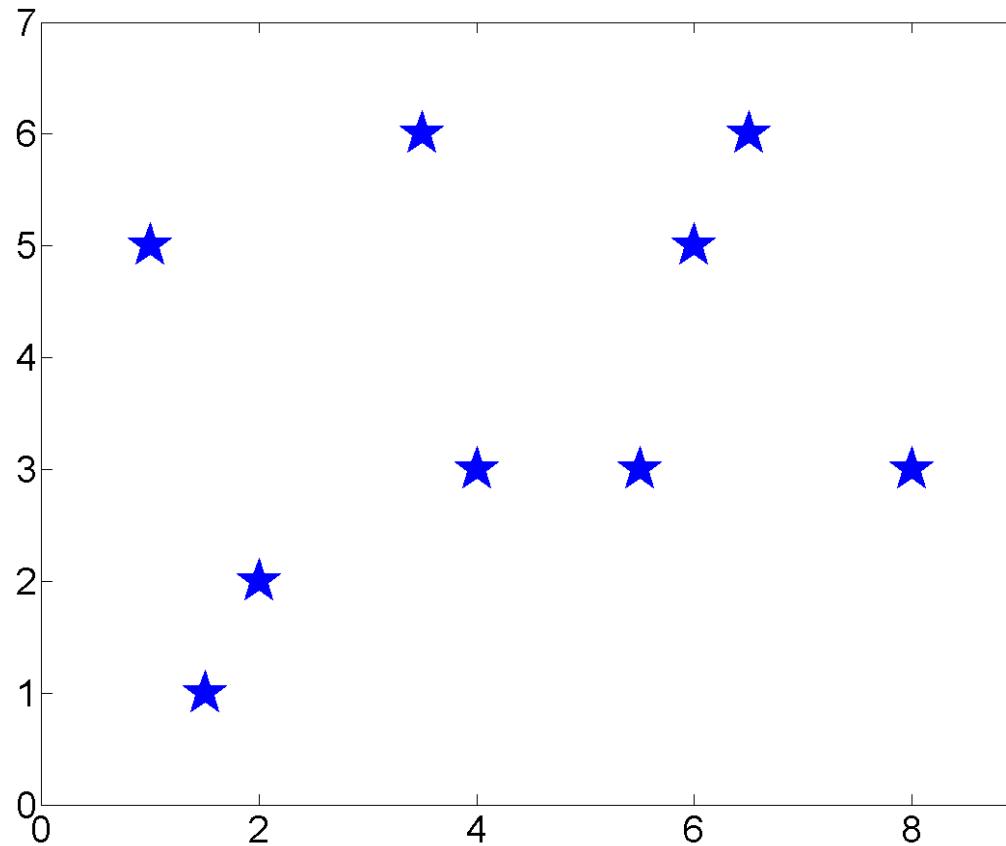
$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Standard Deviation

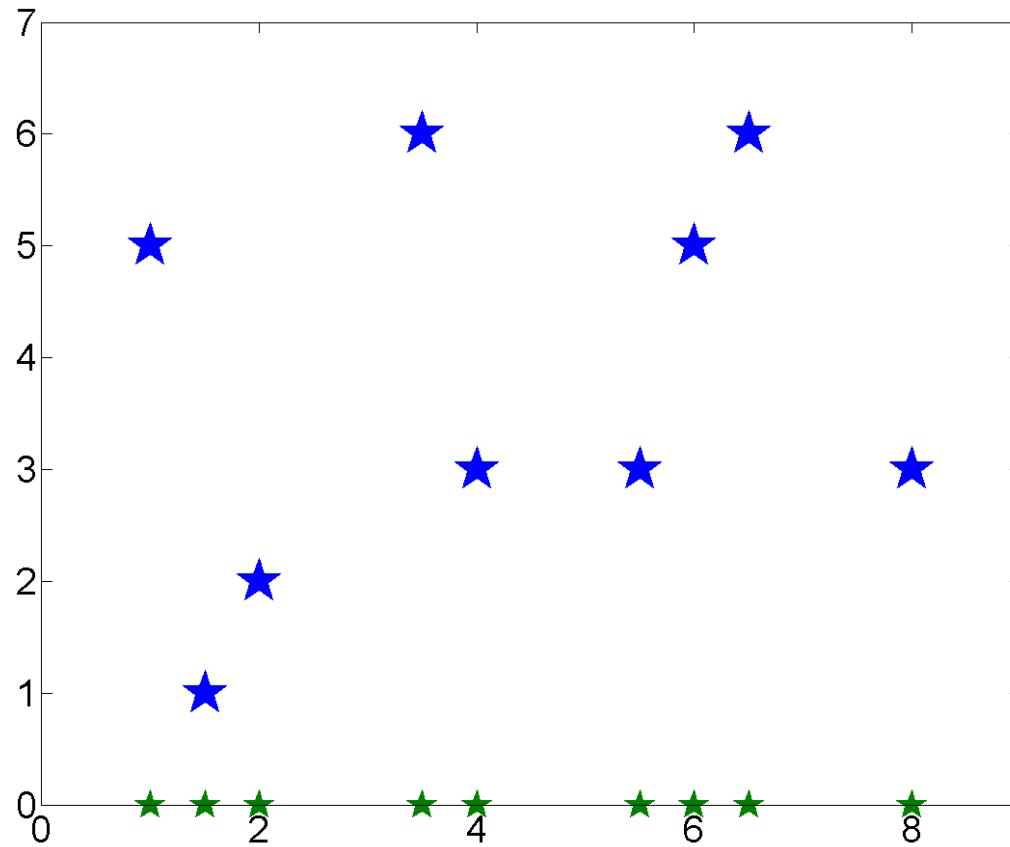
$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

[standard deviation = square root of the variance]

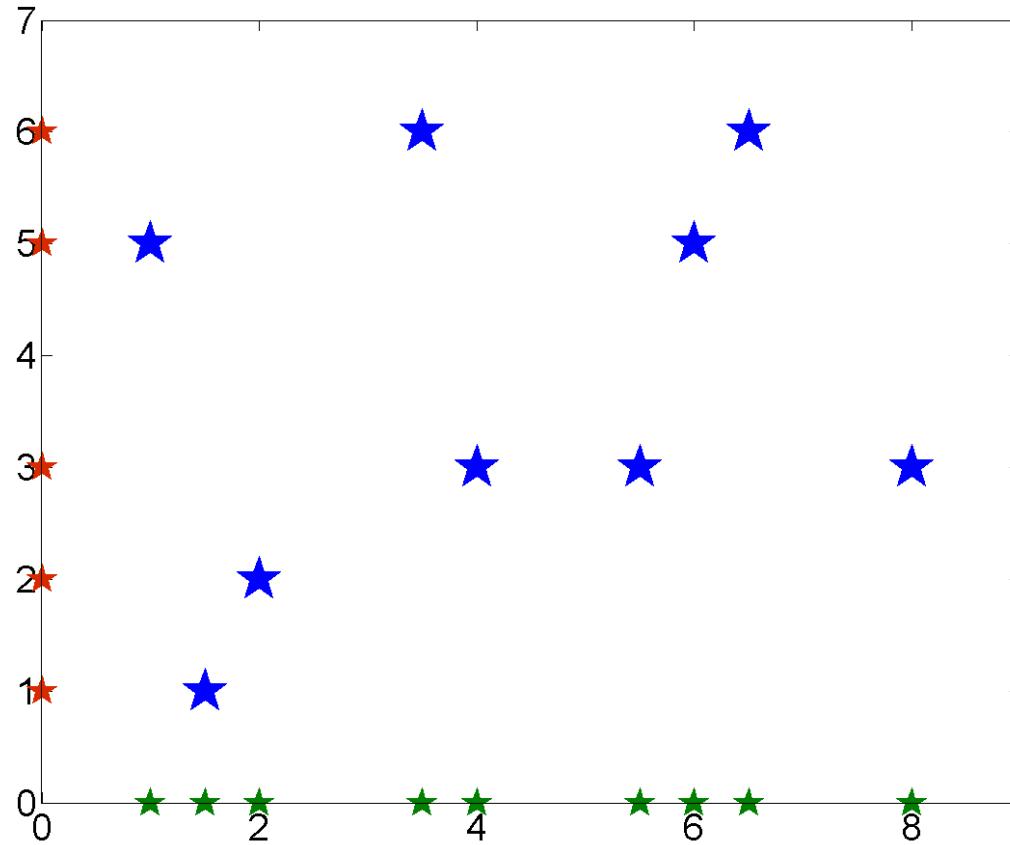
Variance (2D)



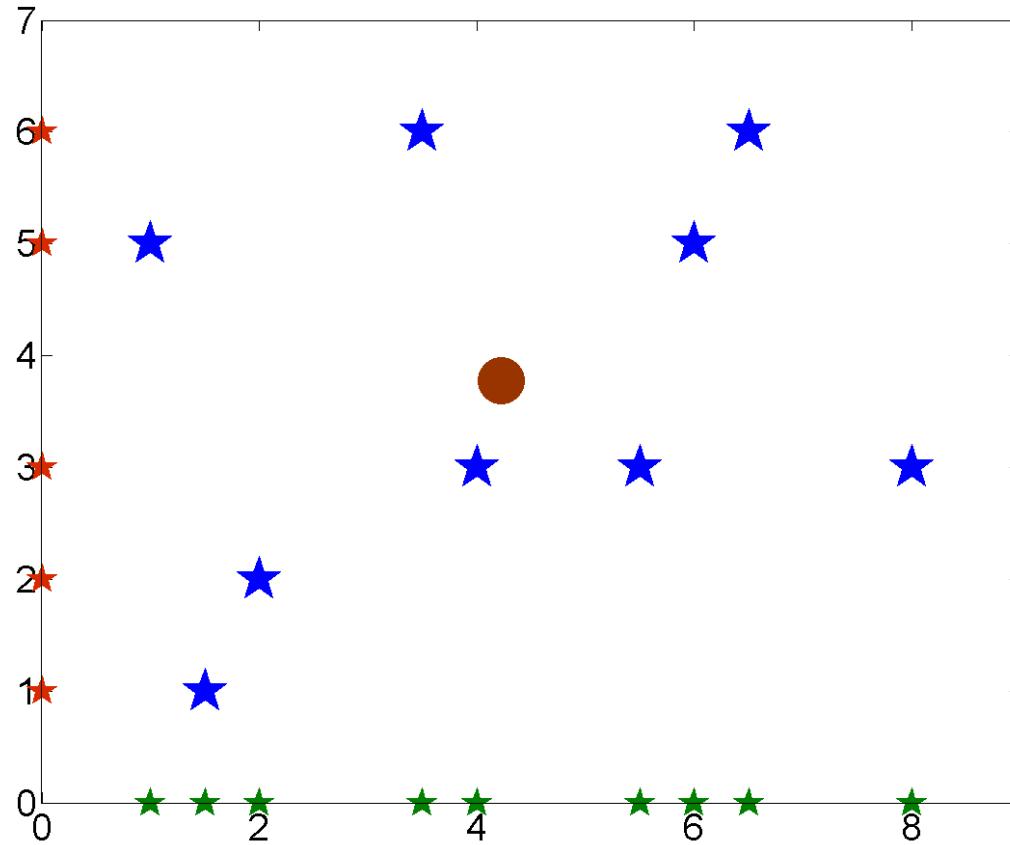
Variance (2D)



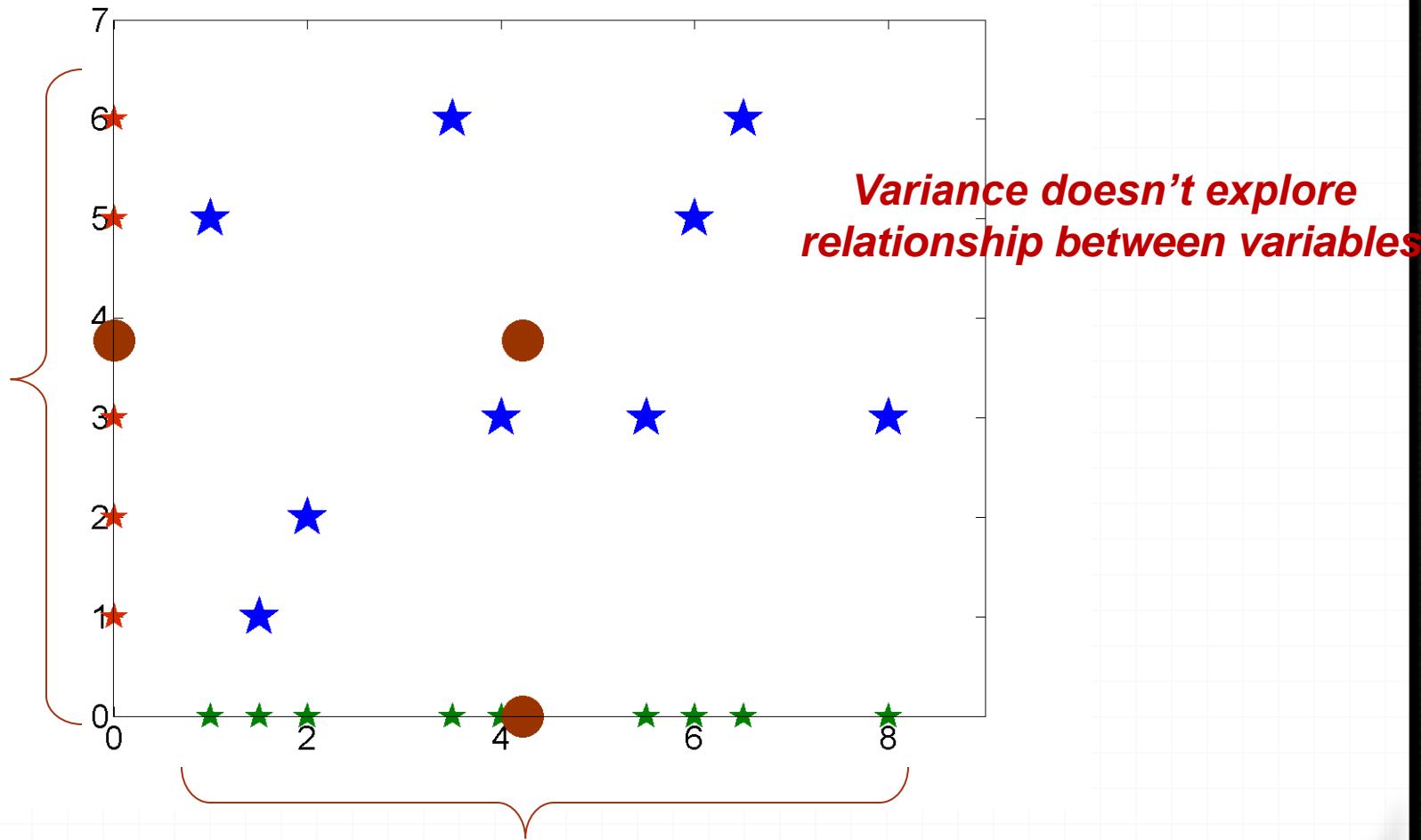
Variance (2D)



Variance (2D)



Variance (2D)



Covariance

$$\text{Variance}(x) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

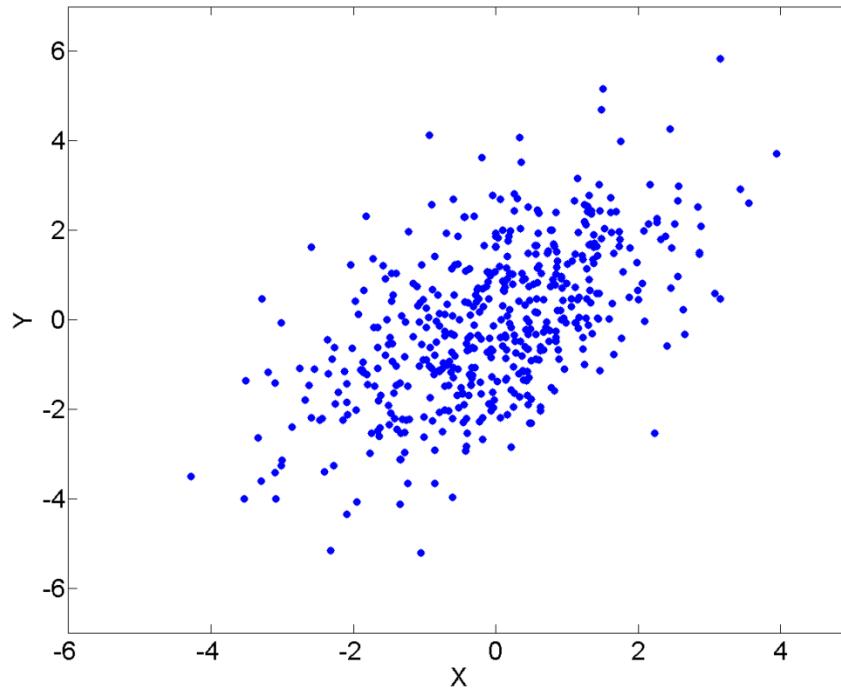
$$= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})$$

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

- ❖ Covariance(x, x) = var(x)
- ❖ Covariance(x, y) = Covariance(y, x)

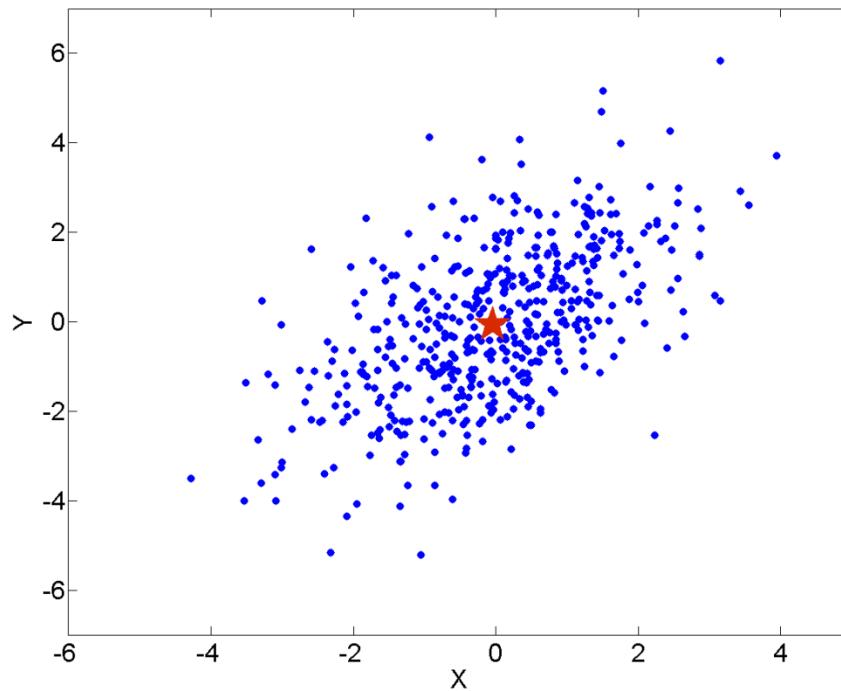
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



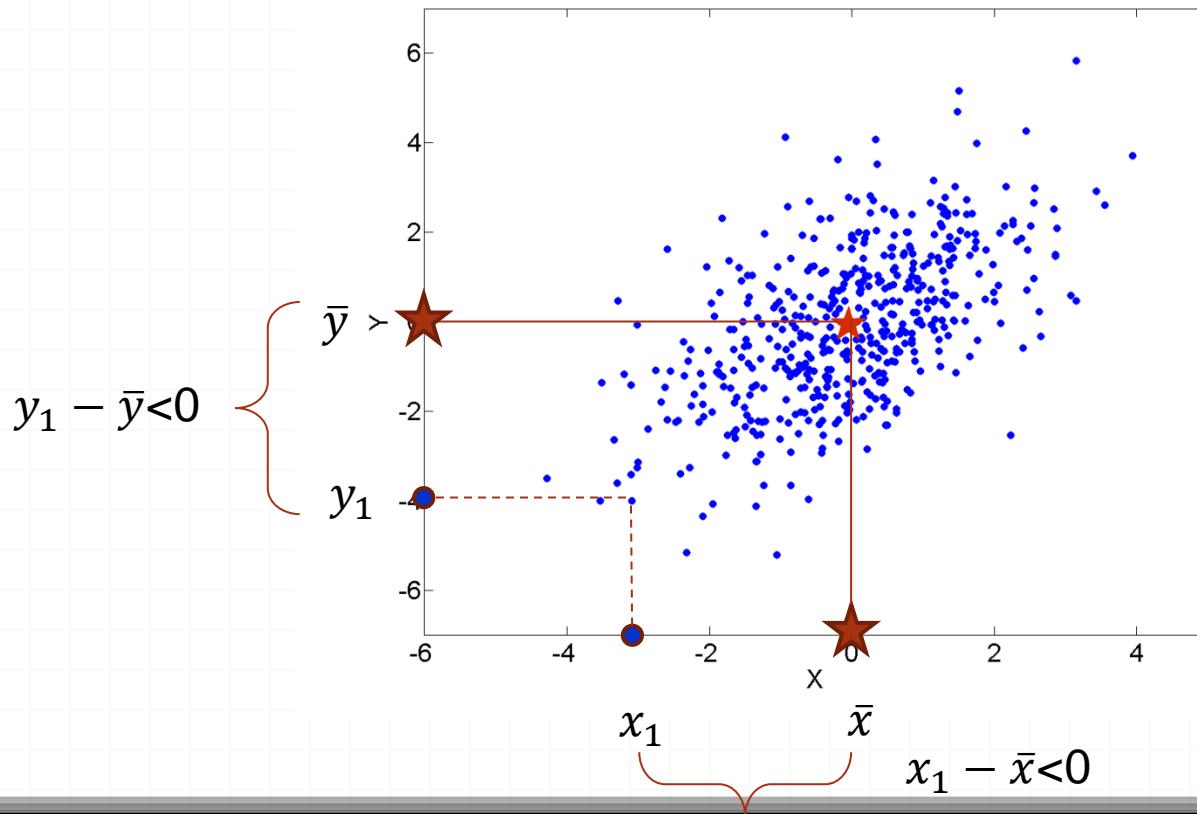
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



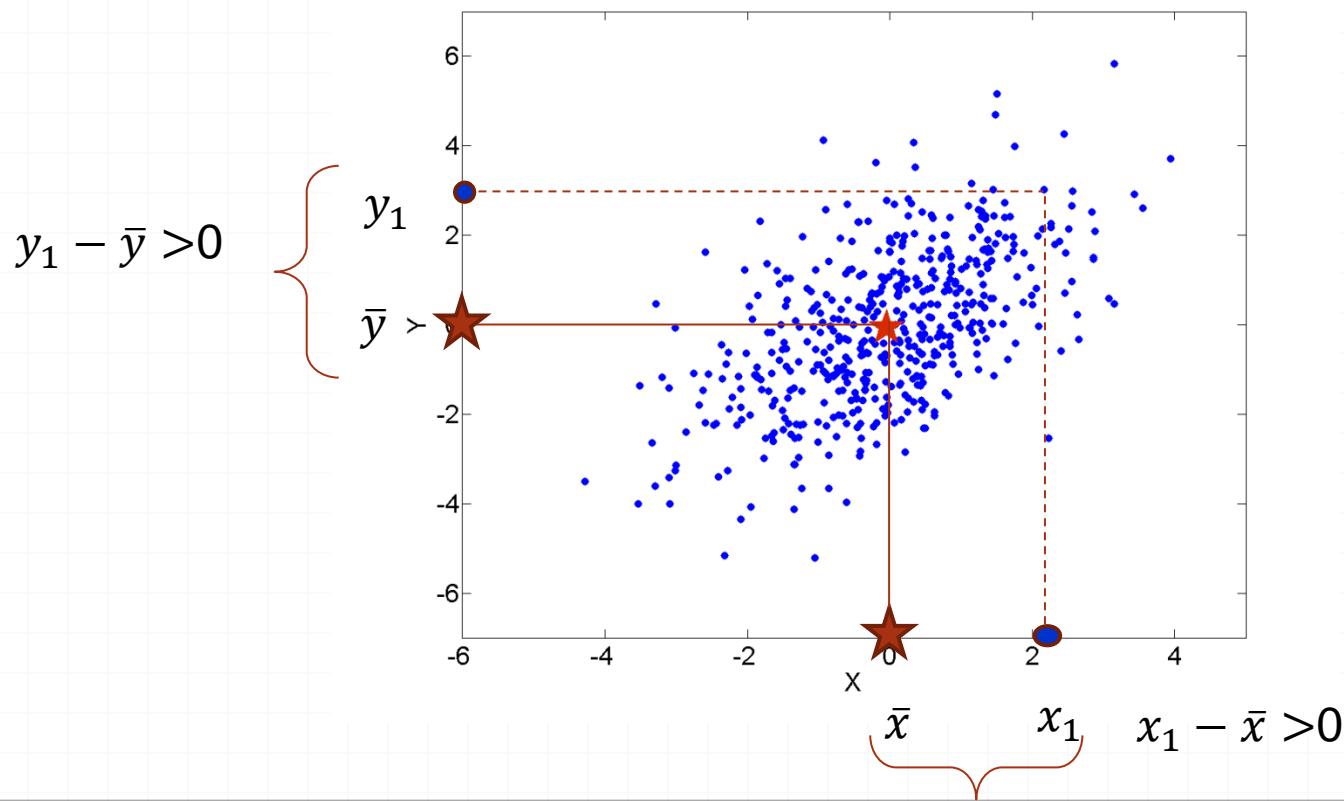
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



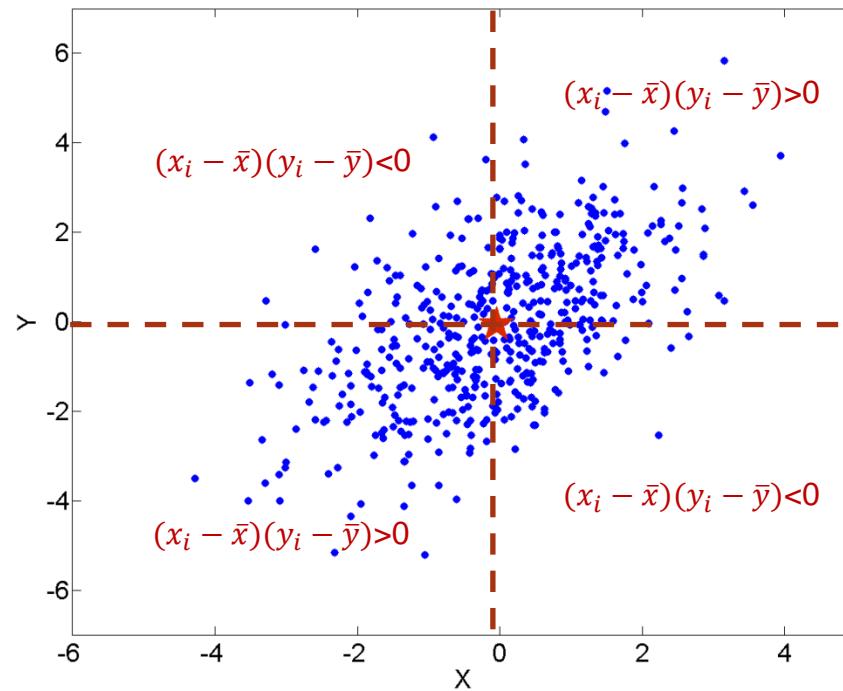
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



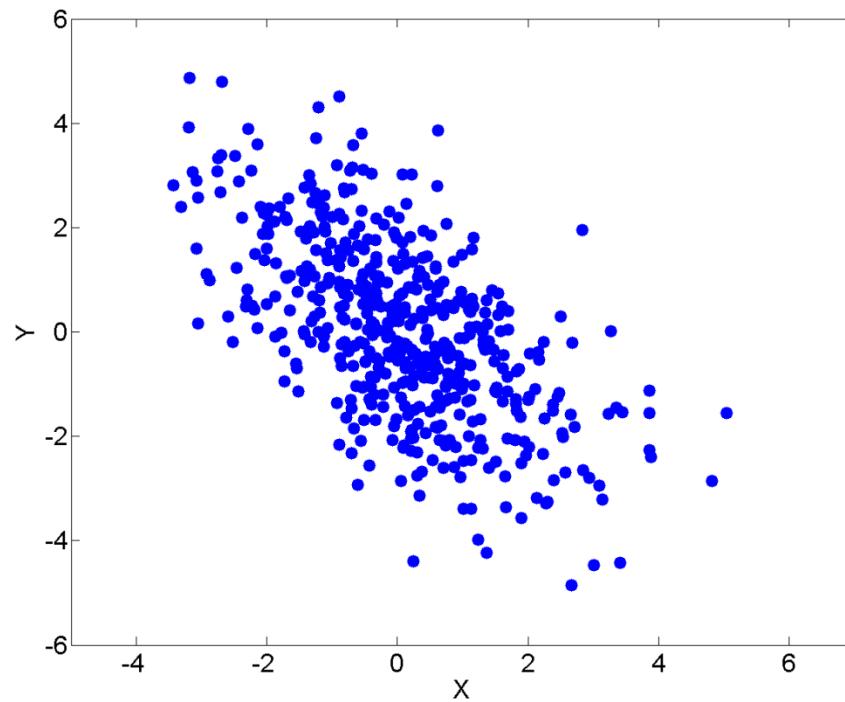
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



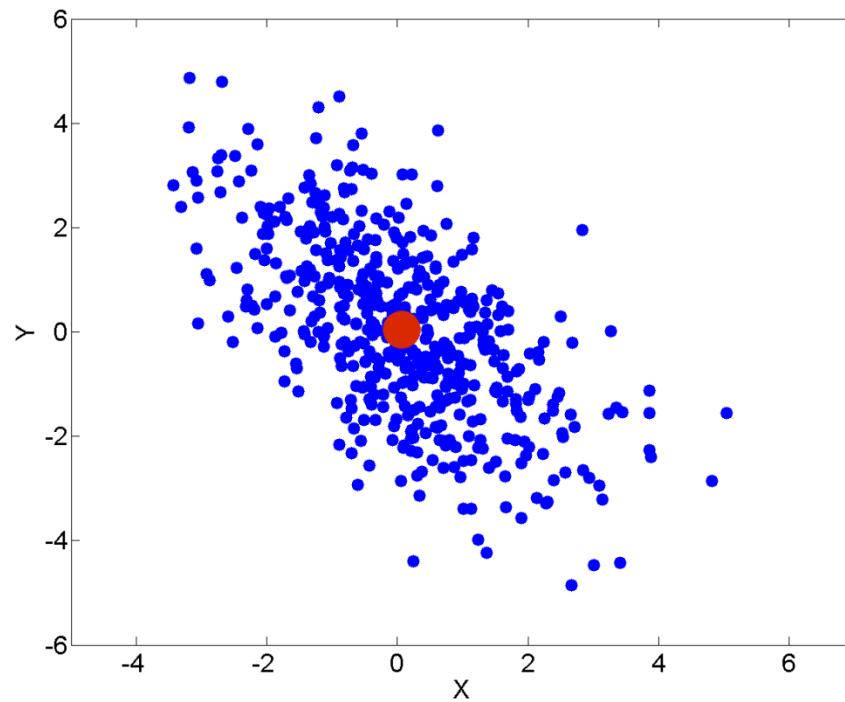
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



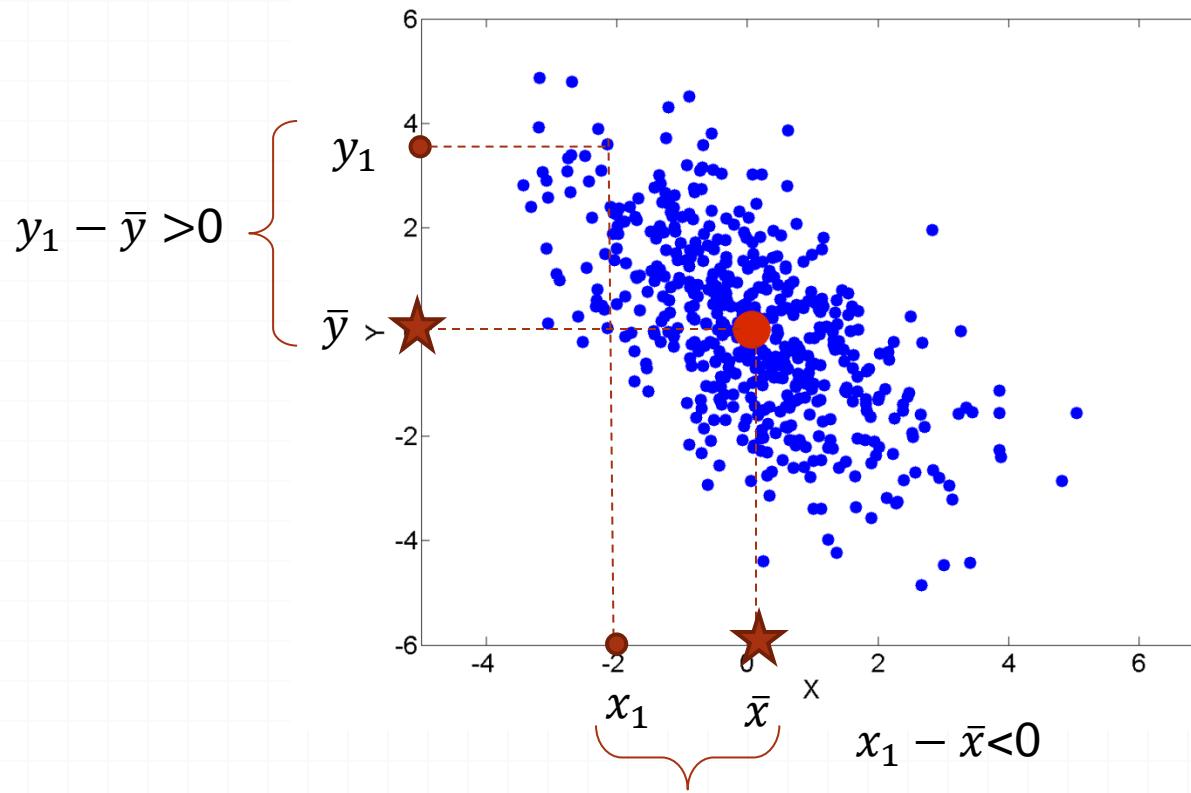
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



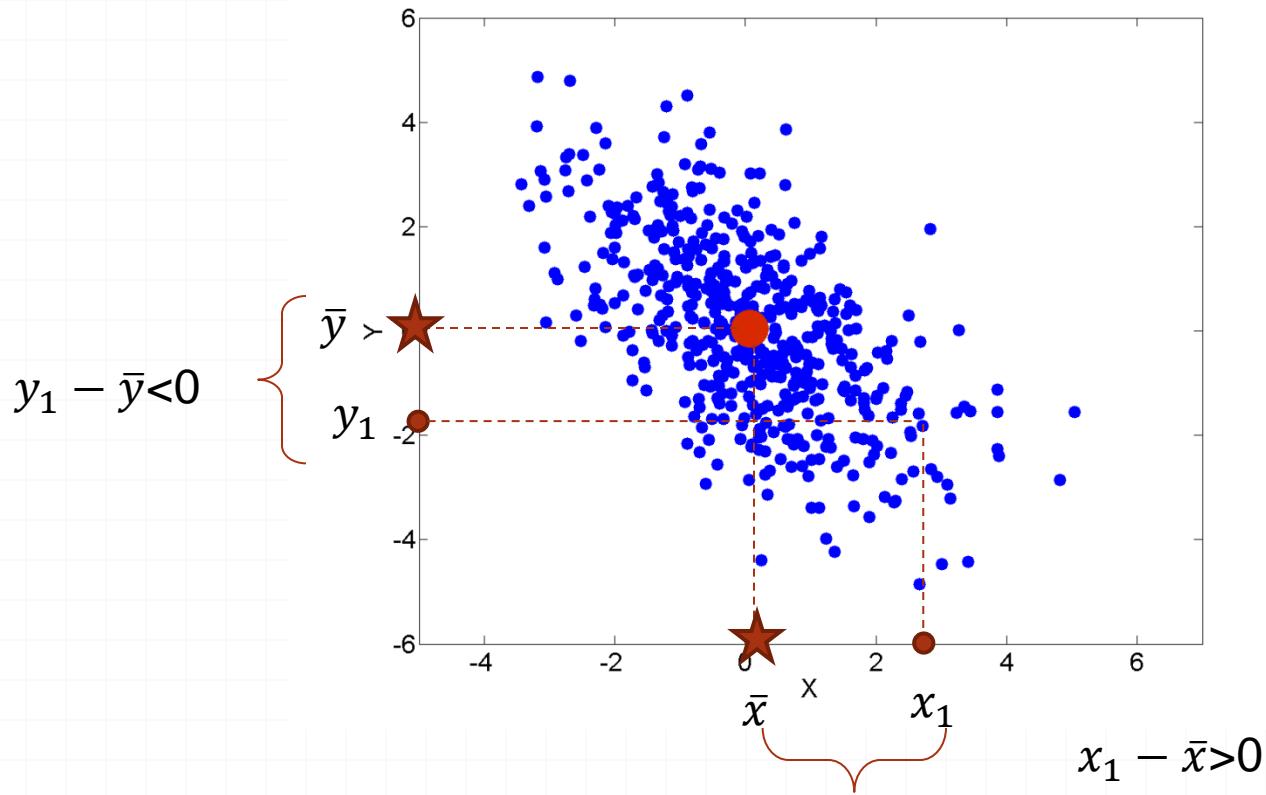
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



Covariance

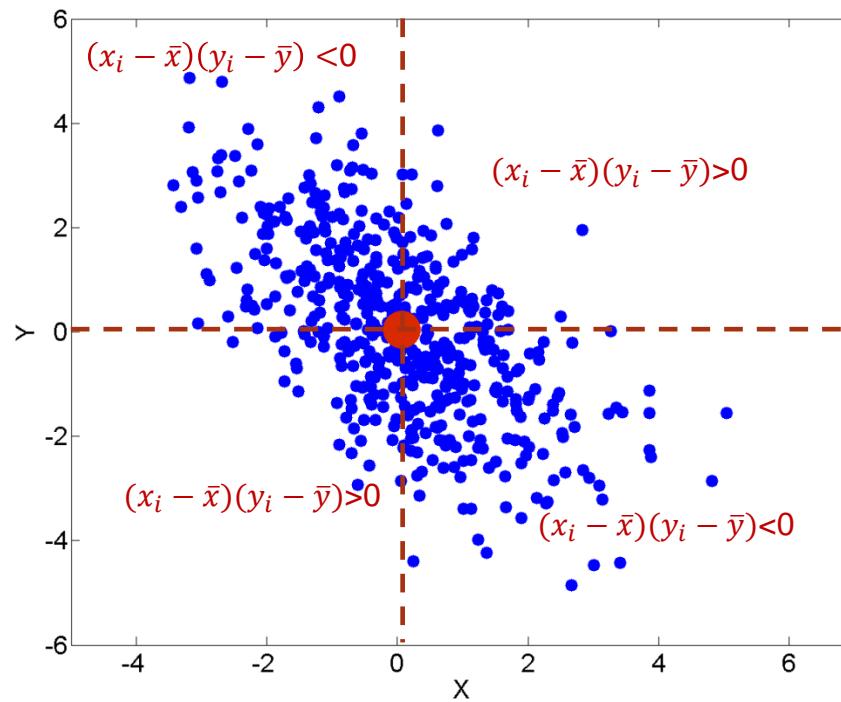
$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



Covariance

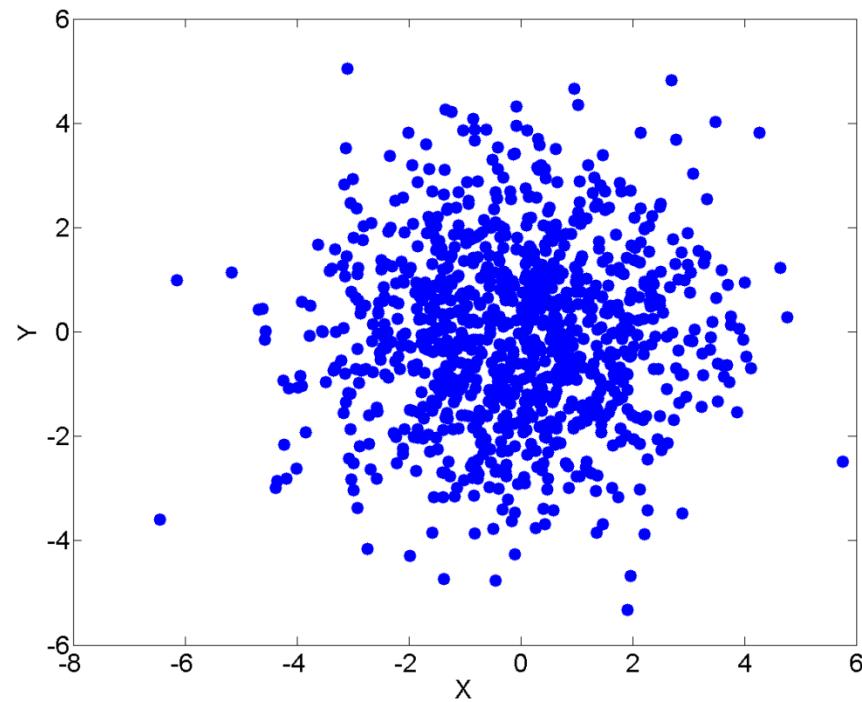
$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

*Negative
Relation*



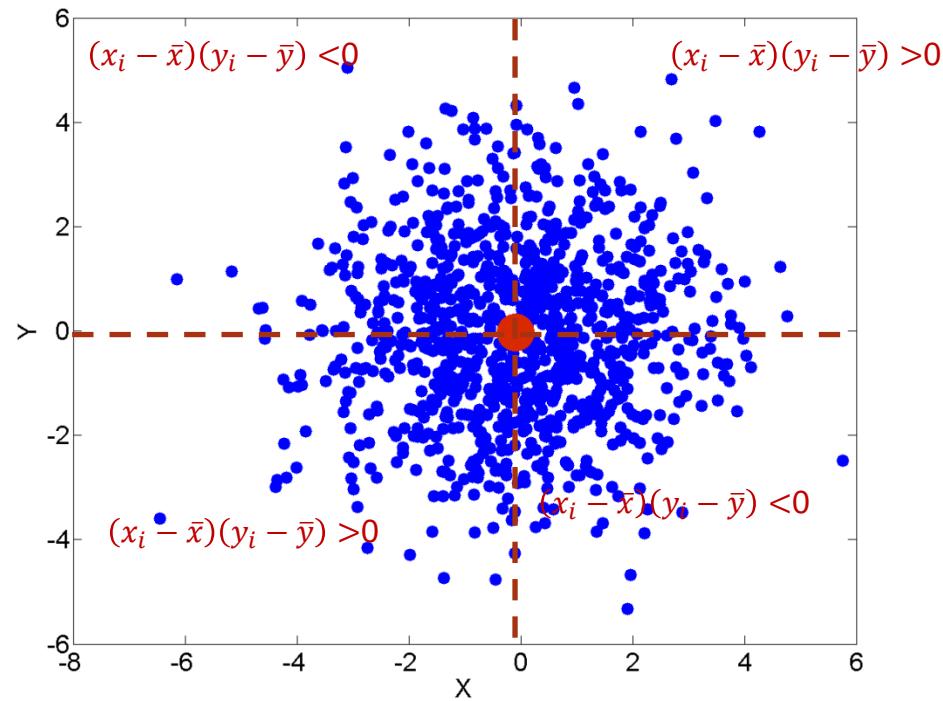
Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



Covariance

$$\text{Covariance}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$



Covariance Matrix

$$Cov(\Sigma) = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_m) \\ cov(x_2, x_1) & cov(x_2, x_2) & \cdots & cov(x_2, x_m) \\ \vdots & \vdots & \vdots & \vdots \\ cov(x_m, x_1) & cov(x_m, x_2) & \cdots & cov(x_m, x_m) \end{bmatrix}$$

$$Cov(\Sigma) = \frac{1}{n} (X - \bar{X})(X - \bar{X})^T; \text{ where } X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Covariance Matrix

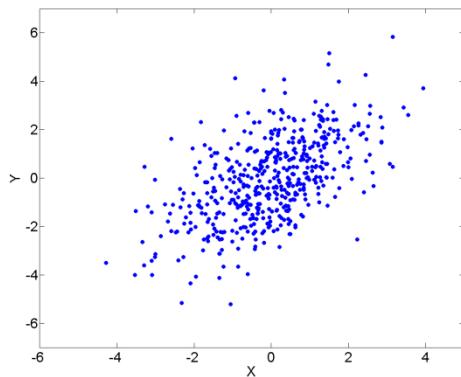
$$Cov(\Sigma) = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_m) \\ cov(x_2, x_1) & cov(x_2, x_2) & \cdots & cov(x_2, x_m) \\ \vdots & \vdots & \vdots & \vdots \\ cov(x_m, x_1) & cov(x_m, x_2) & \cdots & cov(x_m, x_m) \end{bmatrix}$$

- Diagonal elements are variances, i.e. $\text{Cov}(x, x) = \text{var}(x)$.
- Covariance Matrix is symmetric.
- It is a positive semi-definite matrix.

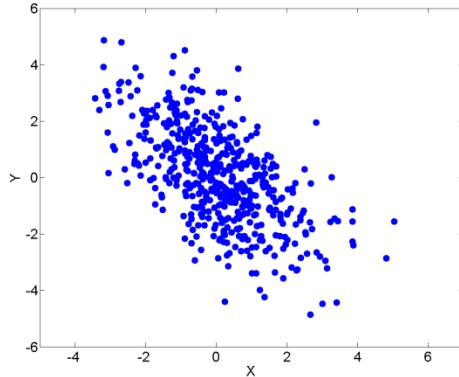
Covariance Matrix

- Covariance is a real symmetric positive semi-definite matrix.
 - ❖ All eigenvalues must be real
 - ❖ Eigenvectors corresponding to different eigenvalues are orthogonal
 - ❖ All eigenvalues are greater than or equal to zero
 - ❖ Covariance matrix can be diagonalized,
i.e. $\text{Cov} = \mathbf{P}\mathbf{D}\mathbf{P}^T$

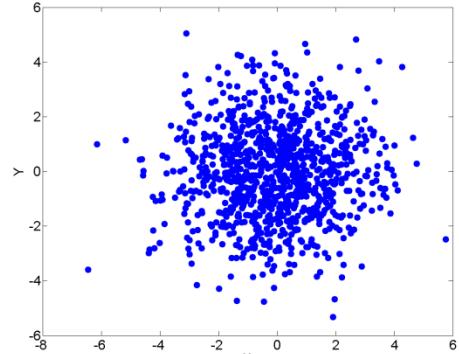
Correlation



Positive relation



Negative relation



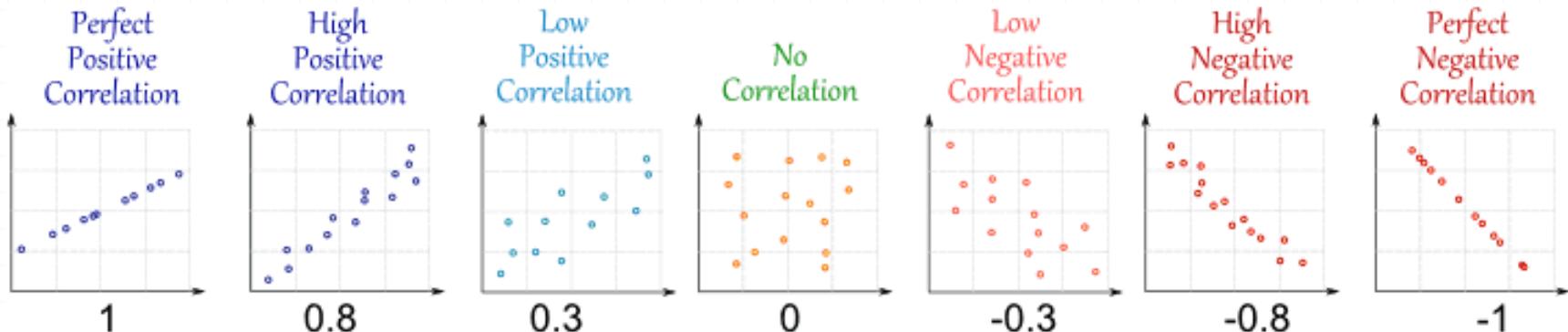
No relation

- Covariance determines whether relation is positive or negative, but it was impossible to measure the **degree** to which the variables are related.
- Correlation is another way to determine how two variables are related.
- In addition to whether variables are positively or negatively related, correlation also tells the **degree** to which the variables are related each other.

Correlation

$$\rho_{xy} = \text{Correlation}(x, y) = \frac{\text{cov}(x, y)}{\sqrt{\text{var}(x)}\sqrt{\text{var}(y)}}.$$

$$-1 \leq \text{Correlation}(x, y) \leq +1$$



Feature Extraction

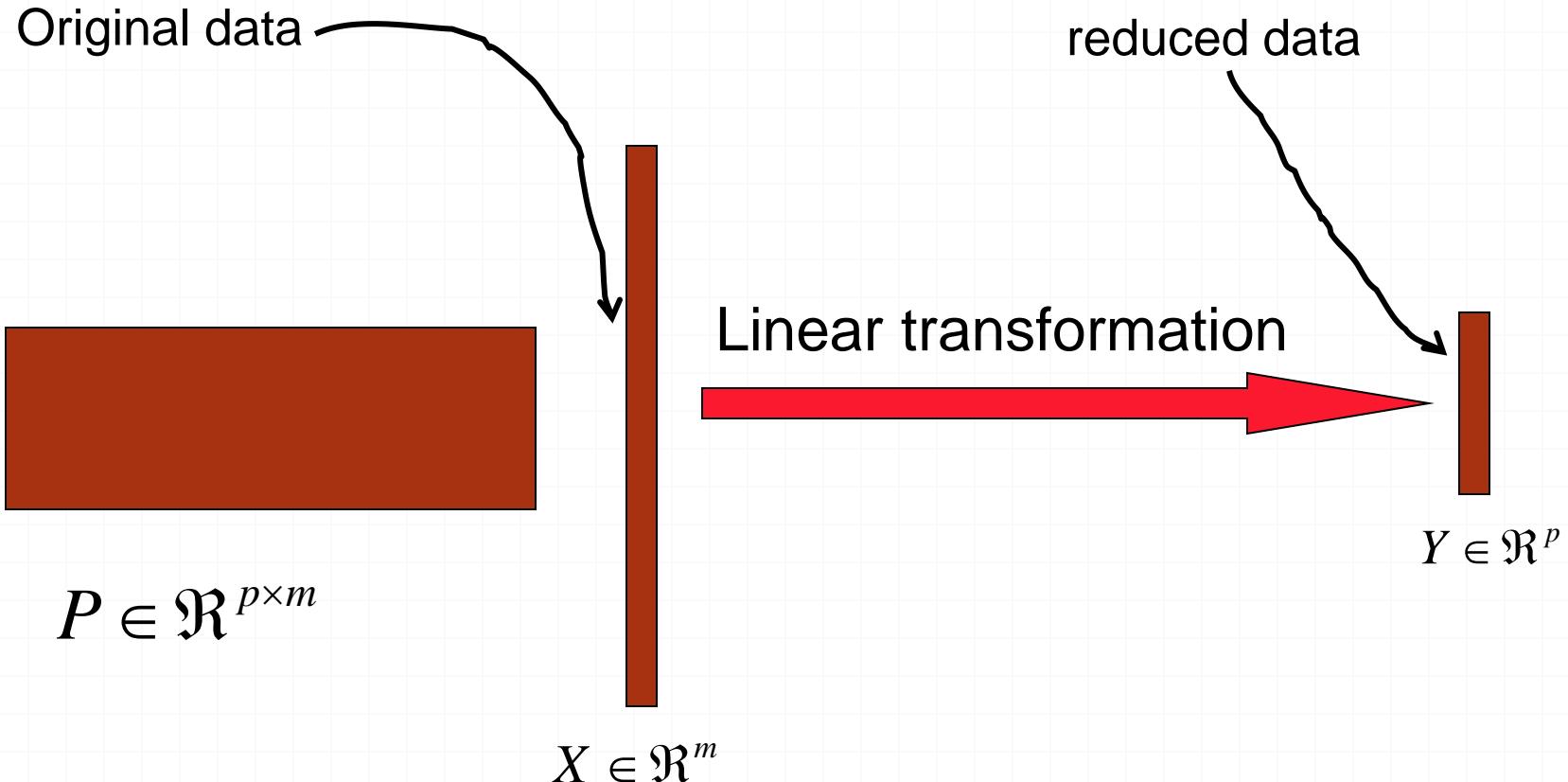
What is feature extraction?

- Feature extraction refers to the mapping of the original high-dimensional data onto a lower-dimensional space.
 - Criterion for feature extraction can be different based on different problem settings.
 - Unsupervised setting: minimize the information loss
 - Supervised setting: maximize the class discrimination
- Given a set of data points of m features $\{x_1, x_2, \dots, x_m\}$

Compute the linear transformation (projection)

$$P \in \mathbb{R}^{p \times m} : x \in \mathbb{R}^m \rightarrow y = Px \in \mathbb{R}^p \quad (p \ll m)$$

What is feature extraction?

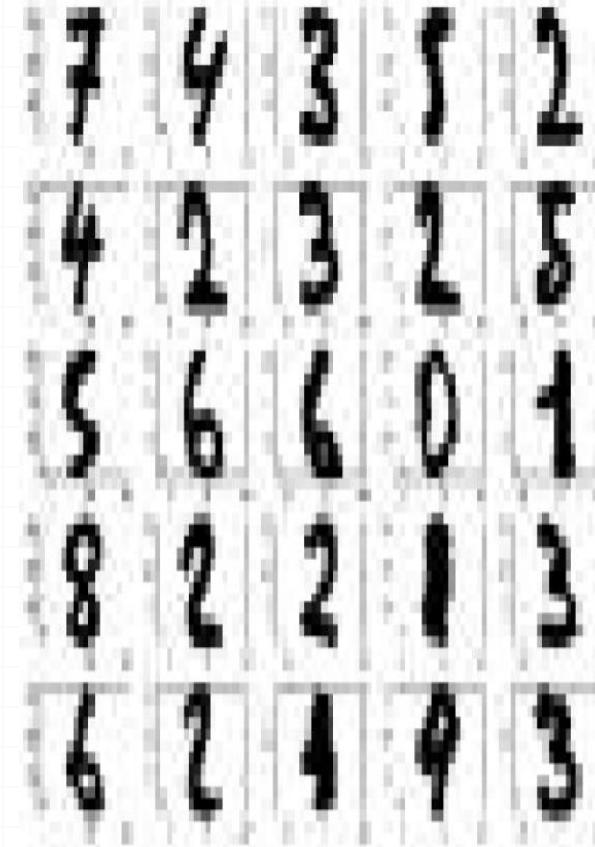


$$P \in \mathcal{R}^{p \times m} : X \rightarrow Y = PX \in \mathcal{R}^p$$

High-dimensional data in computer vision



Face images



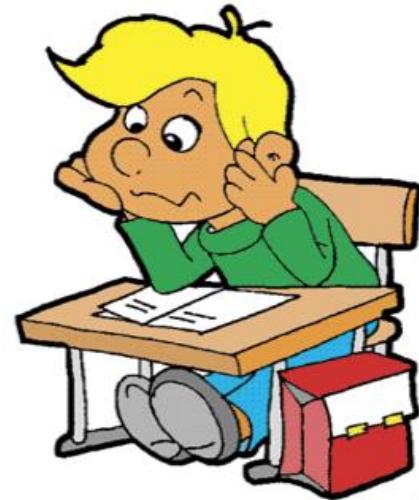
Handwritten digits

Principal Component Analysis

Example of a problem

0 We collected m features about 100 students:

- Height
- Weight
- Hair color
- Average grade
-



0 We want to find the most important features that best describe a student.

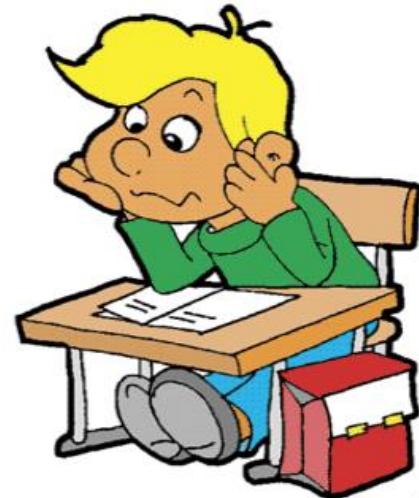
Example of a problem

0 Each student has a vector of data which describes him of length m :

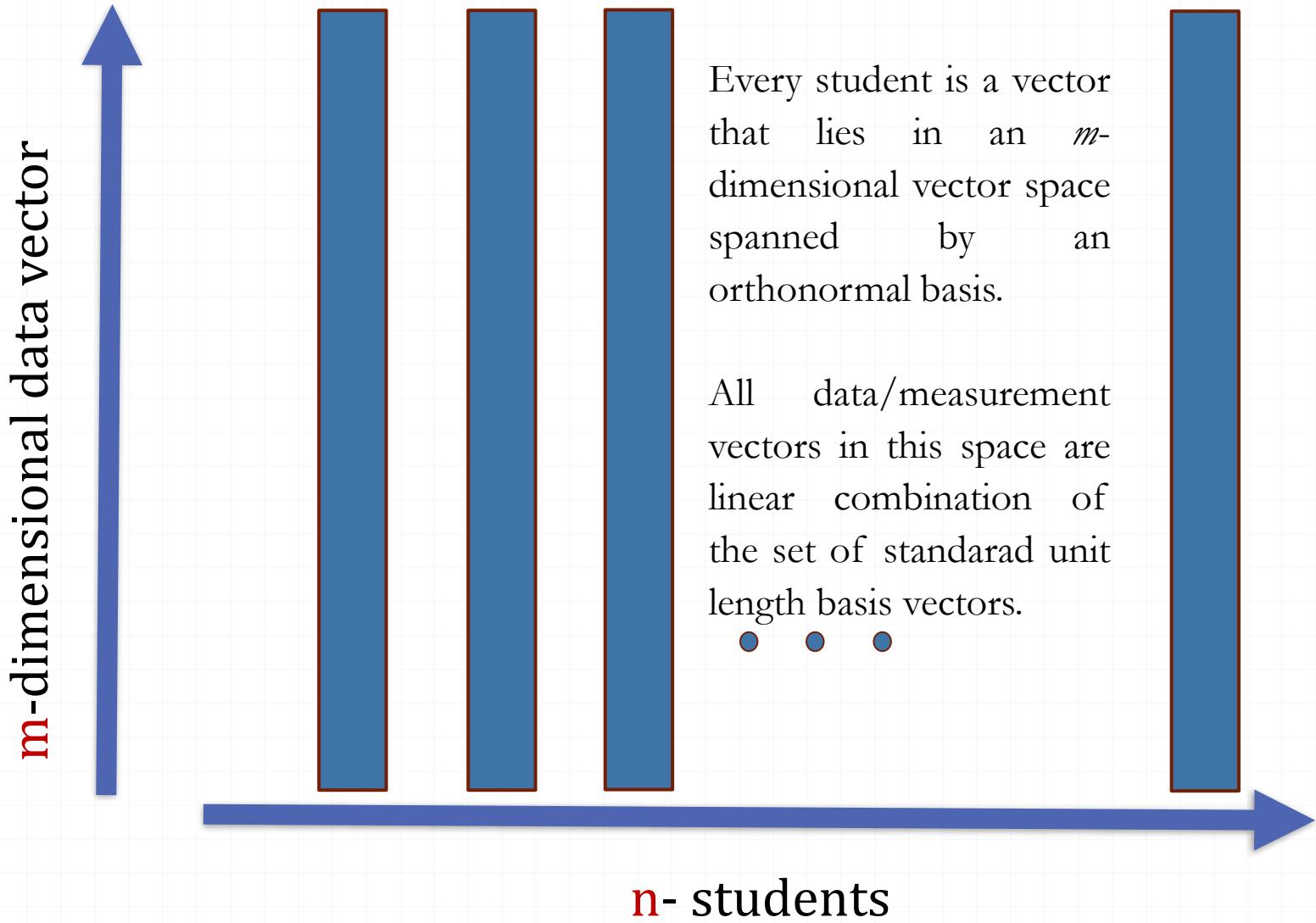
- (180,70,'purple',84,...)

0 We have $n = 100$ such vectors. Let's put them in one matrix, where each column is one student vector.

0 So we have a mxn matrix. This will be the input of our problem.



Example of a problem



What features can we ignore?

- 0 Constant feature (number of heads)
 - 1,1,...,1.
- 0 Constant feature with some noise - (thickness of hair)
 - 0.003, 0.005, 0.002, ..., 0.0008 low variance
- 0 Feature that is linearly dependent on other features (head size and height) : $Z = aX + bY$

Which features do we want to keep?

- 0 Feature that doesn't depend on others (e.g. eye color), i.e. uncorrelated or low covariance.
- 0 Feature that changes a lot (grades)
 - The opposite of noise
 - High variance

Questions

- 0 How we describe 'most important' features using math?
 - High Variance and Low Covariance
- 0 How do we represent our data so that the most important features can be extracted easily?
 - Change of basis

Change of Basis

- Let X and Y be $m \times n$ matrices related by a linear transformation P .
- X is the original recorded data set and Y is a re-representation of that data set.

$$PX = Y$$

- Let's define;
 - p_i 's are the rows of P .
 - x_i 's are the columns of X .
 - y_i 's are the columns of Y .

Change of Basis

- Let X and Y be $m \times n$ matrices related by a linear transformation P .
- X is the original recorded data set and Y is a re-representation of that data set.

$$PX = Y$$

- What does this mean?
 - P is a matrix that transforms X into Y .
 - Geometrically, P is a rotation and a stretch (scaling) which again transforms X into Y .
 - The rows of $P = \{p_1, p_2, \dots, p_m\}$ are a set of new basis vectors for expressing the columns of X .

Change of Basis

o Lets write out the explicit dot products of $\mathbf{P}\mathbf{X}$.

$$\mathbf{P}\mathbf{X} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} [x_1, x_2, \dots, x_n]$$

o We can note the form of each column of \mathbf{Y} .

$$y_i = \begin{bmatrix} p_1 \cdot x_i \\ p_2 \cdot x_i \\ \vdots \\ p_m \cdot x_i \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} p_1 \cdot x_1 & p_1 \cdot x_2 & \cdots & p_1 \cdot x_n \\ p_2 \cdot x_1 & p_2 \cdot x_2 & \cdots & p_2 \cdot x_n \\ \vdots & \vdots & \ddots & \vdots \\ p_m \cdot x_1 & p_m \cdot x_2 & \cdots & p_m \cdot x_n \end{bmatrix}$$

o We can see that each coefficient of y_i is a dot

product of x_i with the corresponding row in \mathbf{P} .

- In other words, the j^{th} coefficient of y_i is a projection onto the j^{th} row of \mathbf{P} .
- Therefore, the rows of \mathbf{P} are a new set of basis vectors for representing the columns of \mathbf{X} .

Change of Basis

- 0 Changing the basis doesn't change the data – only its representation.
- 0 Changing the basis is actually projecting the data vectors on the basis vectors.
- 0 Geometrically, P is a rotation and a stretch of X .
 - If P basis is orthonormal (length = 1) then the transformation P is only a rotation

Questions Remaining !!!

- 0 Assuming linearity, the problem now is to find the appropriate change of basis.
- 0 The rows vectors $\{p_1, p_2, \dots, p_m\}$ in this transformation will become the principal components of \mathbf{X} .
- 0 Now,
 - What is the best way to *re-express X ?*
 - What is the good choice of basis \mathbf{P} ?
- 0 Moreover,
 - *what features we would like Y to exhibit?*

What does “best express” the data means?

- ① As we’ve said before, we want to filter out noise and extract the relevant information from the given data set.
- ② Hence, the representation we are looking for will decrease both *noise* and *redundancy* in the data set at hand.

Noise

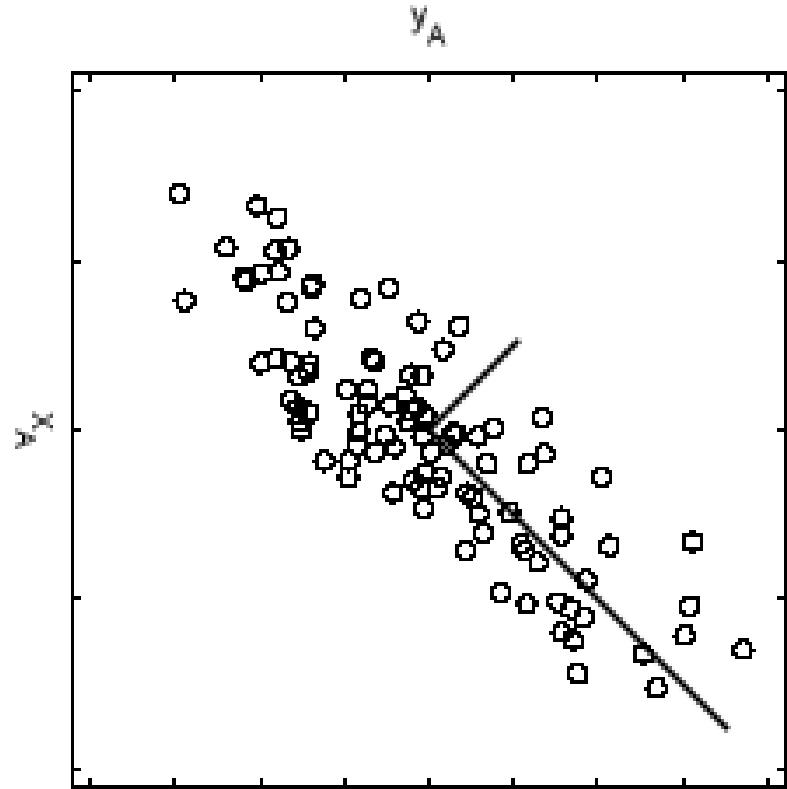
- 0 Noise in any data must be low or – no matter the analysis technique – no information about a system can be extracted.
- 0 There exists no absolute scale for the noise but rather it is measured relative to the measurement, e.g. recorded ball positions.
- 0 A common measure is the *signal-to-noise ratio (SNR)*, or a ratio of variances.

$$SNR = \frac{\sigma_{signal}^2}{\sigma_{noise}^2}$$

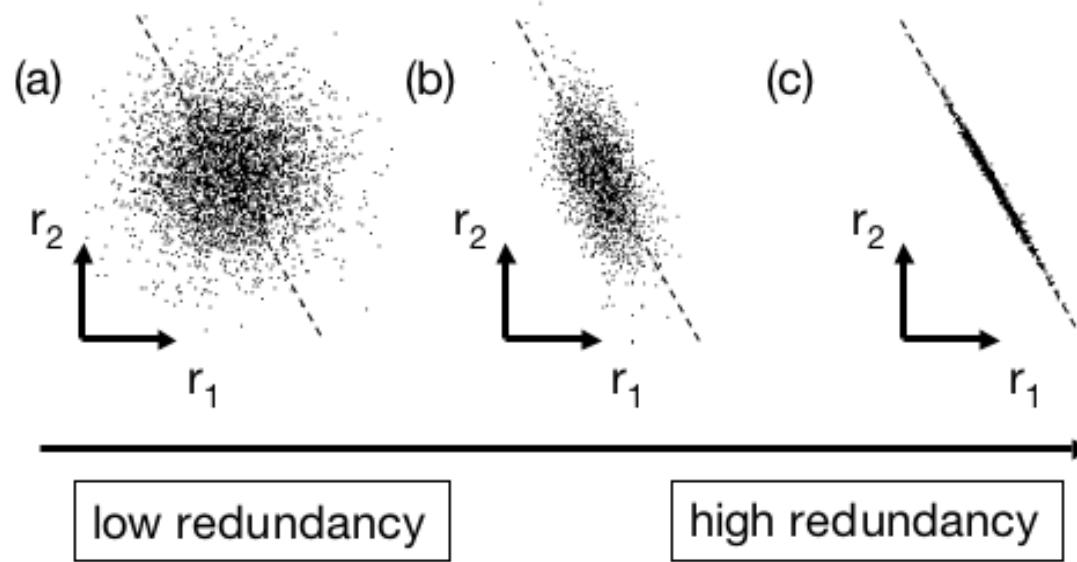
- 0 A high SNR ($>>1$) indicates high precision data, while a low SNR indicates noise contaminated data.

Why Variance ?!!!

○ Find the axis rotation that maximizes SNR = maximizes the variance between axis.

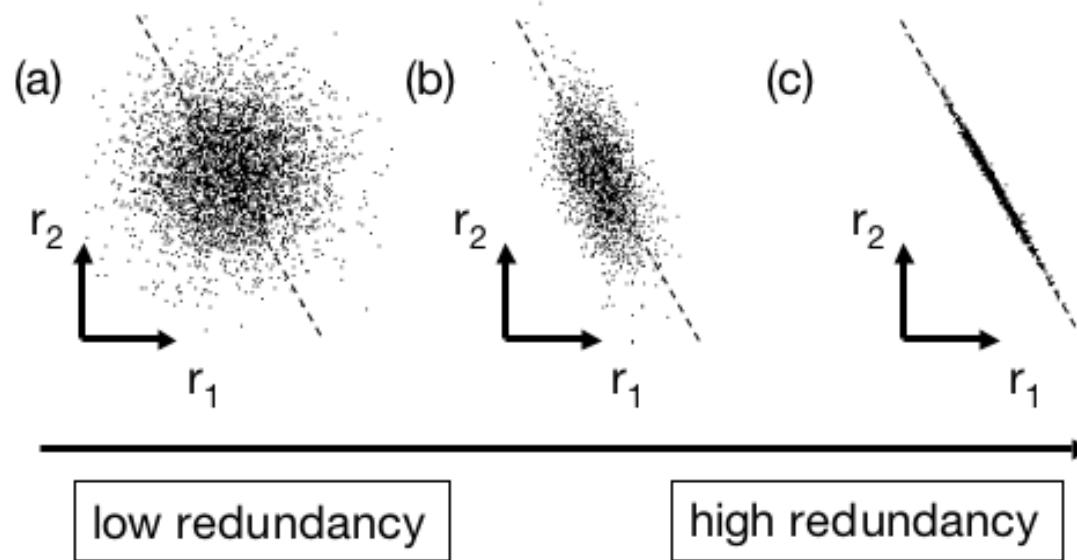


Redundancy ...



- Multiple sensors record the same dynamic information.
- Consider a range of possible plots between two arbitrary measurement r_1 and r_2 .
- Panel(a) depicts two recordings with no redundancy, i.e. they are *un-correlated*, e.g. *person's height and his GPA*.
- However, in panel(c) both recordings appear to be strongly related, i.e. one can be expressed in terms of the other.

Redundancy ...



- Computing **Covariance Matrix (S)** quantifies the correlations between all possible pairs of measurements. Between one pair of measurements, a large covariance/correlation corresponds to a situation like panel (c), while zero covariance corresponds to entirely uncorrelated data as in panel (a)

Redundancy ...

0 Suppose, we have the option of manipulating S_X . We will suggestively define our manipulated covariance matrix S_Y .

What features do you want to optimize in S_Y ?

Diagonalize the Covariance Matrix

Our goals are to find the covariance matrix that:

1. Minimizes redundancy, measured by covariance. (off-diagonal).
 - i.e. we would like each variable to co-vary as little as possible with other variables.
2. Maximizes the signal, measured by variance. (the diagonal)

Diagonalize the Covariance Matrix

PCA Assumptions

- PCA assumes that all basis vectors $\{p_1, p_2, \dots, p_m\}$ are orthonormal (i.e. $p_i \cdot p_j = \delta_{ij}$).
- Hence, in the language of linear algebra, PCA assumes **P is an orthonormal matrix.**
- Secondly, PCA assumes the directions with the largest variances are the most “important” or in other words, most principal.
- *Why are these assumptions easiest?*

Diagonalize the Covariance Matrix

PCA Assumptions

- By the variance assumption PCA first selects a normalized direction in *m-dimensional space along which the variance in \mathbf{X} is maximized* - it saves this as \mathbf{p}_1 .
- Again it finds another direction along which variance is maximized, however, because of the orthonormality condition, it restricts its search to all directions perpendicular to all previous selected directions.
- This could continue until *m directions are selected. The resulting ordered set of \mathbf{p} 's are the principal components.*
- The variances associated with each direction \mathbf{p}_i quantify how principal each direction is. We could thus rank-order each basis vector \mathbf{p}_i according to the corresponding variances.

Solving PCA: Eigen Vectors of Covariance Matrix

- o We will derive an algebraic solution to PCA using linear algebra.
This solution is based on an important property of **eigenvector decomposition.**
- o Once again, the data set is \mathbf{X} , an $m \times n$ matrix, where n is the number of data patterns and m is the number of feature types.
- o The goal is summarized as follows:
 - Find some orthonormal matrix \mathbf{P} where $\mathbf{Y} = \mathbf{P}\mathbf{X}$ such that is $S_Y = \frac{1}{n} \mathbf{Y}\mathbf{Y}^T$ diagonalized. (Assuming zero mean data i.e. subtract the mean)

Solving PCA: Eigen Vectors of Covariance Matrix

We begin by rewriting \mathbf{S}_Y in terms of our variable of choice \mathbf{P} .

$$\begin{aligned} \mathbf{S}_Y &= \frac{1}{n} \mathbf{Y} \mathbf{Y}^T \\ &= \frac{1}{n} (\mathbf{P} \mathbf{X})(\mathbf{P} \mathbf{X})^T \\ &= \frac{1}{n} \mathbf{P} \mathbf{X} \mathbf{X}^T \mathbf{P}^T \\ &= \frac{1}{n} \mathbf{P} (\mathbf{X} \mathbf{X}^T) \mathbf{P}^T \\ &= \mathbf{P} \mathbf{A} \mathbf{P}^T \end{aligned}$$

- Note that we defined a new matrix $\mathbf{A} = (1/n)\mathbf{X}\mathbf{X}^T$, where \mathbf{A} is *symmetric* .
- Our roadmap is to recognize that a symmetric matrix (\mathbf{A}) is diagonalized by an orthogonal matrix of its eigenvectors.
- A symmetric matrix \mathbf{A} can be written as $\mathbf{E} \mathbf{D} \mathbf{E}^T$, where \mathbf{D} is a diagonal matrix and \mathbf{E} is a matrix of eigenvectors of \mathbf{A} arranged as columns.
- The matrix \mathbf{A} has $r \leq m$ orthonormal eigenvectors where r is the rank of the matrix.

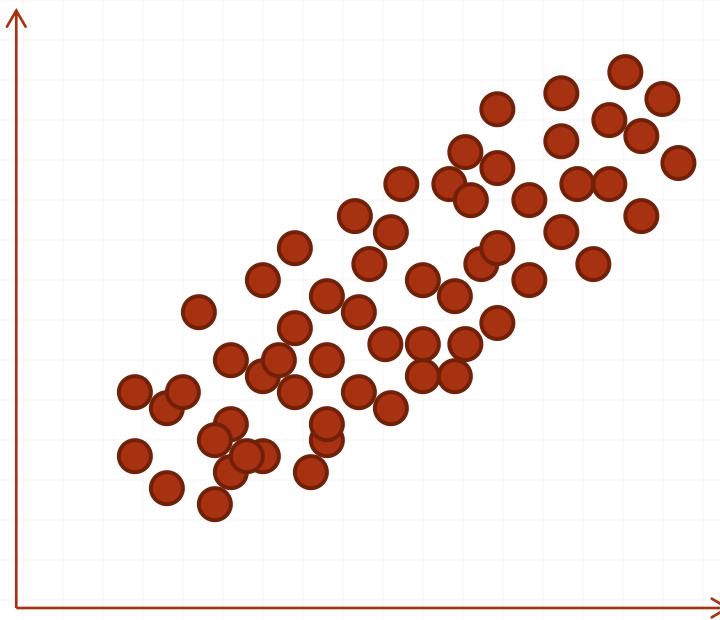
Solving PCA: Eigen Vectors of Covariance Matrix

- Now comes the trick. We select the matrix \mathbf{P} to be a matrix where each row \mathbf{p}_i is an eigenvector of $(1/n)\mathbf{XX}^T$.
- By this selection, $\mathbf{P}=\mathbf{E}^T$. Hence $\mathbf{A}=\mathbf{P}^T\mathbf{D}\mathbf{P}$.
- With this relation and the fact that $\mathbf{P}^{-1}=\mathbf{P}^T$ since the inverse of orthonormal matrix is its transpose, we can finish evaluating \mathbf{S}_Y as follows;

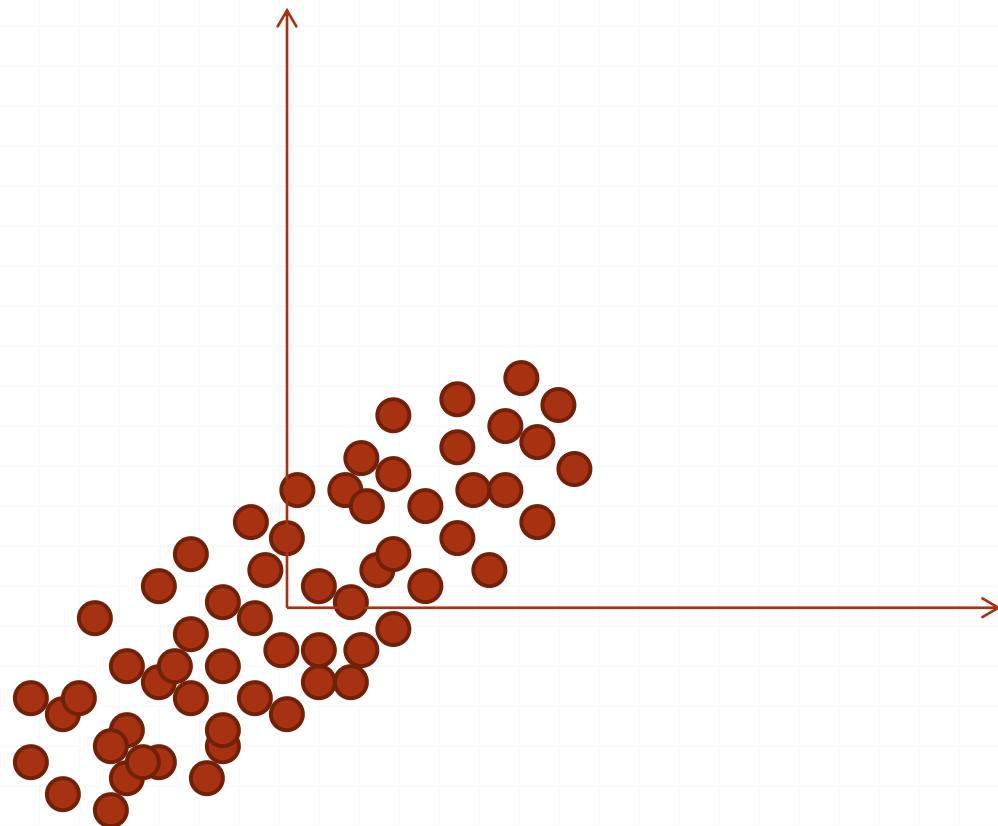
$$\begin{aligned} S_Y &= \mathbf{P}\mathbf{A}\mathbf{P}^T \\ &= \mathbf{P}(\mathbf{P}^T\mathbf{D}\mathbf{P})\mathbf{P}^T \\ &= \mathbf{P}(\mathbf{P}^T\mathbf{D}\mathbf{P})\mathbf{P}^T \\ &= (\mathbf{P}\mathbf{P}^T)\mathbf{D}(\mathbf{P}\mathbf{P}^T) \\ &= (\mathbf{P}\mathbf{P}^{-1})\mathbf{D}(\mathbf{P}\mathbf{P}^{-1}) \\ S_Y &= \mathbf{D} \end{aligned}$$

It is evident that the choice of \mathbf{P} **diagonalizes \mathbf{S}_Y** . This was the goal for PCA.

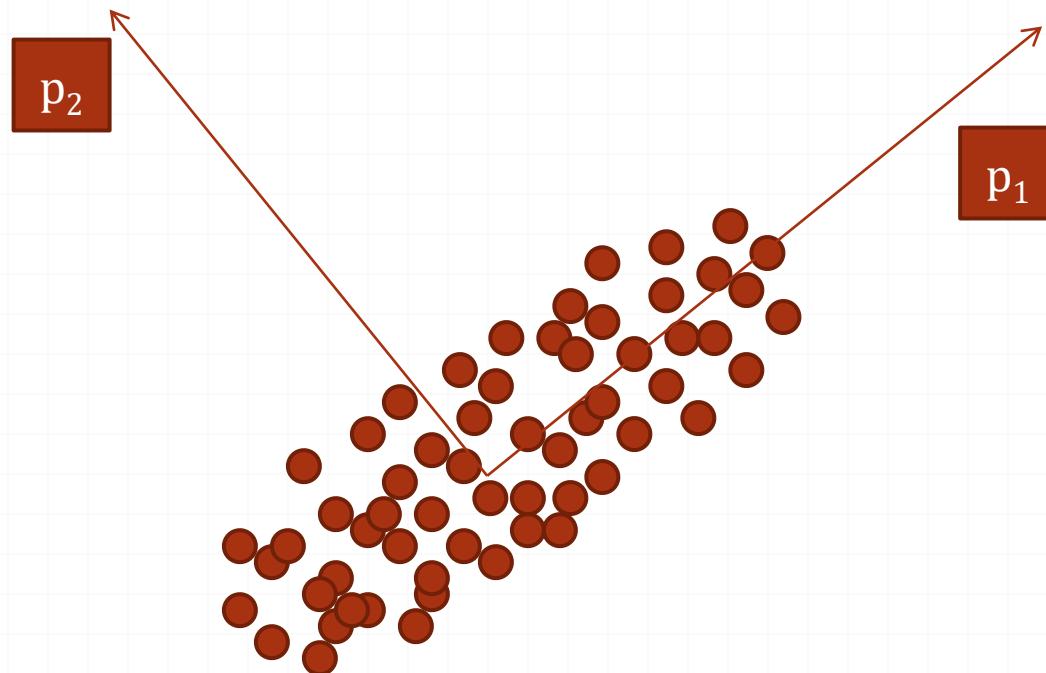
What We Have Done So Far....



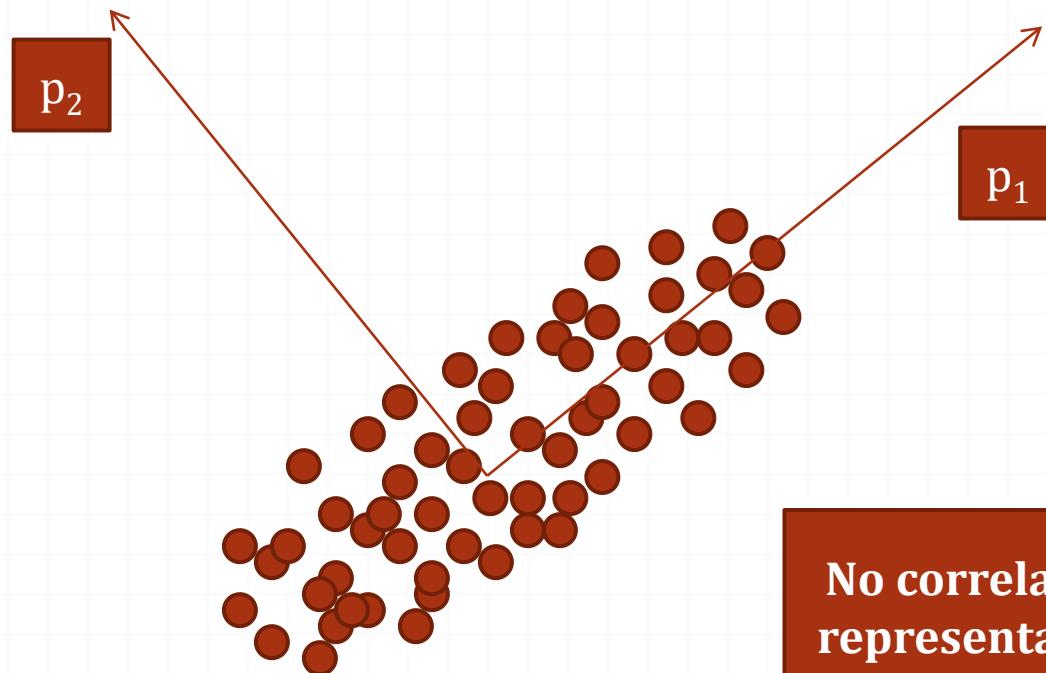
What We Have Done So Far....



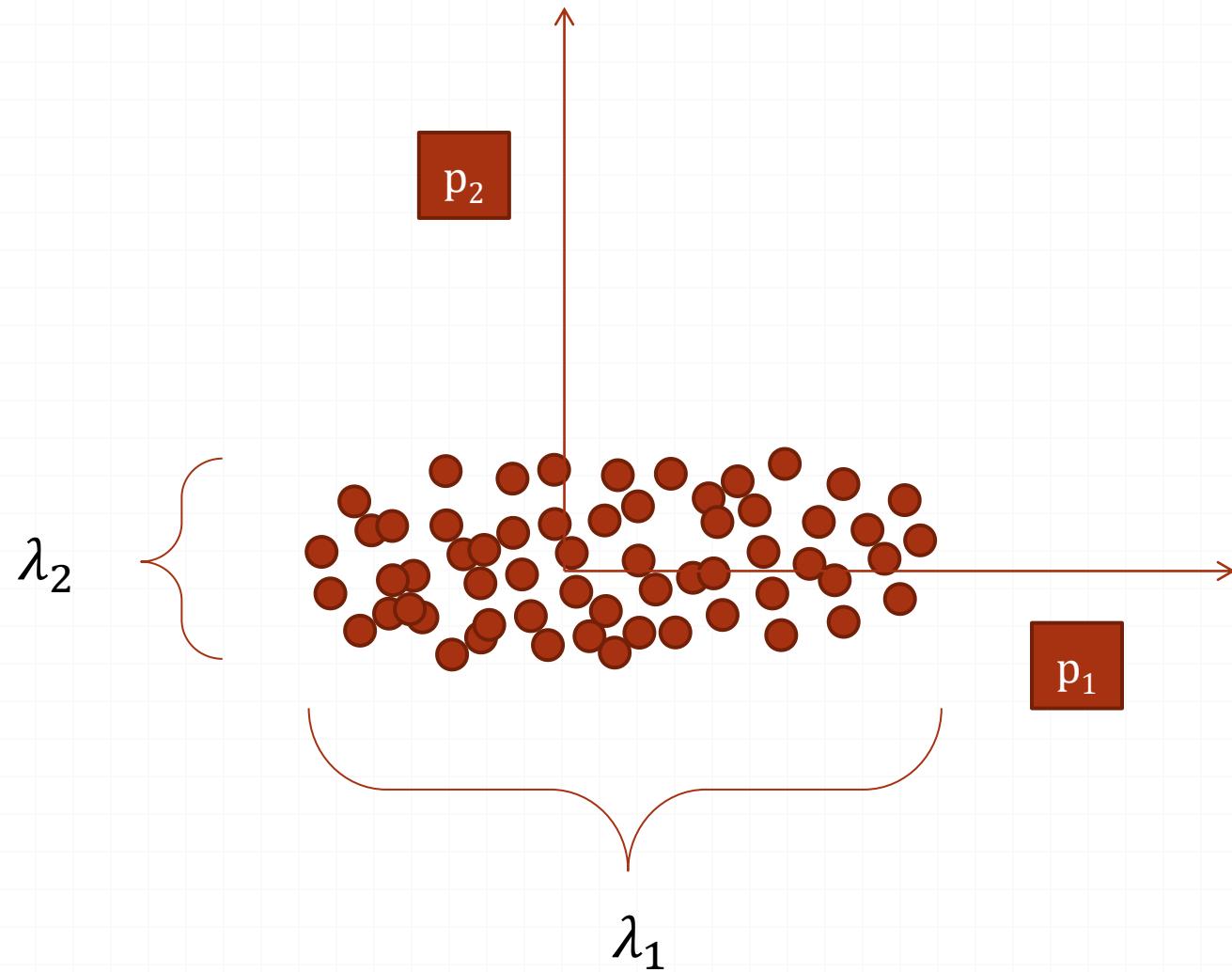
What We Have Done So Far....



What We Have Done So Far....



What We Have Done So Far....



PCA Process – STEP 1

- Subtract the mean from each of the dimensions
- This produces a data set whose mean is zero.
- Subtracting the mean makes variance and covariance calculation easier by simplifying their equations.
- The variance and co-variance values are not affected by the mean value.
- Suppose we have two measurement types \mathbf{X}_1 and \mathbf{X}_2 , hence $m = 2$, and ten samples each, hence $n = 10$.

PCA Process – STEP 1

X_1	X_2		X'_1	X'_2
2.5	2.4		0.69	0.49
0.5	0.7		-1.31	-1.21
2.2	2.9		0.39	0.99
1.9	2.2		0.09	0.29
3.1	3.0	$\Rightarrow \frac{\bar{X}_1 = 1.81}{\bar{X}_2 = 1.91}$	1.29	1.09
2.3	2.7		0.49	0.79
2.0	1.6		0.19	-0.31
1.0	1.1		-0.81	-0.81
1.5	1.6		-0.31	-0.31
1.2	0.9		-0.71	-1.01

PCA Process – STEP 2

- Calculate the covariance matrix

$$\text{cov} = \begin{bmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{bmatrix}$$

- Since the non-diagonal elements in this covariance matrix are positive, we should expect that both the \mathbf{X}_1 and \mathbf{X}_2 variables increase together.
- Since it is symmetric, we expect the eigenvectors to be orthogonal.

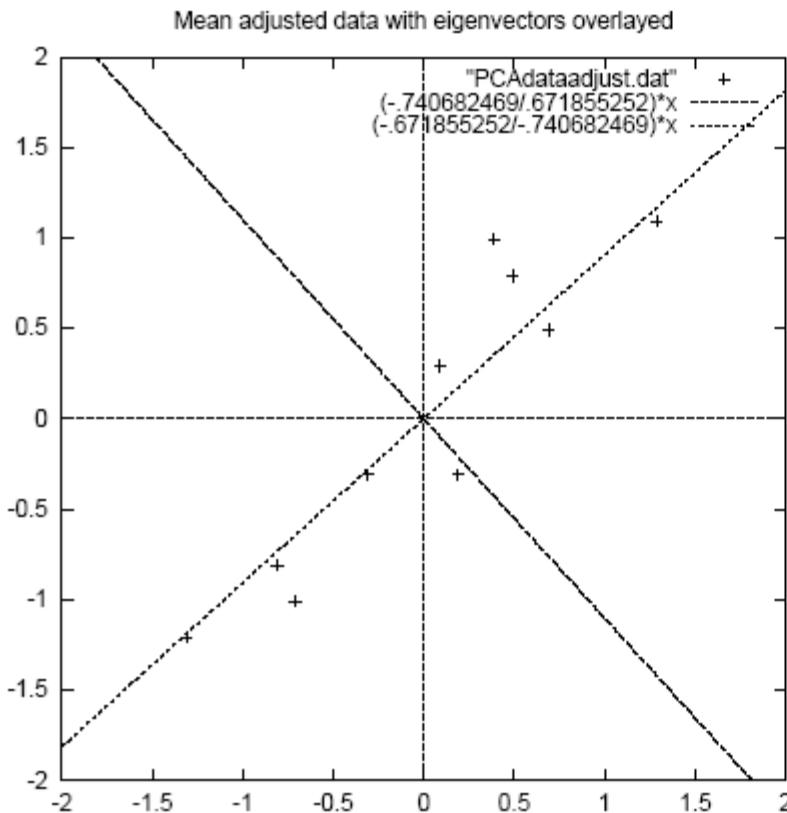
PCA Process – STEP 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix

$$eigenvalues = \begin{bmatrix} 0.490833989 \\ 1.28402771 \end{bmatrix}$$

$$eigenvectors = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

PCA Process – STEP 3



A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

Eigenvectors are plotted as diagonal dotted lines on the plot. (note: they are perpendicular to each other).

One of the eigenvectors goes through the middle of the points, like drawing a line of best fit.

The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

PCA Process – STEP 4

- Reduce dimensionality and form *feature vector*

The eigenvector with the *highest* eigenvalue is the *principal component* of the data set.

In our example, the eigenvector with the largest eigenvalue is the one that points down the middle of the data.

Once eigenvectors are found from the covariance matrix, the next step is to *order them by eigenvalue*, highest to lowest. This gives the components in order of significance.

PCA Process – STEP 4

Now, if you'd like, you can decide to *ignore* the components of lesser significance.

You do *lose some information*, but if the eigenvalues are small, you don't lose much

- m dimensions in your data
- calculate m eigenvectors and eigenvalues
- choose only the first r eigenvectors
- final data set has only r dimensions.

PCA Process – STEP 4

- When the λ_i 's are sorted in descending order, the proportion of variance explained by the r principal components is:

$$\frac{\sum_{i=1}^r \lambda_i}{\sum_{i=1}^m \lambda_i} = \frac{\lambda_1 + \lambda_2 + \dots + \lambda_r}{\lambda_1 + \lambda_2 + \dots + \lambda_p + \dots + \lambda_m}$$

- If the dimensions are highly correlated, there will be a small number of eigenvectors with large eigenvalues and r will be much smaller than m .
- If the dimensions are not correlated, r will be as large as m and PCA does not help.

PCA Process – STEP 4

- Feature Vector

$$\text{FeatureVector} = (\lambda_1 \lambda_2 \lambda_3 \dots \lambda_r)$$

(take the eigenvectors to keep from the ordered list of eigenvectors, and form a matrix with these eigenvectors in the columns)

We can either form a feature vector with both of the eigenvectors:

$$\begin{bmatrix} -0.677873399 & -0.735178656 \\ -0.735178656 & 0.677873399 \end{bmatrix}$$

or, we can choose to leave out the smaller, less significant component and only have a single column:

$$\begin{bmatrix} -0.677873399 \\ -0.735178656 \end{bmatrix}$$

PCA Process – STEP 5

- Derive the new data

FinalData = RowFeatureVector x RowZeroMeanData

RowFeatureVector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top.

RowZeroMeanData is the mean-adjusted data *transposed*, i.e., the data items are in each column, with each row holding a separate dimension.

PCA Process – STEP 5

- FinalData is the final data set, with data items in columns, and dimensions along rows.
- What does this give us?

The original data *solely in terms of the vectors we chose.*

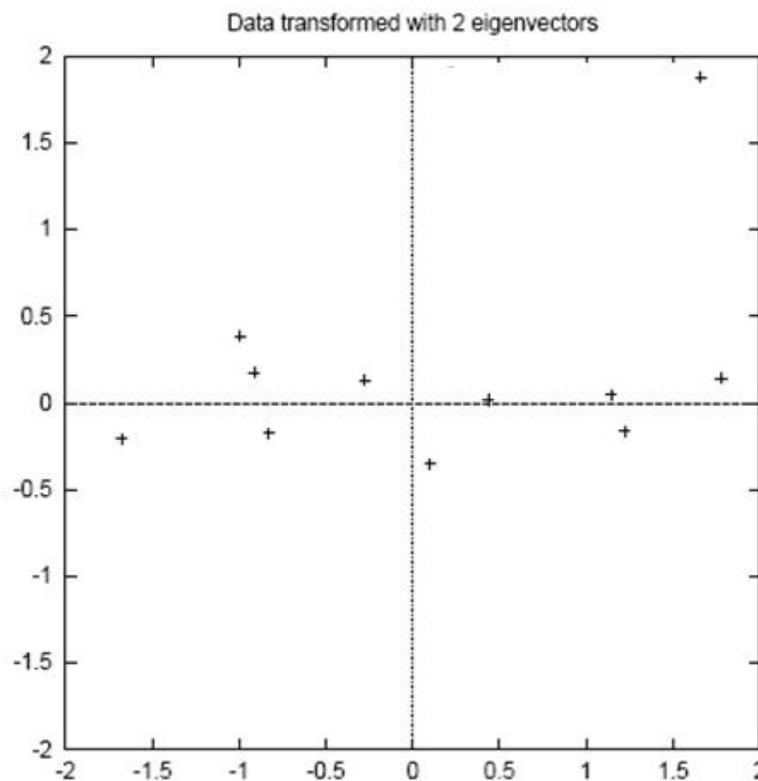
- We have changed our data from being in terms of the axes \mathbf{X}_1 and \mathbf{X}_2 , to now be in terms of our 2 eigenvectors.

PCA Process – STEP 5

FinalData (transpose: dimensions along columns)

$newX_1$	$newX_2$
-0.827870186	-0.175115307
1.77758033	0.142857227
-0.992197494	0.384374989
-0.274210416	0.130417207
-1.67580142	-0.209498461
-0.912949103	0.175282444
0.0991094375	-0.349824698
1.14457216	0.0464172582
0.438046137	0.0177646297
1.22382956	-0.162675287

PCA Process – STEP 5



The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

Reconstruction of Original Data

- Recall that:

$$\text{FinalData} = \text{RowFeatureVector} \times \text{RowZeroMeanData}$$

- Then:

$$\text{RowZeroMeanData} = \text{RowFeatureVector}^{-1} \times \text{FinalData}$$

- And thus:

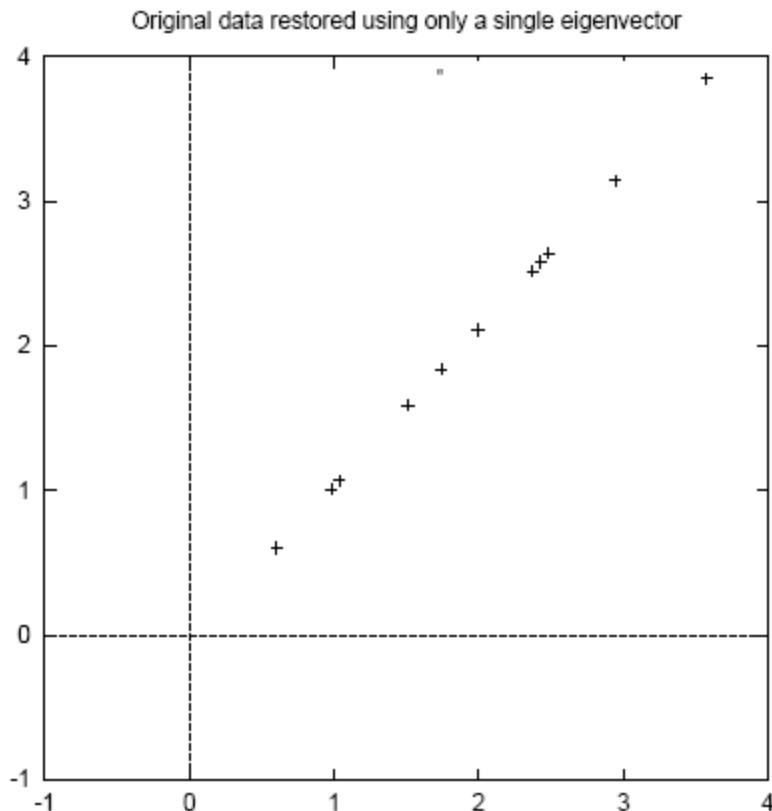
$$\text{RowOriginalData} = (\text{RowFeatureVector}^{-1} \times \text{FinalData}) + \text{OriginalMean}$$

- If we use unit eigenvectors, the inverse is the same as the transpose (hence, easier).

Reconstruction of Original Data

- If we reduce the dimensionality (i.e., $r < m$), obviously, when reconstructing the data we lose those dimensions we chose to discard.
- In our example let us assume that we considered only a single eigenvector.
- The final data is $newX_1$, only and the reconstruction yields...

Reconstruction of Original Data



The variation along
the principal
component is
preserved.

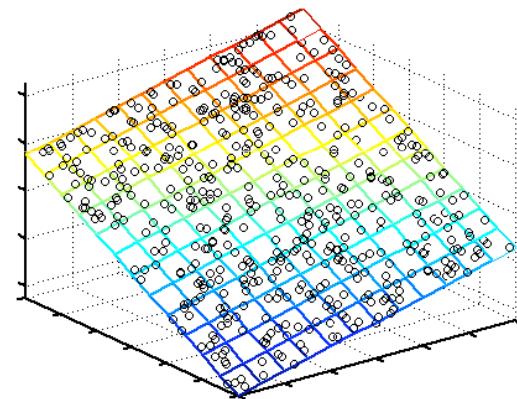
The variation along
the other component
has been lost.

The reconstruction from the data that was derived using only
a single eigen vector

Dimensionality Reduction

- Data representation

Inputs are real-valued vectors in a high dimensional space.



- Linear structure

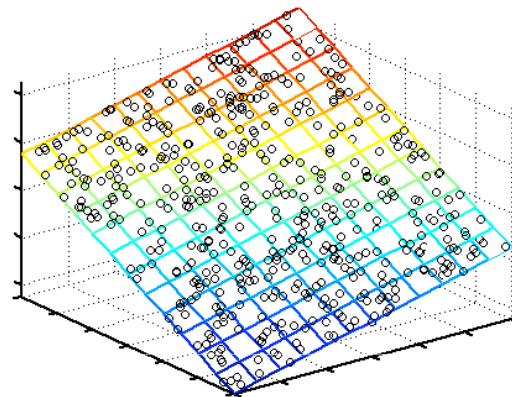
Does the data live in a low dimensional subspace?

- Nonlinear structure

Does the data live on a low dimensional submanifold?



Dimensionality Reduction so far

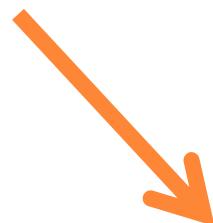


PCA

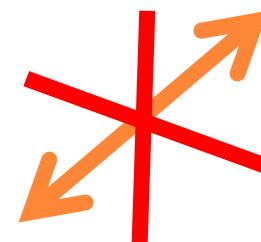


Manifold learning methods

for non linear
structure



Kernel PCA



but does not
unfold the data

Notations

- Inputs (**high dimensional**)

x_1, x_2, \dots, x_n points in R^D

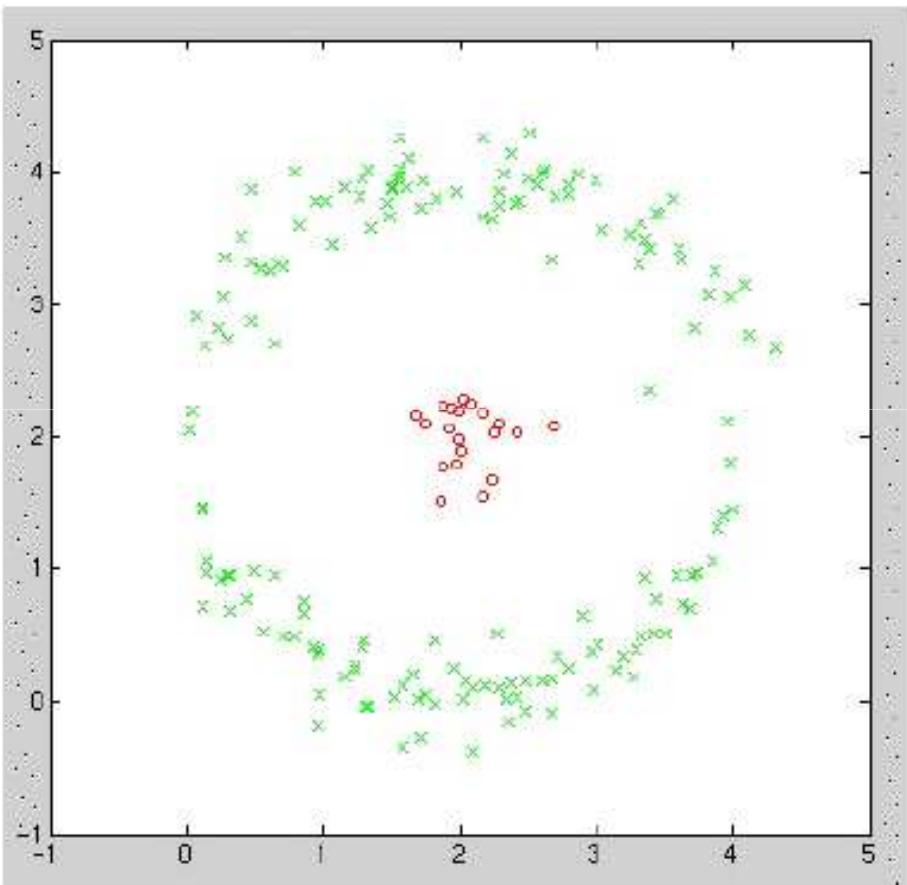
- Outputs (**low dimensional**)

y_1, y_2, \dots, y_n points in R^d ($d \ll D$)

The “magic” of high dimensions

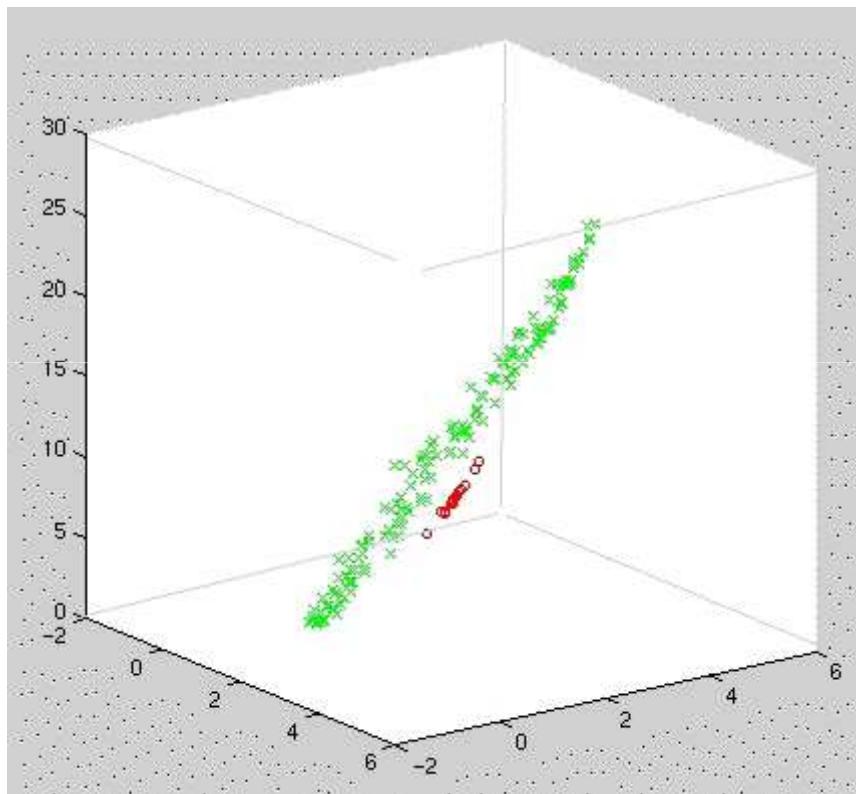
- Given some problem, how do we know what classes of functions are capable of solving that problem?
- VC (Vapnik-Chervonenkis) theory tells us that often mappings which take us into a higher dimensional space than the dimension of the input space provide us with greater classification power.

Example in \mathbb{R}^2



These classes are
linearly inseparable in
the input space.

Example: High-Dimensional Mapping



We can make the problem linearly separable by a simple mapping

$$\Phi : \mathbf{R}^2 \rightarrow \mathbf{R}^3$$

$$(x_1, x_2) \mapsto (x_1, x_2, x_1^2 + x_2^2)$$

Kernel Trick

- High-dimensional mapping can seriously increase computation time.
- Can we get around this problem and still get the benefit of high-D?
- Yes! **Kernel Trick**

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

- Given *any* algorithm that can be expressed solely in terms of dot products, this trick allows us to construct different nonlinear versions of it.

Popular Kernels

Gaussian
$$K(\vec{x}, \vec{x}') = \exp(-\beta \|\vec{x} - \vec{x}'\|^2)$$

Polynomial
$$K(\vec{x}, \vec{x}') = (1 + \vec{x} \cdot \vec{x}')^p$$

Hyperbolic tangent
$$K(\vec{x}, \vec{x}') = \tanh(\vec{x} \cdot \vec{x}' + \delta)$$

Kernel Principal Component Analysis (KPCA)

- Extends conventional principal component analysis (PCA) to a high dimensional feature space using the “kernel trick”.
- Can extract up to n (number of samples) nonlinear principal components without expensive computations.

Making PCA Non-Linear

- Suppose that instead of using the points x_i , we would first map them to some nonlinear **feature space** $\phi(x_i)$
E.g. using polar coordinates instead of cartesian coordinates would help us deal with the circle.
- Extract principal component in that space (PCA)
- The result will be non-linear in the original data space!

Derivation

- Suppose that the mean of the data in the feature space is

$$\mu = \frac{1}{n} \sum_{i=1}^n \phi(x_i) = 0$$

- Covariance:

$$C = \frac{1}{n} \sum_{i=1}^n \phi(x_i) \phi(x_i)^T$$

- Eigenvectors

$$Cv = \lambda v$$

Derivation Cont.

- Eigenvectors can be expressed as linear combination of features:

$$v = \sum_{i=1}^n \alpha_i \phi(x_i)$$

- Proof:

$$Cv = \frac{1}{n} \sum_{i=1}^n \phi(x_i) \phi(x_i)^T v = \lambda v$$

thus

$$v = \frac{1}{\lambda n} \sum_{i=1}^n \phi(x_i) \phi(x_i)^T v = \frac{1}{\lambda n} \sum_{i=1}^n (\phi(x_i) \cdot v) \phi(x_i)^T$$

Showing that $xx^T v = (x \cdot v)x^T$

$$\begin{aligned}(xx^T)v &= \begin{pmatrix} x_1x_1 & x_1x_2 & \dots & x_1x_M \\ x_2x_1 & x_2x_2 & \dots & x_2x_M \\ \vdots & \vdots & \ddots & \vdots \\ x_Mx_1 & x_Mx_2 & \dots & x_Mx_M \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_M \end{pmatrix} \\ &= \begin{pmatrix} x_1x_1v_1 + x_1x_2v_2 + \dots + x_1x_Mv_M \\ x_2x_1v_1 + x_2x_2v_2 + \dots + x_2x_Mv_M \\ \vdots \\ x_Mx_1v_1 + x_Mx_2v_2 + \dots + x_Mx_Mv_M \end{pmatrix}\end{aligned}$$

Showing that $xx^T v = (x \cdot v)x^T$

$$\begin{aligned} &= \begin{pmatrix} (x_1v_1 + x_2v_2 + \dots + x_Mv_M) x_1 \\ (x_1v_1 + x_2v_2 + \dots + x_Mv_M) x_2 \\ \vdots \\ (x_1v_1 + x_2v_2 + \dots + x_Mv_M) x_M \end{pmatrix} \\ &= \left(x_1v_1 + x_2v_2 + \dots + x_Mv_M \right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} \\ &= (x \cdot v)x \quad \square \end{aligned}$$

Derivation Cont.

- So, from before we had,

$$v = \frac{1}{n\lambda} \sum_{i=1}^n \phi(x_i) \phi(x_i)^T v = \frac{1}{n\lambda} \sum_{i=1}^n (\phi(x_i) \cdot v) \phi(x_i)^T$$

just a scalar

- this means that all solutions v with $\lambda = 0$ lie in the span of $\phi(x_1), \dots, \phi(x_n)$, i.e.,

$$v = \sum_{i=1}^n \alpha_i \phi(x_i)$$

- Finding the eigenvectors is equivalent to finding the coefficients α_i

Derivation Cont.

- By substituting this back into the equation we get:

$$\frac{1}{n} \sum_{i=1}^n \phi(x_i) \phi(x_i)^T \left(\sum_{l=1}^n \alpha_{jl} \phi(x_l) \right) = \lambda_j \sum_{l=1}^n \alpha_{jl} \phi(x_l)$$

- We can rewrite it as

$$\frac{1}{n} \sum_{i=1}^n \phi(x_i) \left(\sum_{l=1}^n \alpha_{jl} K(x_i, x_l) \right) = \lambda_j \sum_{l=1}^n \alpha_{jl} \phi(x_l)$$

- Multiply this by $\phi(x_k)$ from the left:

$$\frac{1}{n} \sum_{i=1}^n \phi(x_k)^T \phi(x_i) \left(\sum_{l=1}^n \alpha_{jl} K(x_i, x_l) \right) = \lambda_j \sum_{l=1}^n \alpha_{jl} \phi(x_k)^T \phi(x_l)$$

Derivation Cont.

- By plugging in the kernel and rearranging we get:

$$K^2 \alpha_j = n \lambda_j K \alpha_j$$

We can remove a factor of K from both sides of the matrix (this will only affect the eigenvectors with zero eigenvalue, which will not be a principle component anyway):

$$K \alpha_j = n \lambda_j \alpha_j$$

- We have a normalization condition for the α_j vectors:

$$\nu_j^T \nu_j = 1 \Rightarrow \sum_{k=1}^n \sum_{l=1}^n \alpha_{jl} \alpha_{jk} \phi(x_l)^T \phi(x_k) = 1 \Rightarrow \alpha_j^T K \alpha_j = 1$$

Derivation Cont.

- By multiplying $K\alpha_j = n\lambda_j\alpha_j$ by α_j and using the normalization condition we get:

$$\lambda_j n \alpha_j^T \alpha_j = 1, \quad \forall j$$

- For a new point x , its projection onto the principal components is:

$$\phi(x)^T v_j = \sum_{i=1}^n \alpha_{ji} \phi(x)^T \phi(x_i) = \sum_{i=1}^n \alpha_{ji} K(x, x_i)$$

Normalizing the feature space

- In general, $\phi(x_i)$ may not be zero mean.
- Centered features:

$$\tilde{\phi}(x_k) = \phi(x_i) - \frac{1}{n} \sum_{k=1}^n \phi(x_k)$$

- The corresponding kernel is:

$$\begin{aligned}\tilde{K}(x_i, x_j) &= \tilde{\phi}(x_i)^T \tilde{\phi}(x_j) \\ &= \left(\phi(x_i) - \frac{1}{n} \sum_{k=1}^n \phi(x_k) \right)^T \left(\phi(x_j) - \frac{1}{n} \sum_{k=1}^n \phi(x_k) \right) \\ &= K(x_i, x_j) - \frac{1}{n} \sum_{k=1}^n K(x_i, x_k) - \frac{1}{n} \sum_{k=1}^n K(x_j, x_k) + \frac{1}{n^2} \sum_{l,k=1}^n K(x_l, x_k)\end{aligned}$$

Normalizing the feature space (cont)

$$\tilde{K}(x_i, x_j) = K(x_i, x_j) - \frac{1}{n} \sum_{k=1}^n K(x_i, x_k) - \frac{1}{n} \sum_{k=1}^n K(x_j, x_k) + \frac{1}{n^2} \sum_{l,k=1}^n K(x_l, x_k)$$

- In a matrix form

$$\tilde{\mathbf{K}} = \mathbf{K} - 2\mathbf{1}_{1/n}\mathbf{K} + \mathbf{1}_{1/n}\mathbf{K}\mathbf{1}_{1/n}$$

- where $\mathbf{1}_{1/n}$ is a matrix with all elements $1/n$.

Summary of kernel PCA

- Pick a kernel
- Construct the normalized kernel matrix \tilde{K} of the data (dimension $m \times m$):

$$\tilde{K} = K - \frac{2}{n} \mathbf{1}_{1/n} K + \frac{1}{n} \mathbf{1}_{1/n} K \mathbf{1}_{1/n}$$

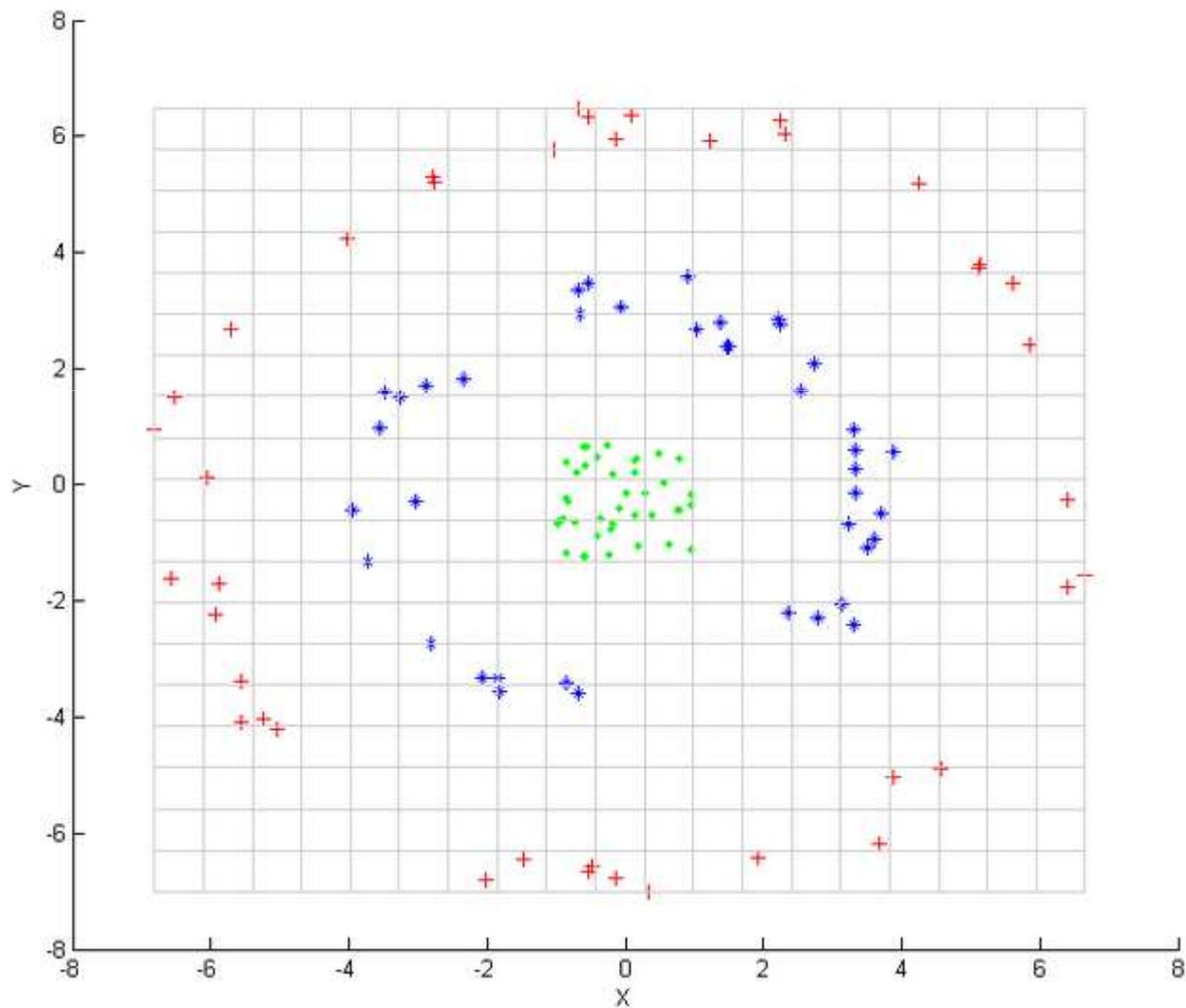
- Solve an eigenvalue problem:

$$\tilde{K} \alpha_i = \lambda_i \alpha_i$$

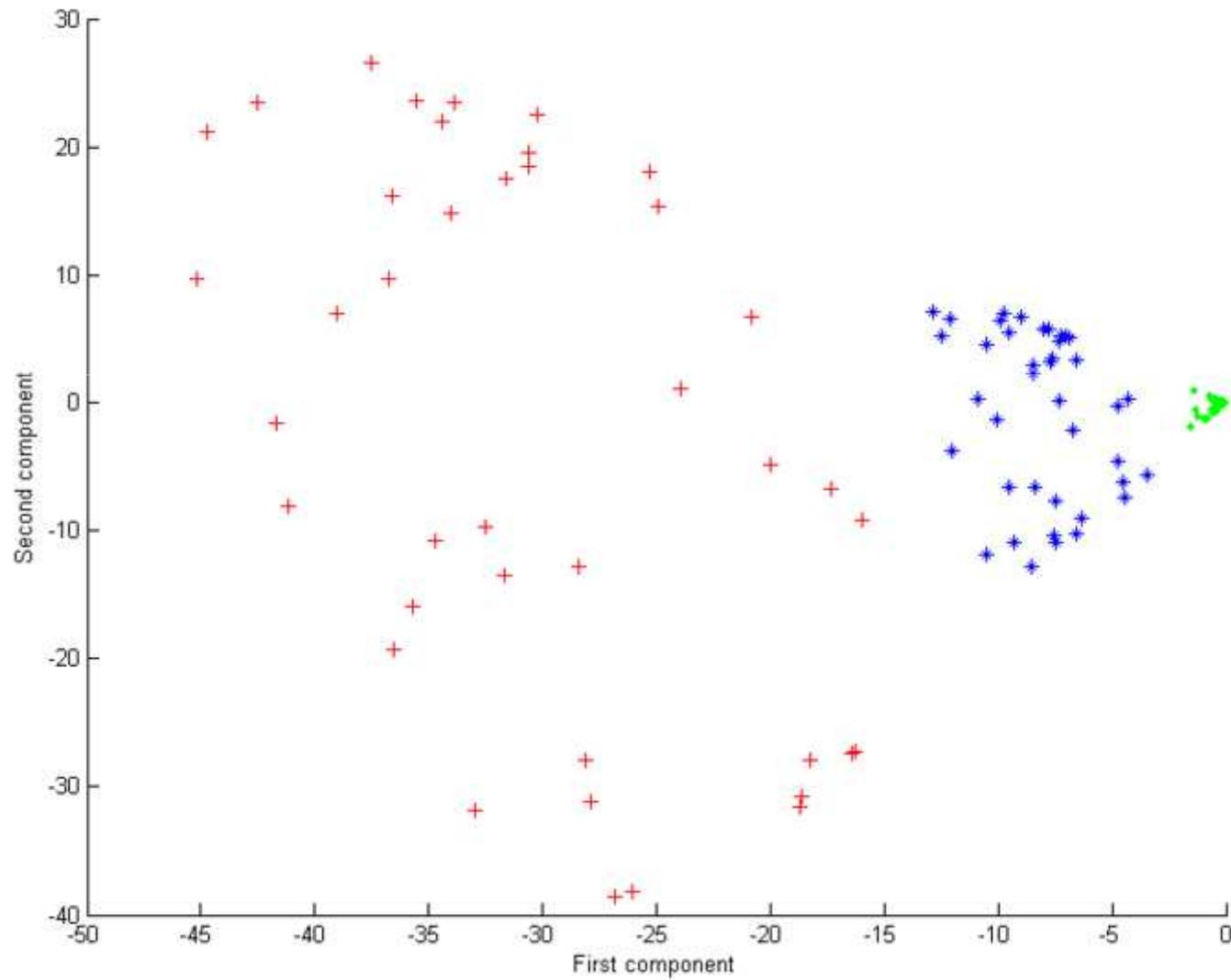
- For any data point (new or old), we can represent it as

$$y_j = \sum_{i=1}^n \alpha_{ji} K(x, x_i), \quad j = 1, \dots, d$$

Example: Input Points



Example: KPCA



Example: De-noising images

Original data



Data corrupted with Gaussian noise



Result after linear PCA



Result after kernel PCA, Gaussian kernel



Properties of KPCA

- Kernel PCA can give a good re-encoding of the data when it lies along a non-linear manifold.
- The kernel matrix is $n \times n$, so kernel PCA will have difficulties if we have lots of data points.