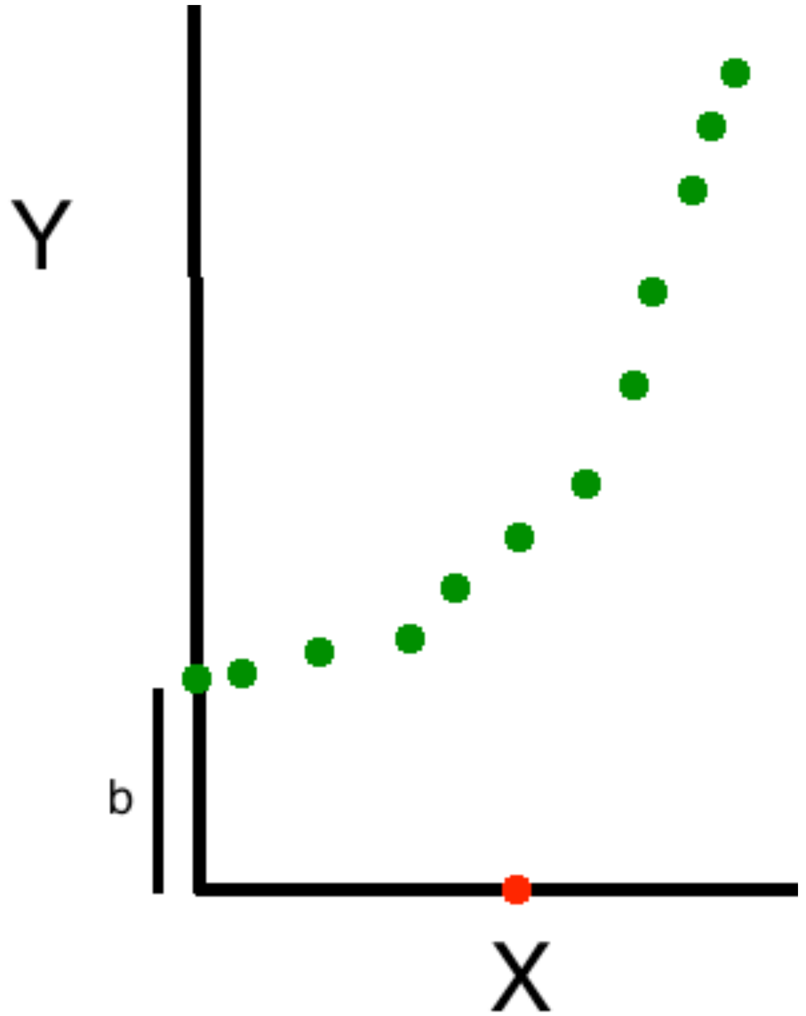
CS4801 : Regression

Sahely Bhadra

1. Regression : How linear regression is useful?
2. Multivariate Linear Regression
3. Overfitting and regularisation
4. Ridge Regression and Lasso
5. Cross validation (model selection)
6. Gradient descent
7. Probabilistic Interpretation

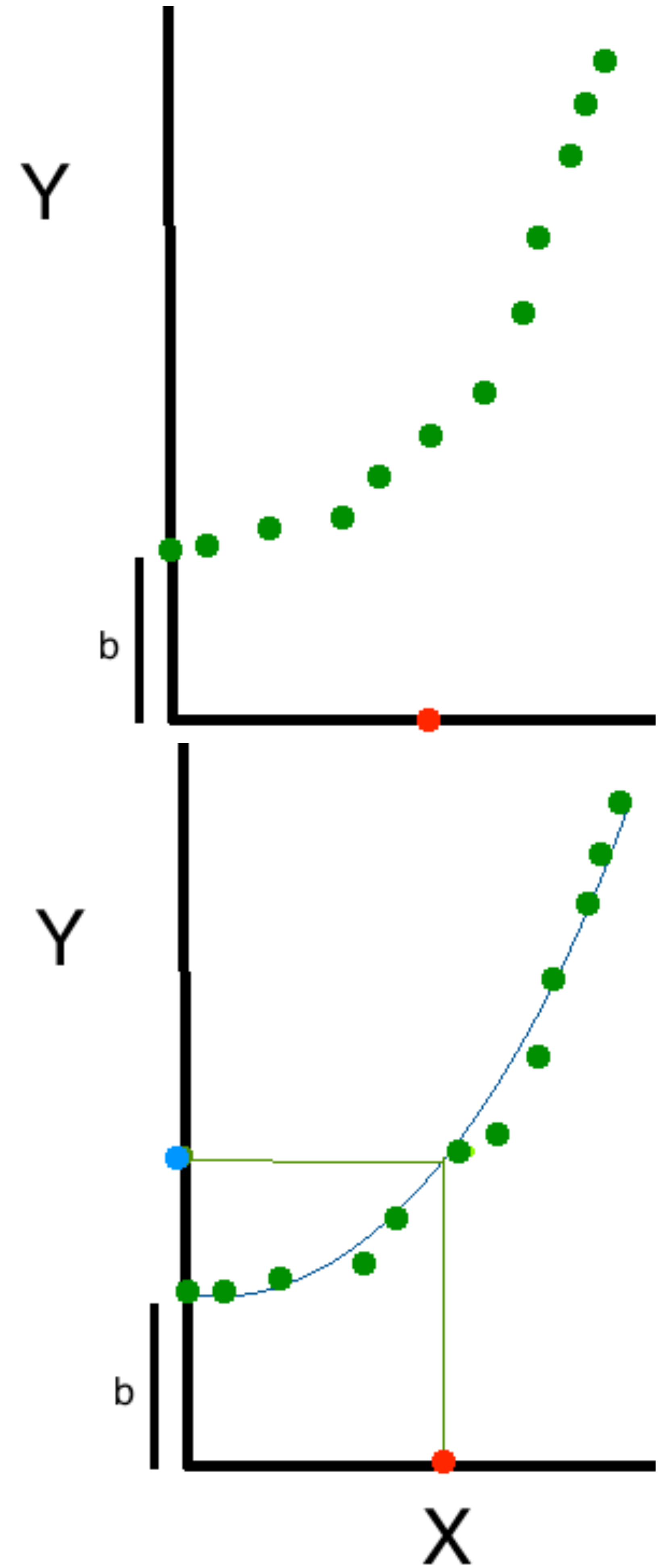
Regression

- Quantitative prediction model
    - Problem
        - Given known data points (input(x) and output (y))
        - Find output for a unknown input



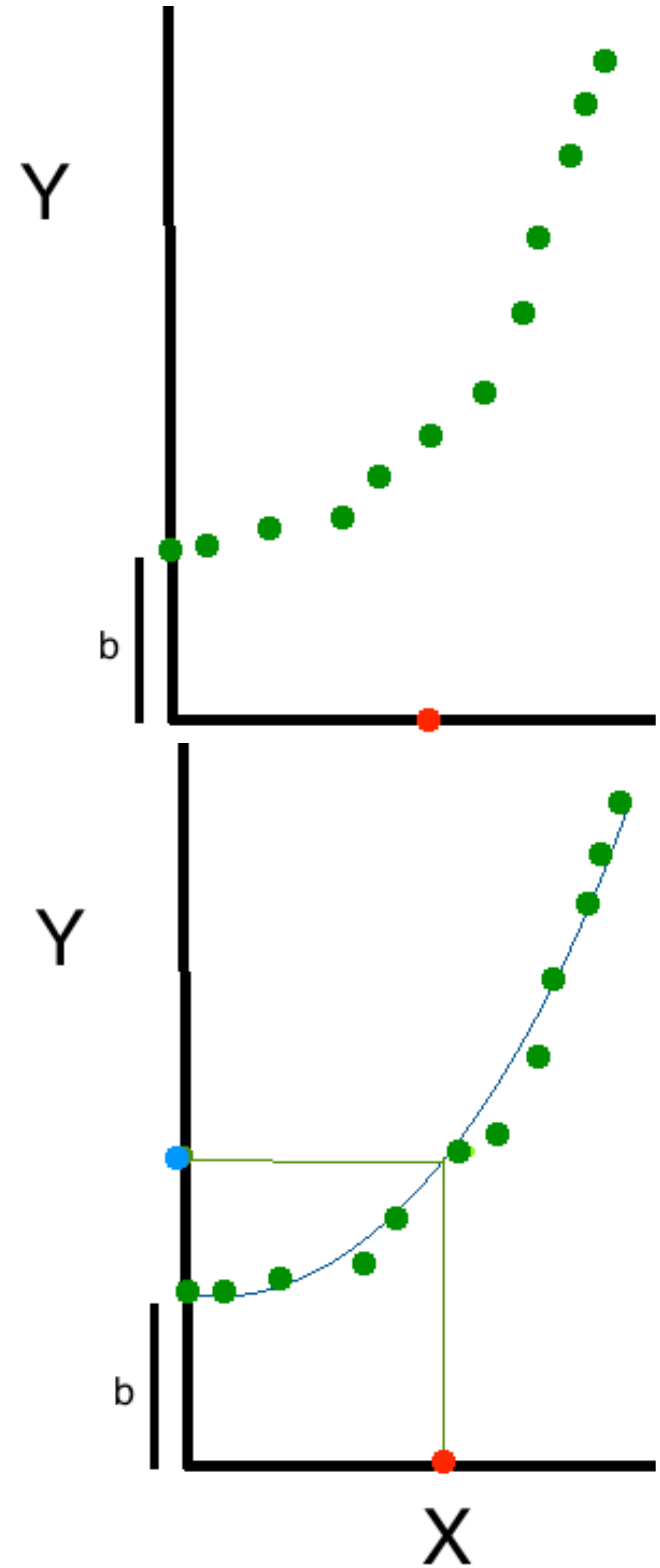
Regression

- Quantitative prediction model
  - Problem
    - Given known data points (input(x) and output (y))
    - Find output for a unknown input
  - Solution
    - Learn function f(x) such that y=f(x) for all inputs
    - predict y for unknown data points using function f(x)



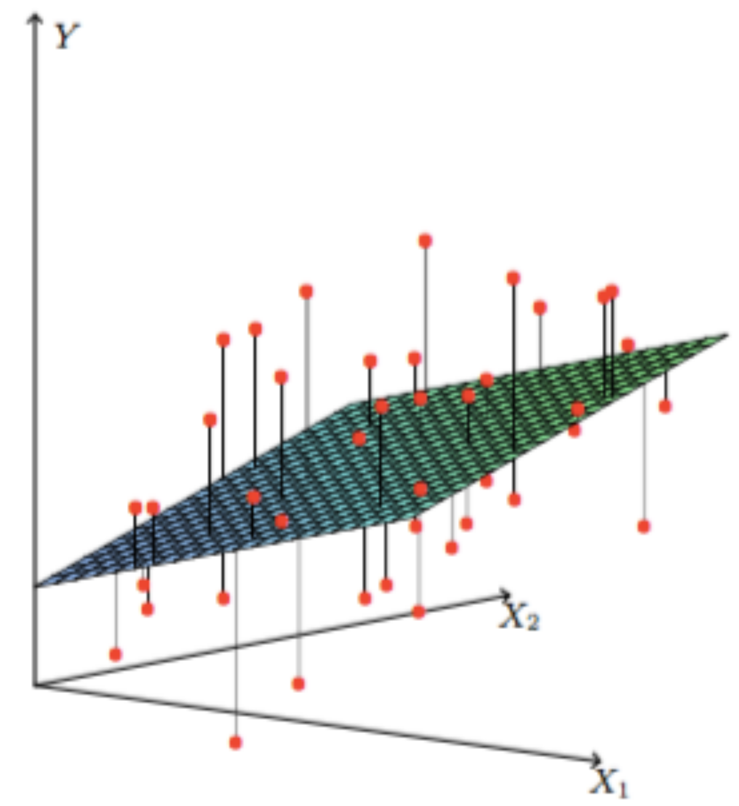
Regression

- Quantitative prediction model
  - Problem
    - Given known data points (input(x) and output (y))
    - Find output for a unknown input
  - Solution
    - Learn function f(x) such that y=f(x) for all inputs
    - predict y for unknown data points using function f(x)
    - **$Y=X^2+b$ (intuitively)**



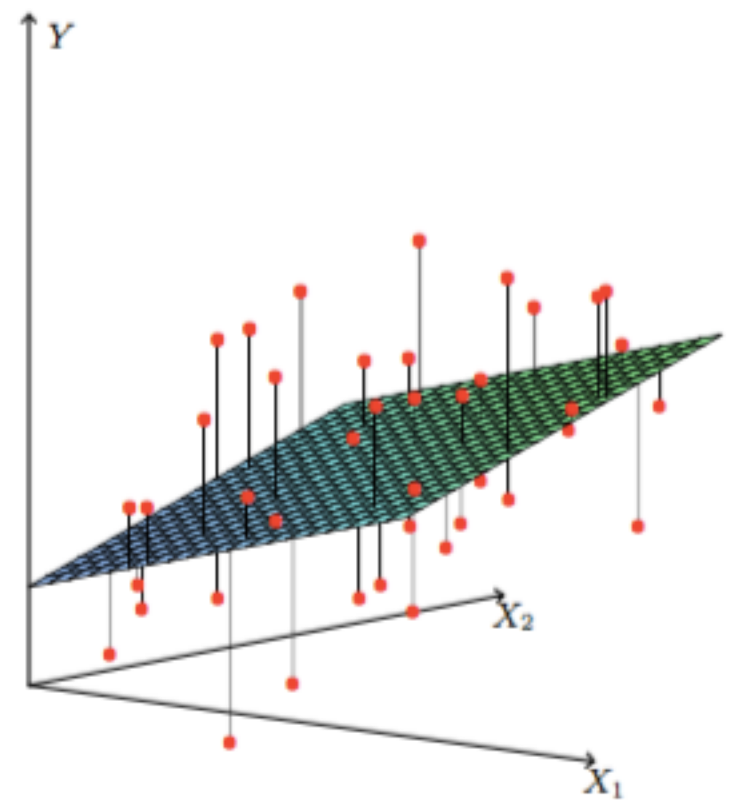
Linear Regression

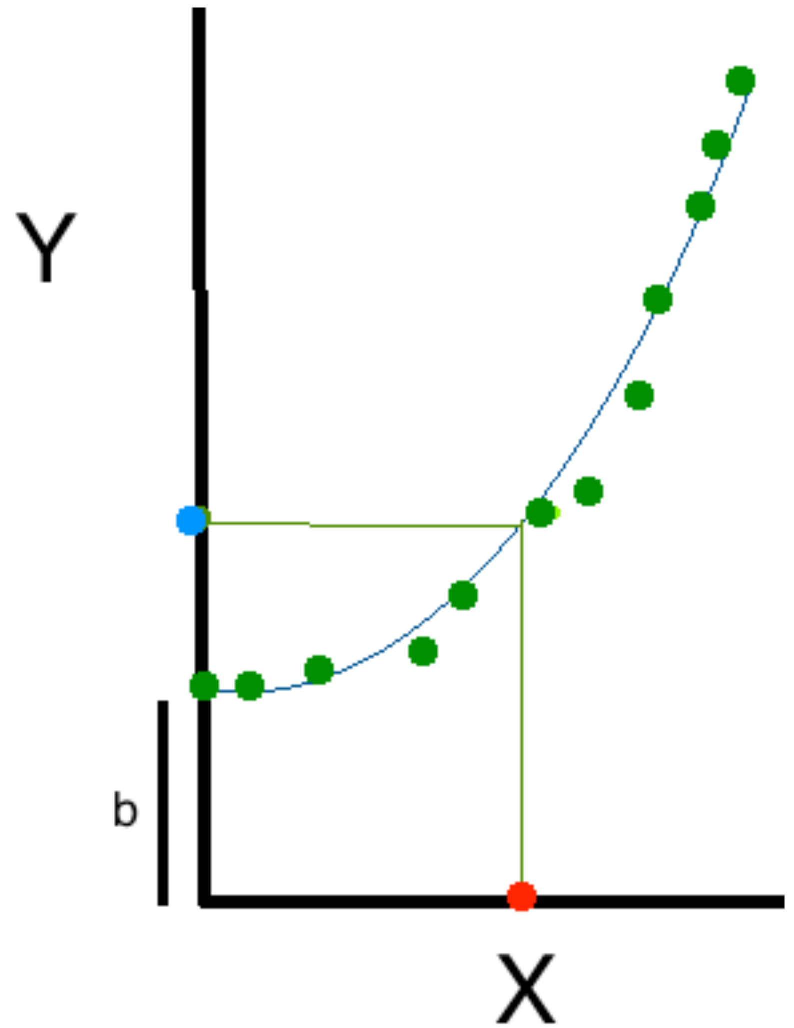
- Learn linear function **$f(x) = w^T x$** such that y=f(x) for all inputs



Linear Regression

- Learn linear function **$f(x) = w^T x$** such that y=f(x) for all inputs





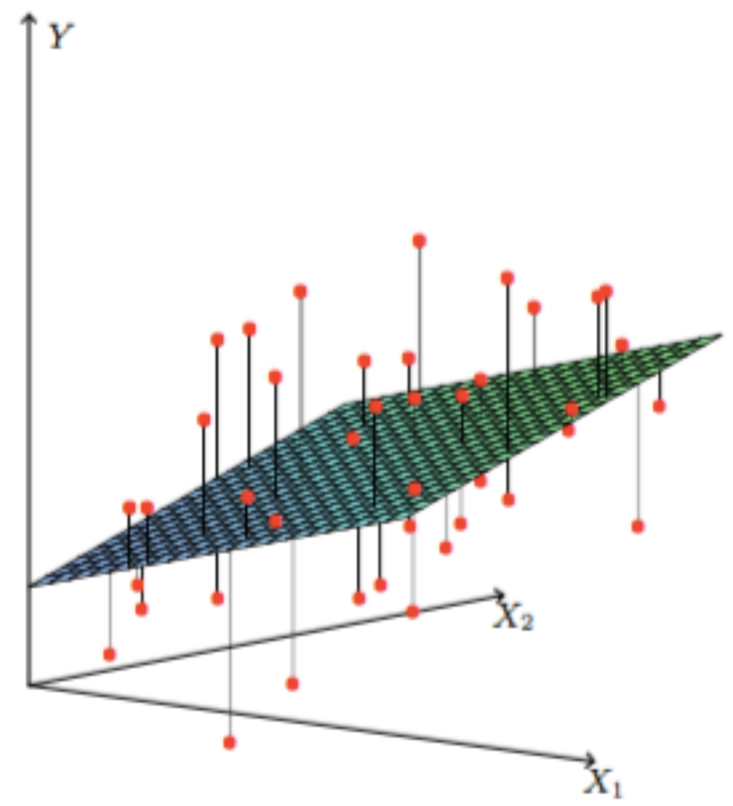
Is then linear regression Important ?

- $y=x^2+ b$ (intuitively)
- Define new set of features $x=[x^2,1]$
- **Radial Basis Function**

Linear Regression

Learn linear function **$f(x) = w^T x = w_1 x_1 + \ldots + w_p x_p$**

such that y=f(x) for all inputs



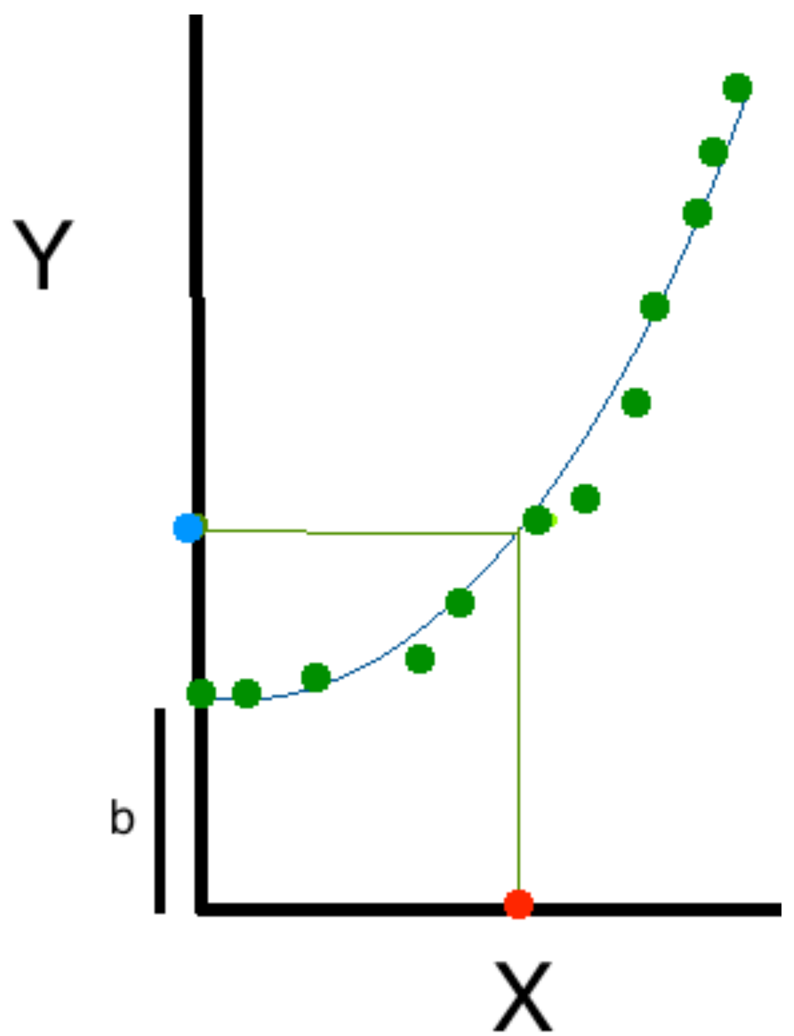
Is then linear regression Important ?

- $y = x^2 + b$ (intuitively)
- Define new set of features $x = [x^2, 1]$
- **Radial Basis Function :**
  - A radial basis function (RBF) is a **real-valued function** measure a **distance** from some other point c (known as centre)

$$\phi(\mathbf{x}, \mathbf{c}) = \phi(\|\mathbf{x} - \mathbf{c}\|)$$

  - Gaussian $\phi(r) = e^{-(\varepsilon r)^2}$

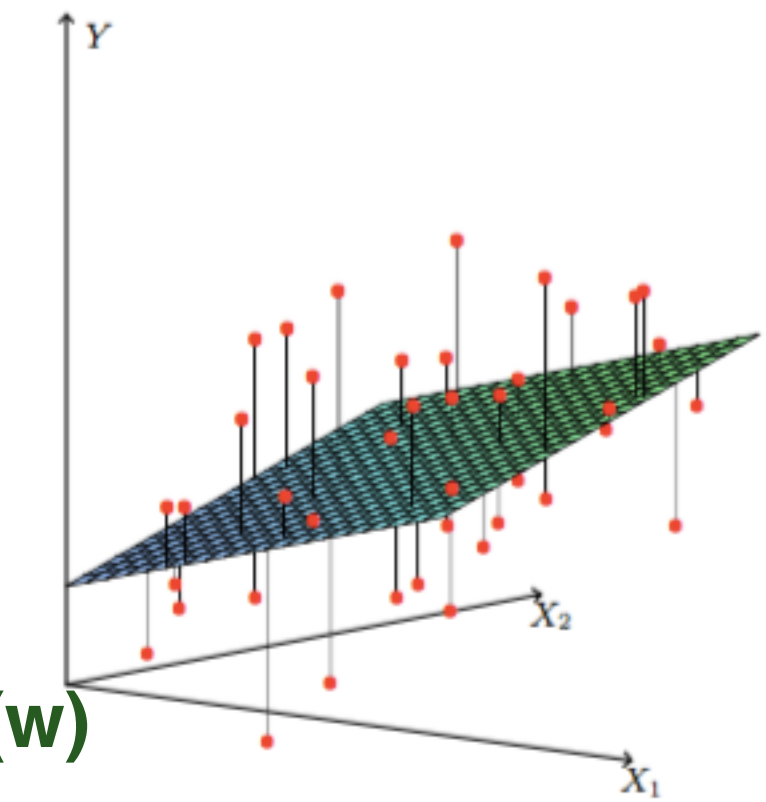
$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \, \phi(\|\mathbf{x} - \mathbf{x}_i\|)$$



Multivariate Linear Regression

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_p, y_p)\}$$

$$y = w^T x = w_1 x_1 + \ldots + w_p x_p + \varepsilon$$



- Our goal is to estimate w from a training data of $<x_i,y_i>$ pairs
- This could be done using a least squares approach

$$\arg\min_w \sum_i (y_i - w^T x_i)^2 \quad \textbf{= Loss(w)}$$

- Why least squares?

  - minimizes squared distance between measurements and predicted line

  - has a nice probabilistic interpretation

  - easy to compute

If the noise is Gaussian with mean 0 then least squares is also the maximum likelihood estimate of w

Multivariate Linear Regression

$$\mathbf{w} = \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_p \end{pmatrix} \quad X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_p \\ 1 & x_{21} & \cdots & x_p \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & \cdots & x_p \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- We can thus re-write our model as $\mathbf{y} = \mathbf{Xw} + \boldsymbol{\varepsilon}$
- The solution turns out to be: $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$
- This is an instance of a larger set of computational solutions which are usually referred to as ‘generalized least squares’

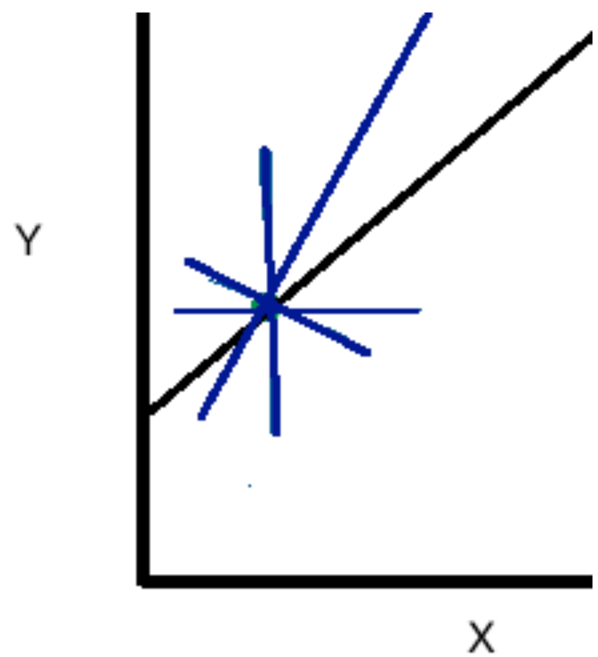
Multivariate Linear Regression

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- We can thus re-write our model as $\mathbf{y = Xw+\varepsilon}$
- The solution turns out to be: $\mathbf{w = (X^T X)^{-1} X^T y}$
- This is an instance of a larger set of computational solutions which are usually referred to as 'generalized least squares'

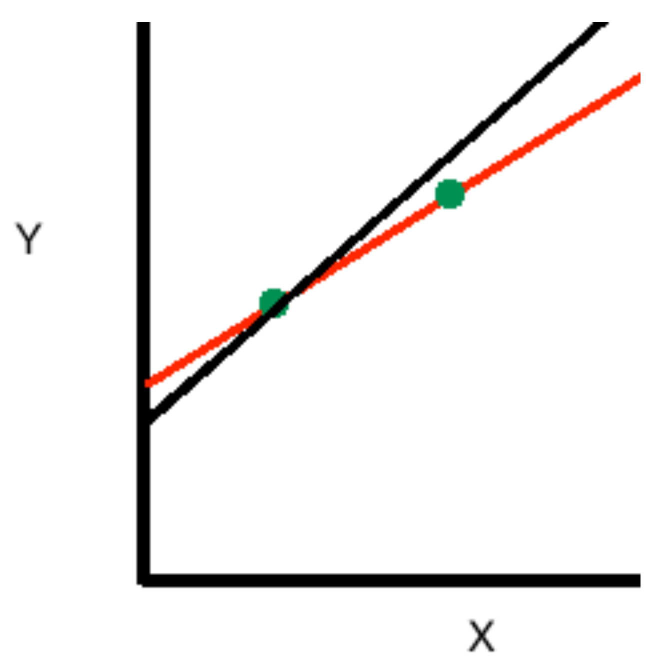
What happen if n<p ?

Regularisation



less than d+1 training samples present for d diminutional data: many possible lines

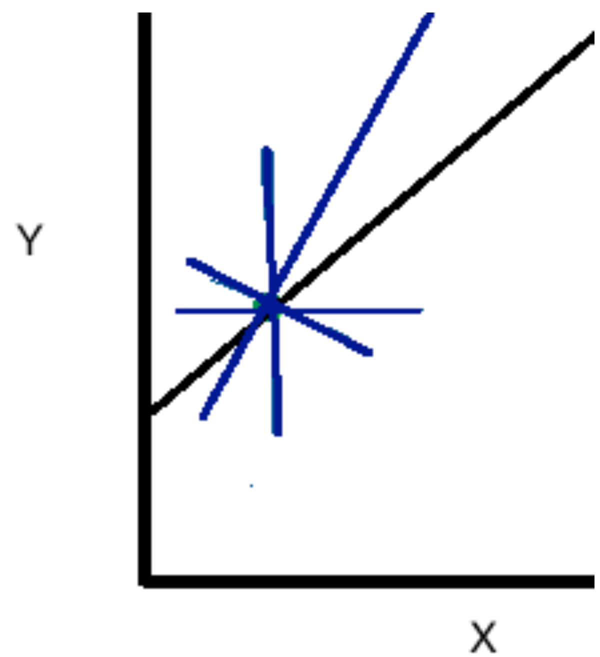
Insufficient data



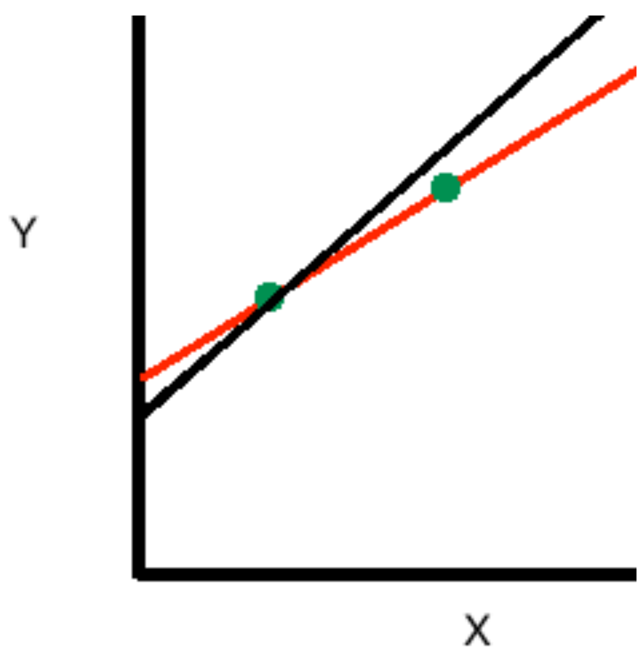
d+1 training samples present for d diminutional data: one line passing through all points

Overfitting

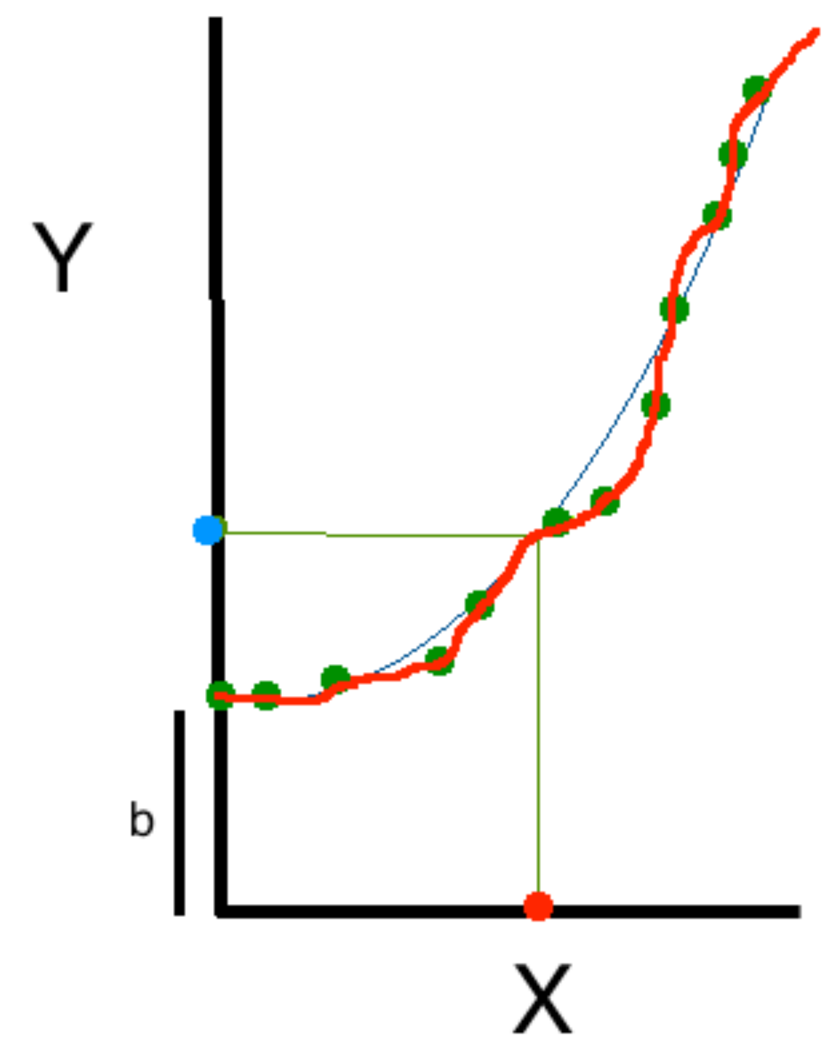
Regularisation



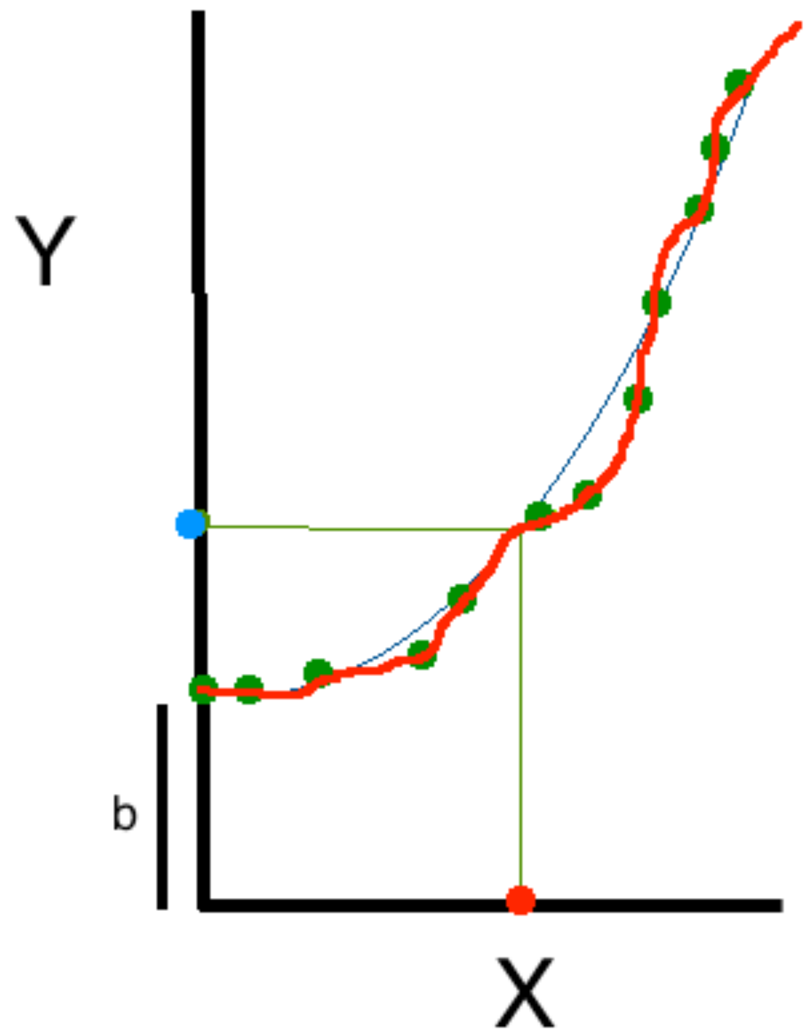
Insufficient data



Overfitting



Occam's Razor



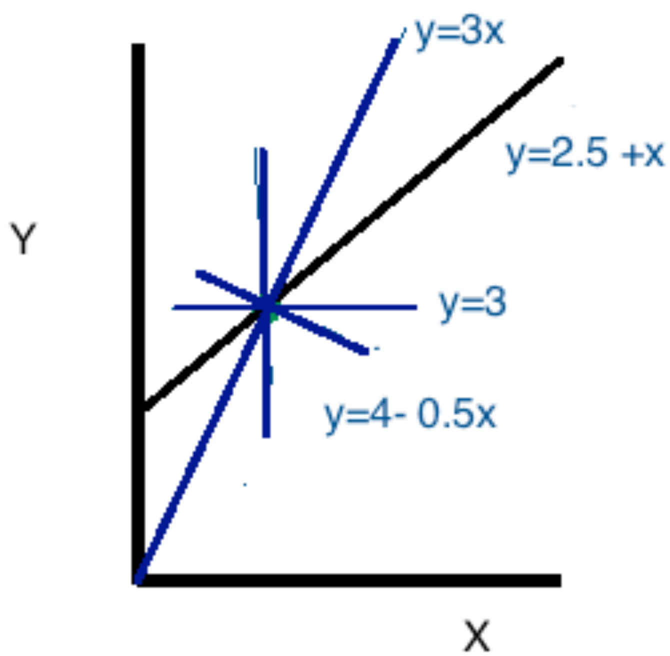
William of Ockham (1285-1349)
Principle of Parsimony:

– "One should not increase, beyond what is necessary, the number of entities required to explain anything."

Regularisation

Considering additional property of function

- Minimize the complexity of function
- Minimize **norm of vector w**
    - **Ridge Regression :** consider $\|w\|_2$
    - **LASSO** (Least Absolute Shrinkage and Selection Operator): consider $\|w\|_1$



Ridge Regression	Lasso
3	**3**
sqrt(7.25)	3.5
3	**3**
sqrt(16.25)	4.5

Regularised Linear Regression

- Advantage
  - Avoid overfitting
  - Useful if n<p

$$\boxed{E(\mathbf{w}) = \ell(\mathbf{X}, \mathbf{Y}, \mathbf{w}) + R(\mathbf{w})}$$

- Minimize sum of **loss function (model fit)** and a **regularisation (not too complex)** term
- λ is the tradeoff parameter. It control how much to regularisation

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

Ridge Regression

$$\boxed{\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}}$$

LASSO

?

Model Selection

1. Selecting ML method
   - Regression
     - Linear Regression
     - Ridge regression
     - Lasso
   - Classifier
     - SVM
     - Logistic regression
   - Clustering ….
2. Selecting **features or basis function**
3. Selecting **hyper-parameter**
   - λ

Validation

IDEA : Model should perform well on **UNSEEN DATA.**

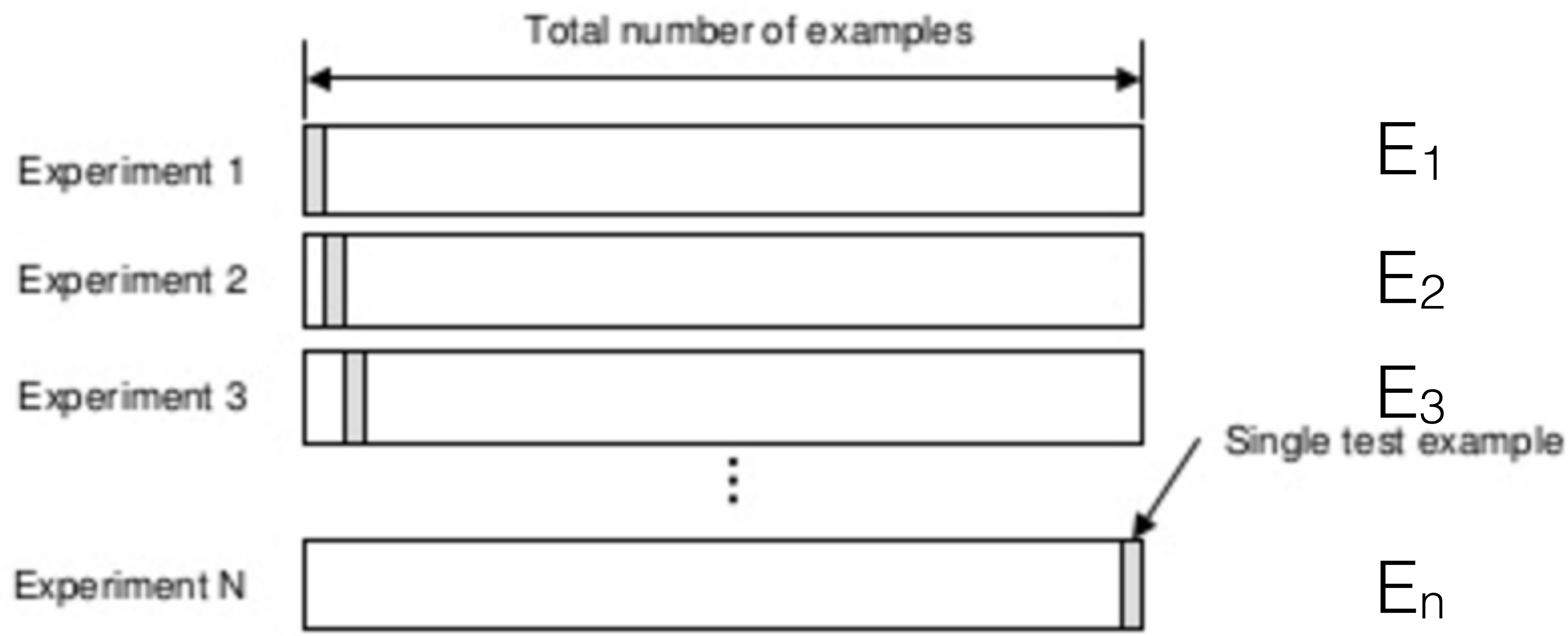
Given a set of data points divide it in two randomly selected part

- Training data set
- Test data set :
  - Test data set must be **independent** of training data set
  - Learning on training data and evaluation on test data

VALIDATION : validate the leaner not hypothesis (no proof)

- Validation is a part of learning
- Randomly divide training data in to 2 parts
  - training data set and validation set
  - Learn using training set
  - Find Error on validation set
  - Selecte hyper-parameters where **validation error is MINIMUM**

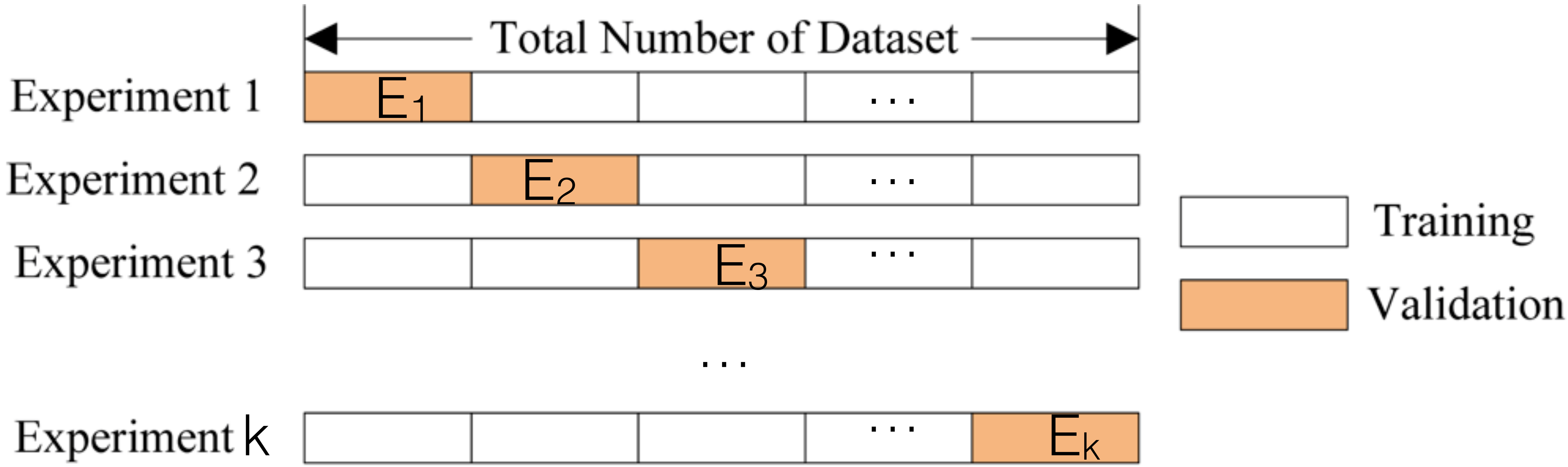
Leave One Out Cross Validation



LOO error : $1/n\ (E_1 + E_2 + \ldots + E_i + \ldots + E_n)$

LOO cross validation is (almost) unbiased estimate of true error!

k-fold Cross validation



$E_1 = Ex_1 + Ex_2 \ldots + E_{Xnk})$

CV error : $1/n\ (E_1 + E_2 + \ldots + E_k)$

k-fold cross validation is faster than LOO.

Alternative optimisation

Ridge Regression :

$$\text{minimize}_{\mathbf{w}} \ \frac{1}{2}(\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

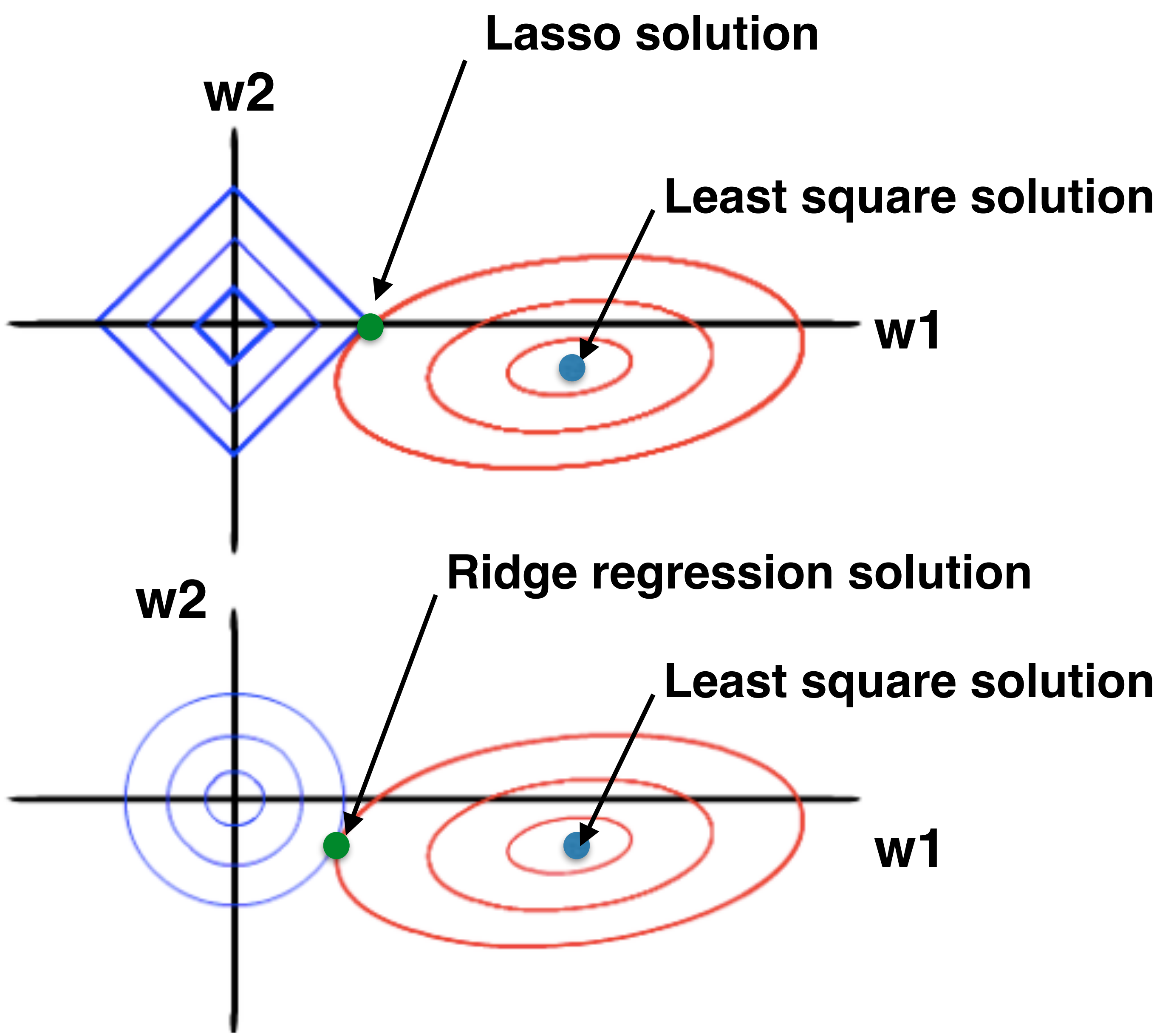
Consider regulariser as a constraint in optimisation problem

$$\text{minimize}_{\mathbf{w}} \ \frac{1}{2}(\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) .$$

$$\text{such that} \quad \mathbf{w}^T\mathbf{w} < \mathbf{t}$$

There is a relationship between t and λ.

LASSO : Sparse Solution



LASSO : Sparse Solution

LASSO : Least Absolute Shrinkage and Selection Opertor

$$E(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda\|\mathbf{w}\|_1$$

Solution for least square regression : $\mathbf{w}^{ls} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

Solution for LASSO :

$$\mathbf{w}^{lasso} = sign(\mathbf{w}^{ls})(|\mathbf{w}^{ls}| - \lambda)^+$$

Gradient Descent

Linear Regression $\mathbf{w}^{ls} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

Ridge Regression $\mathbf{w}^{rigde} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$

Lasso $\mathbf{w}^{lasso} = sign(\mathbf{w}^{ls})(|\mathbf{w}^{ls}| - \lambda)^+$

- Need matrix inverse :
  - Not possible when p is large
- Solution : iteratively minimising the loss function.
- How : Using Gradient descent

Gradient Descent

$$E(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y})$$

Initialize the weight vector $\mathbf{w} = \mathbf{w}^0$
Update $\mathbf{w}$ by moving along the direction of negative gradient $-\frac{\partial \mathbf{E}}{\partial \mathbf{w}}$

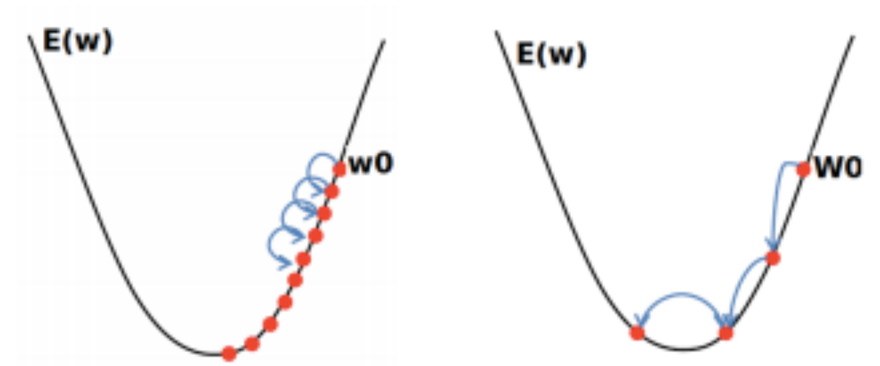
Initialize $\mathbf{w} = \mathbf{w}^0$

Repeat until convergence:

$$\begin{aligned}\mathbf{w} &= \mathbf{w} - \alpha \frac{\partial \mathbf{E}}{\partial \mathbf{w}} \\ &= \mathbf{w} - \alpha \mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{Y}) \\ &= \mathbf{w} - \alpha \sum_{i=1}^{N} \mathbf{x}_i(\mathbf{w}^T\mathbf{x}_i - y_i)\end{aligned}$$

α is the learning rate

It has a unique minimum



Probability Basic

Random Variables and Densities

- Random variables X represents outcomes or states of world.
  Instantiations of variables usually in lower case: x
  We will write $p(x)$ to mean probability($X = x$).
- Sample Space: the space of all possible outcomes/states.
  (May be discrete or continuous or mixed.)
- Probability mass (density) function $p(x) \geq 0$
  Assigns a non-negative number to each point in sample space.
  Sums (integrates) to unity: $\sum_x p(x) = 1$ or $\int_x p(x)dx = 1$.
  Intuitively: how often does x occur, how much do we believe in x.
- Ensemble: random variable $+$ sample space$+$ probability function

Probability Basic

Expectations, Moments

- Expectation of a function $a(x)$ is written $E[a]$ or $\langle a \rangle$

$$E[a] = \langle a \rangle = \sum_x p(x)a(x)$$

  e.g. mean $= \sum_x xp(x)$, variance $= \sum_x (x - E[x])^2 p(x)$

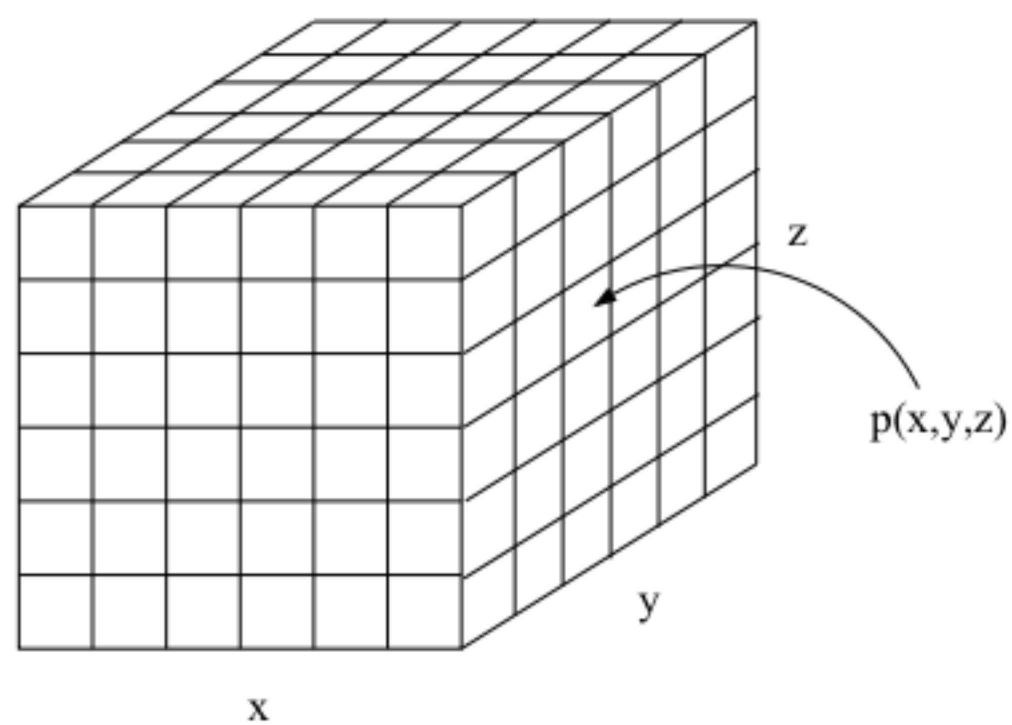
- Moments are expectations of higher order powers.
  (Mean is first moment. Autocorrelation is second moment.)
- Centralized moments have lower moments subtracted away (e.g. variance, skew, curtosis).
- Deep fact: Knowledge of all orders of moments completely defines the entire distribution.

Probability Basic

Joint Probability

- Key concept: two or more random variables may interact. Thus, the probability of one taking on a certain value depends on which value(s) the others are taking.
- We call this a joint ensemble and write

$$p(x, y) = \text{prob}(X = x \text{ and } Y = y)$$

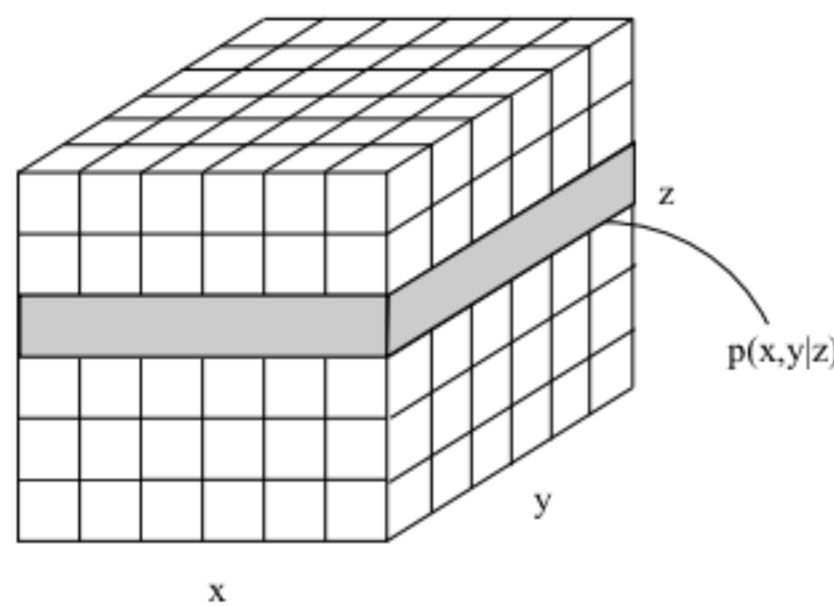


Probability Basic

Conditional Probability

- If we know that some event has occurred, it changes our belief about the probability of other events.
- This is like taking a "slice" through the joint table.

$$\boxed{p(x|y) = p(x, y)/p(y)}$$



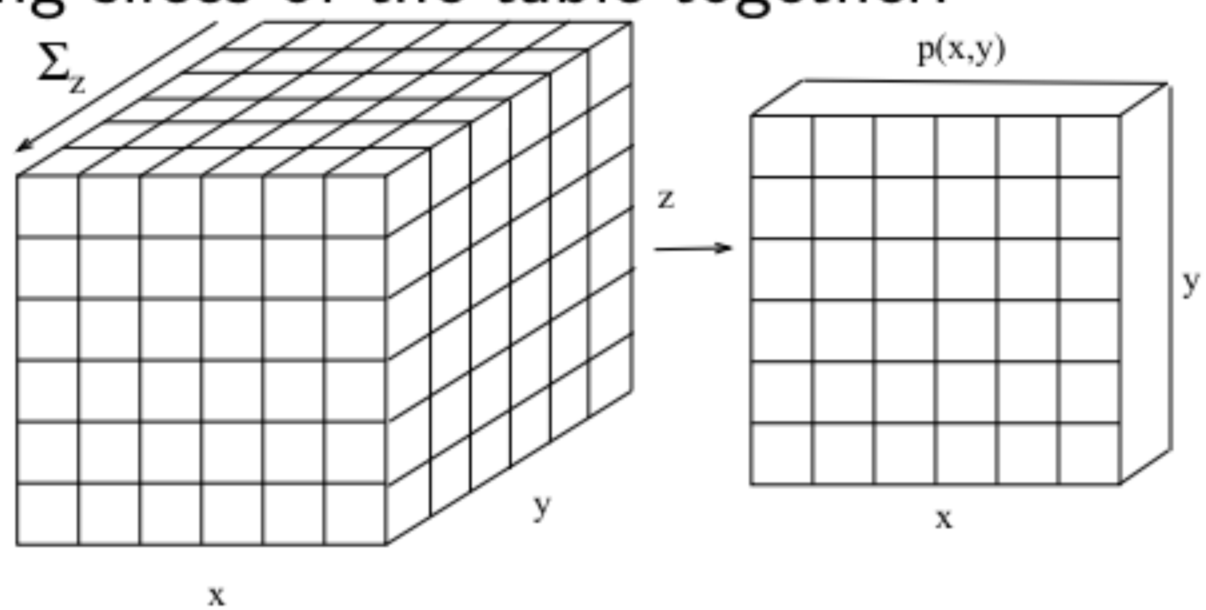
Probability Basic

Marginal Probabilities

- We can "sum out" part of a joint distribution to get the *marginal distribution* of a subset of variables:

$$p(x) = \sum_{y} p(x, y)$$

- This is like adding slices of the table together.



- Another equivalent definition: $p(x) = \sum_y p(x|y)p(y)$.

Probability Basic

Bayes' Rule

- Manipulating the basic definition of conditional probability gives one of the most important formulas in probability theory:

$$\boxed{p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')}}$$

- This gives us a way of "reversing" conditional probabilities.
- Thus, all joint probabilities can be factored by selecting an ordering for the random variables and using the "chain rule":

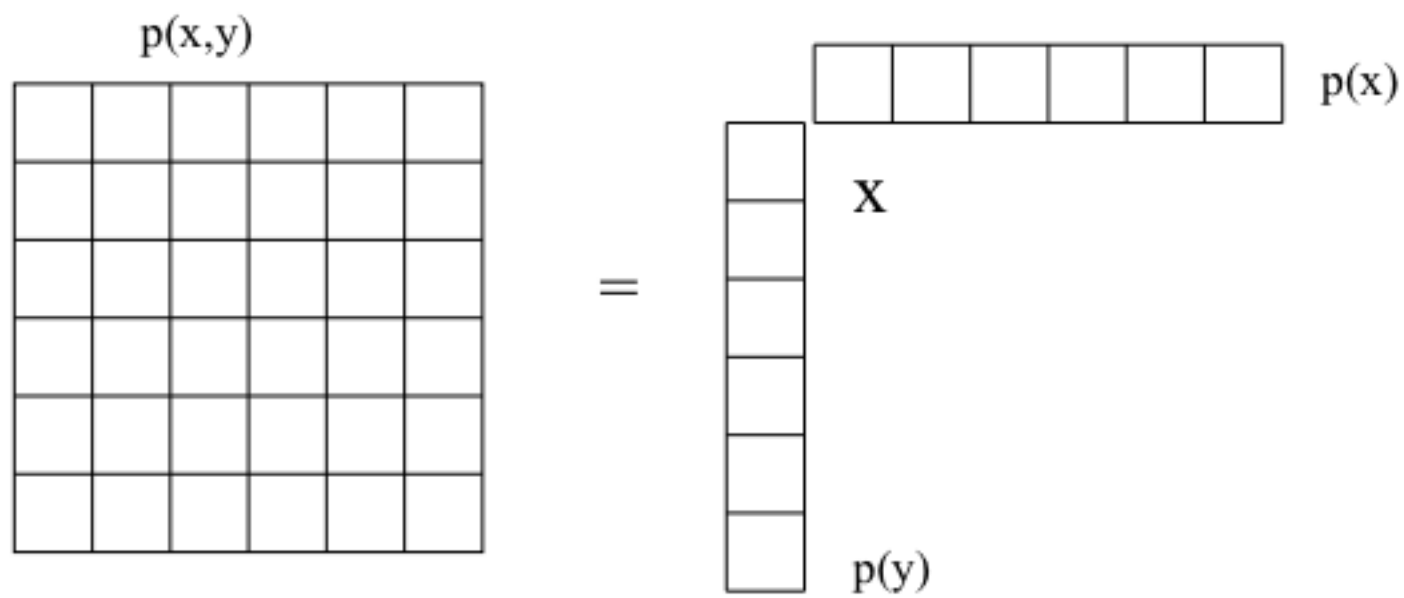
$$\boxed{p(x, y, z, \ldots) = p(x)p(y|x)p(z|x, y)p(\ldots|x, y, z)}$$

Probability Basic

Independence & Conditional Independence

- Two variables are independent iff their joint factors:

$$p(x, y) = p(x)p(y)$$



- Two variables are conditionally independent given a third one if for all values of the conditioning variable, the resulting slice factors:

$$p(x, y|z) = p(x|z)p(y|z) \qquad \forall z$$

Probability Basic

Bernoulli

- For a binary random variable with p(heads)=π:

$$p(x|\pi) = \pi^x (1-\pi)^{1-x}$$

Multinomial

- For a set of integer counts on k trials

$$p(\mathbf{x}|\pi) = \frac{k!}{x_1! x_2! \cdots x_n!} \pi_1^{x_1} \pi_2^{x_2} \cdots \pi_n^{x_n} = h(\mathbf{x}) \exp \left\{ \sum_i x_i \log \pi_i \right\}$$

- But the parameters are constrained: $\sum_i \pi_i = 1$.

Maximum Likelihood Estimation(MLE)

Bernoulli distribution

$$p(D / \theta) = \theta^{x_1}(1-\theta)^{(1-x_1)} \cdots \theta^{x_n}(1-\theta)^{(1-x_n)} = \theta^{(x_1+\cdots+x_n)}(1-\theta)^{n-(x_1+\cdots+x_n)}.$$

Log likelihood function

$$\ln p(D / \theta) = \ln\theta(\sum_{i=1}^{n} x_i) + \ln(1-\theta)(n - \sum_{i=1}^{n} x_i) = n\bar{x}\ln\theta + n(1-\bar{x})\ln(1-\theta).$$

MLE

$$\hat{\theta}(\mathbf{x}) = \bar{x}.$$

Probability Basic

Gaussian (normal)

- For a continuous univariate random variable:

$$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$
$$= \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{\frac{\mu x}{\sigma^2} - \frac{x^2}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} - \log\sigma\right\}$$

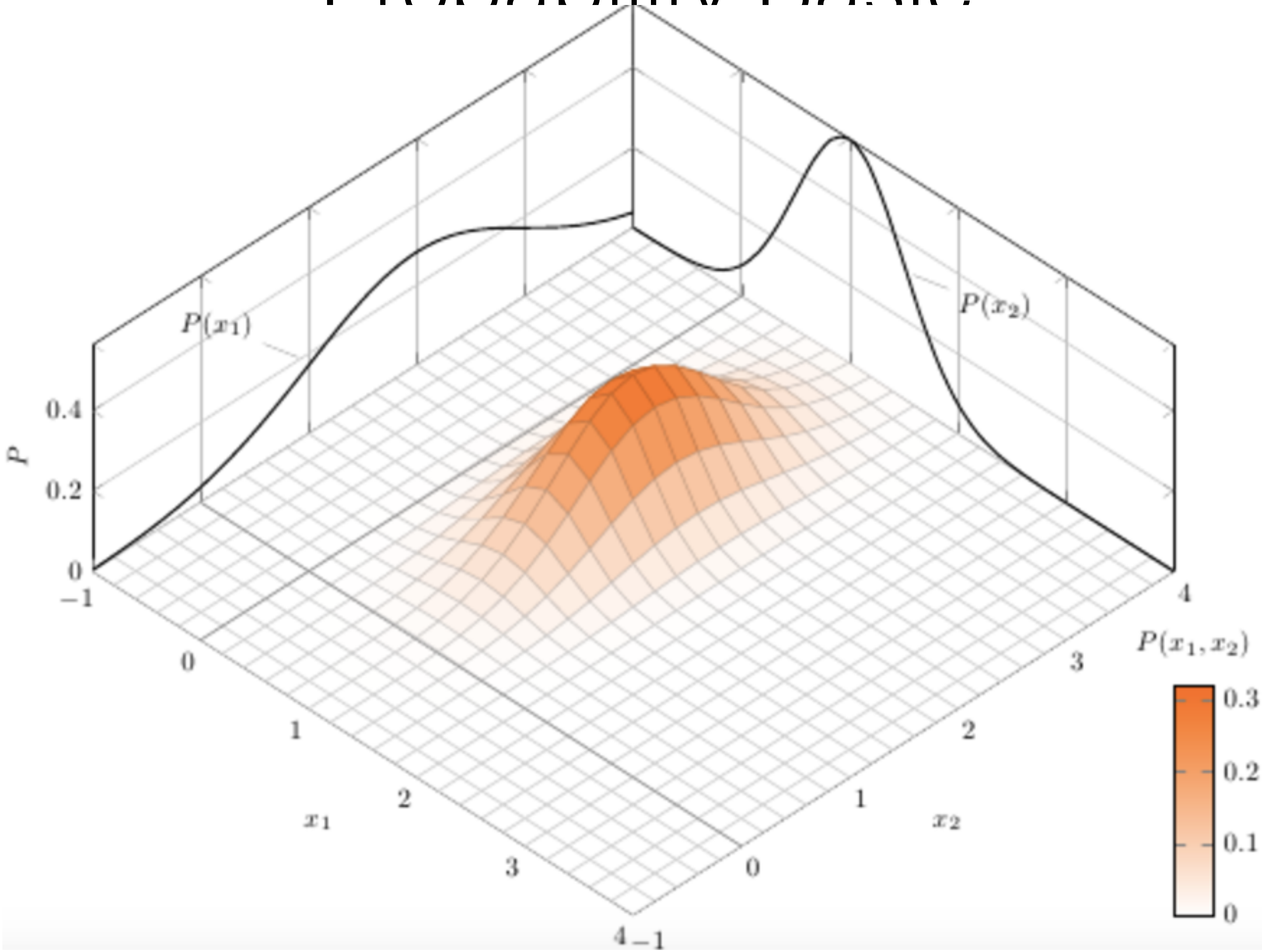
Multivariate Gaussian Distribution

- For a continuous vector random variable:

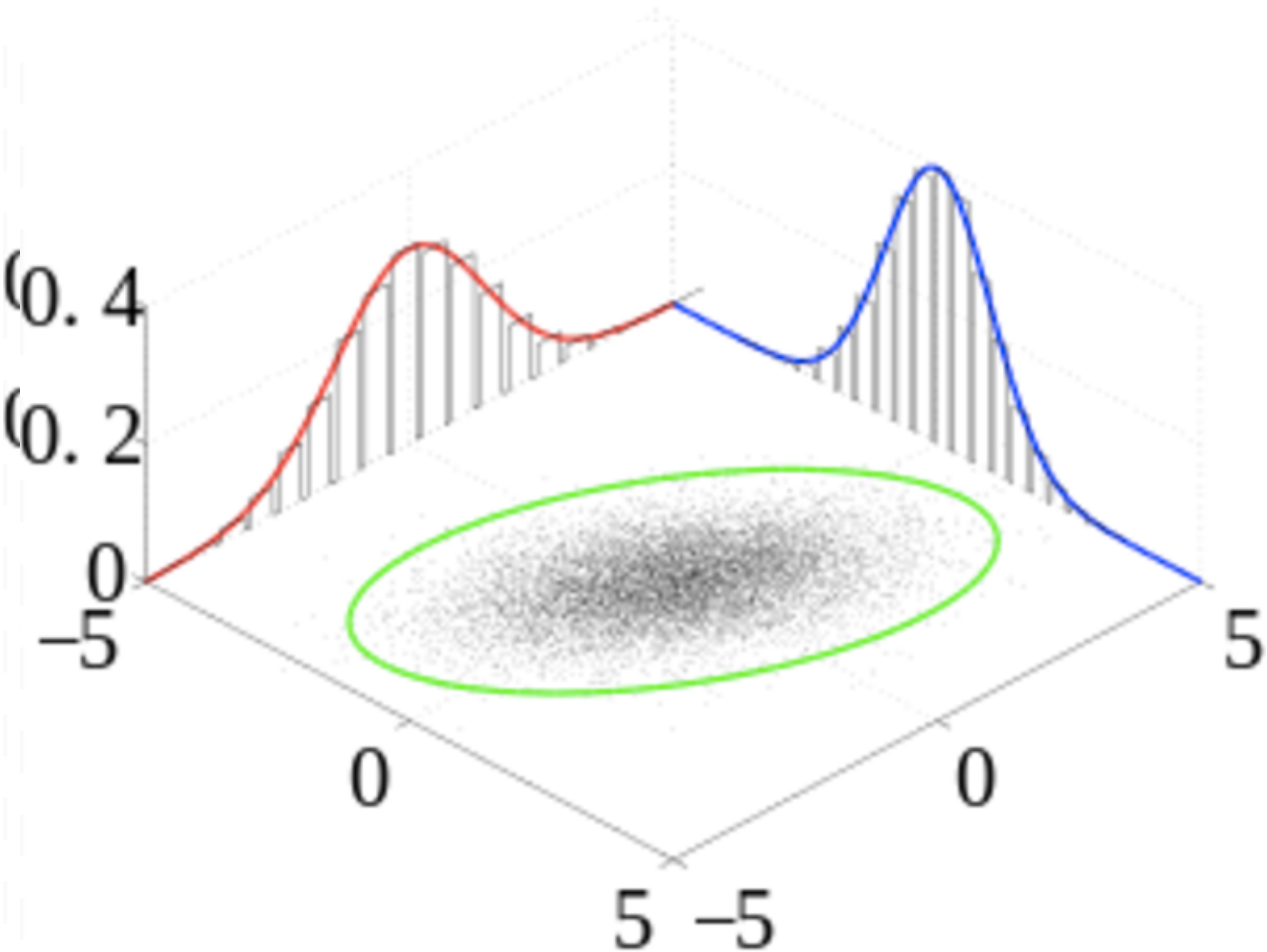
$$p(x|\mu,\Sigma) = |2\pi\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^\top\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

Distribution with maximum entropy for fixed variance

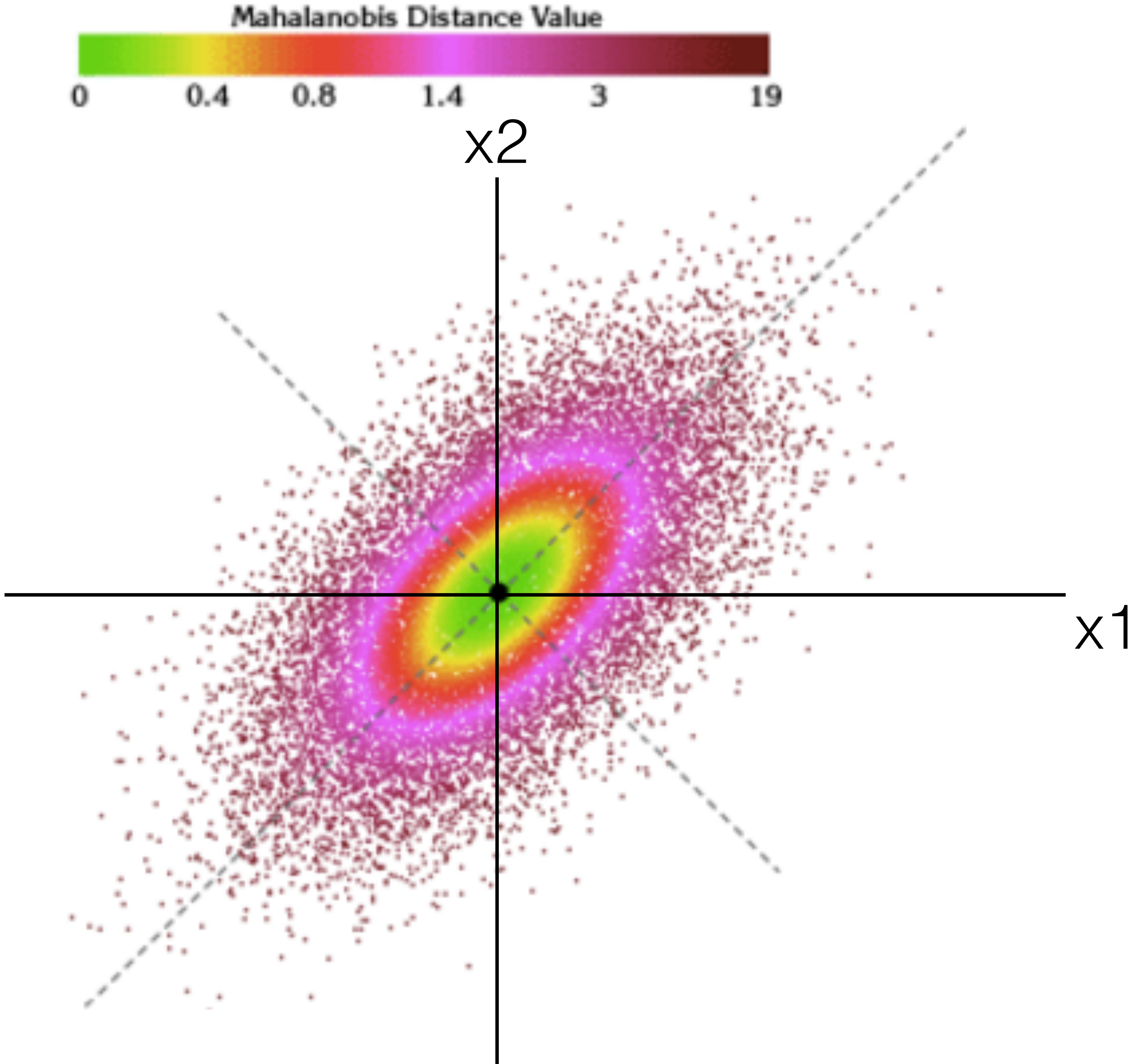
Probability Basic



Probability Basic



Probability Basic

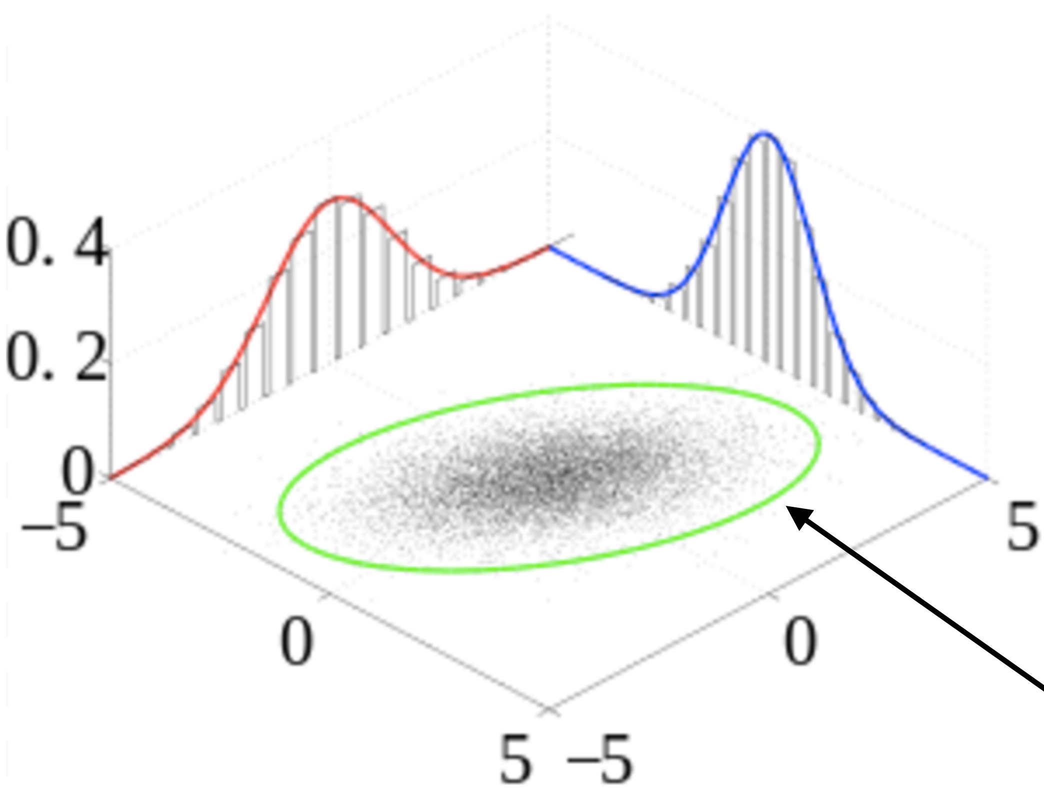


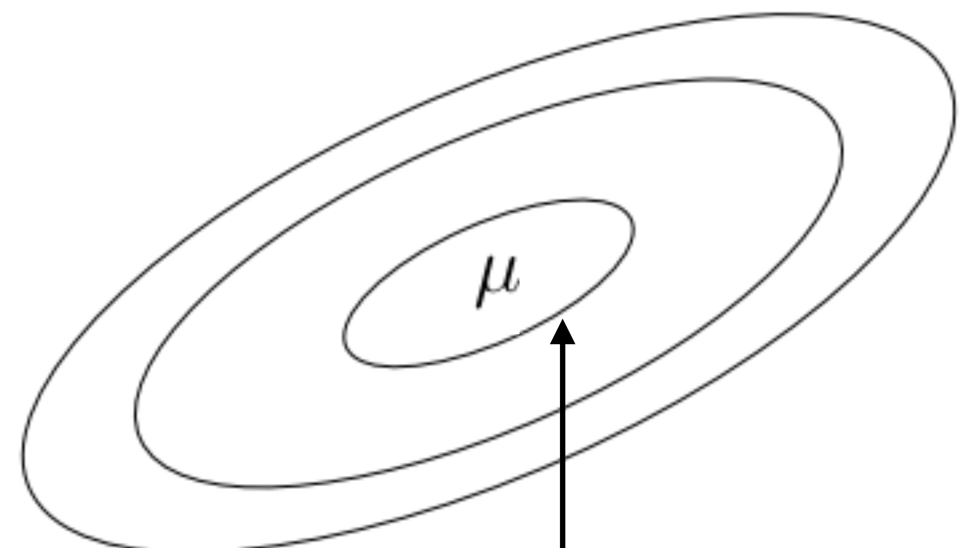
Probability Basic

Multivariate Gaussian Distribution

- For a continuous vector random variable:

$$p(x|\mu,\Sigma) = |2\pi\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^\top\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$





$$\underbrace{(\mathbf{x}-\mu)^\top\Sigma^{-1}(\mathbf{x}-\mu)}_{\textbf{Mahalanobis distance}} = C$$

Maximum Likelihood Estimation(MLE)

Gaussian distribution

p($\mathcal{D}/\mu,\sigma^2$)

$$\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\frac{-(x_1-\mu)^2}{2\sigma^2}\right)\cdots\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\frac{-(x_n-\mu)^2}{2\sigma^2}\right)=\frac{1}{\sqrt{(2\pi\sigma^2)^n}}\exp-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2.$$

log likelihood function

$$\ln p(D/\mu,\sigma^2)=-\frac{n}{2}\ln 2\pi\sigma^2-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2.$$

$$\frac{\partial}{\partial\mu}\ln p(\mathcal{D}/\mu,\sigma^2):\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu)=.\frac{1}{\sigma^2}n(\bar{x}-\mu)$$

$$\frac{\partial}{\partial\sigma^2}\ln p(\mathcal{D}/\mu,\sigma^2)=-\frac{n}{\sigma^2}+\frac{1}{(\sigma^2)^2}\sum_{i=1}^{n}(x_i-\mu)^2=\frac{n}{(\sigma^2)^2}\left(\sigma^2-\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2\right).$$

MLE

$$\hat{\mu}(\mathbf{x})=\bar{x}$$

$$\hat{\sigma}^2(\mathbf{x})=\frac{1}{n}\sum_{i=1}^{n}(x_i-\hat{x})^2.$$

I.I.D samples of the data

Assume data generated via a probabilistic model

$$\mathbf{d} \sim P(\mathbf{d} \mid \theta)$$

$P(\mathbf{d} \mid \theta)$: Probability distribution underlying the data

- θ: fixed but unknown distribution parameter

Given: N independent and identically distributed (i.i.d.) samples of the data

$$\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$$

$$\mathbf{d}_\mathbf{i} = \{\mathbf{x}_\mathbf{i}, \mathbf{y}_\mathbf{i}\}$$

Independent and Identically Distributed:

- Given θ, each sample $\mathbf{d}_\mathbf{i}$ is independent of all other samples
- All samples $\mathbf{d}_\mathbf{i}$ drawn from the same distribution

Goal: Estimate parameter θ that best models/describes the data

Several ways to define the "best"

Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE): Choose the parameter θ that maximizes the probability of the data, *given* that parameter

Probability of the data, given the parameters is called the Likelihood, a function of θ and defined as:

$$\mathcal{L}(\theta) = P(\mathcal{D} \mid \theta) = P(\mathbf{d}_1, \ldots, \mathbf{d}_N \mid \theta) = \prod_{i=1}^{N} P(\mathbf{d}_i \mid \theta)$$

MLE typically maximizes the Log-likelihood instead of the likelihood

Log-likelihood:

$$\log \mathcal{L}(\theta) = \log P(\mathcal{D} \mid \theta) = \log \prod_{i=1}^{N} P(\mathbf{d}_i \mid \theta) = \sum_{i=1}^{N} \log P(\mathbf{d}_i \mid \theta)$$

Maximum Likelihood parameter estimation

$$\boxed{\hat{\theta}_{MLE} = \arg\max_{\theta} \log \mathcal{L}(\theta) = \arg\max_{\theta} \sum_{i=1}^{N} \log P(\mathbf{d}_i \mid \theta)}$$

Maximum-a-posteriori Estimation

Maximum-a-Posteriori Estimation (MAP): Choose θ that maximizes the posterior probability of θ (i.e., probability in the light of the observed data)

Posterior probability of θ is given by the Bayes Rule

$$P(\theta \mid \mathcal{D}) = \frac{P(\theta)P(\mathcal{D} \mid \theta)}{P(\mathcal{D})}$$

$P(\theta)$: Prior probability of θ (without having seen any data)

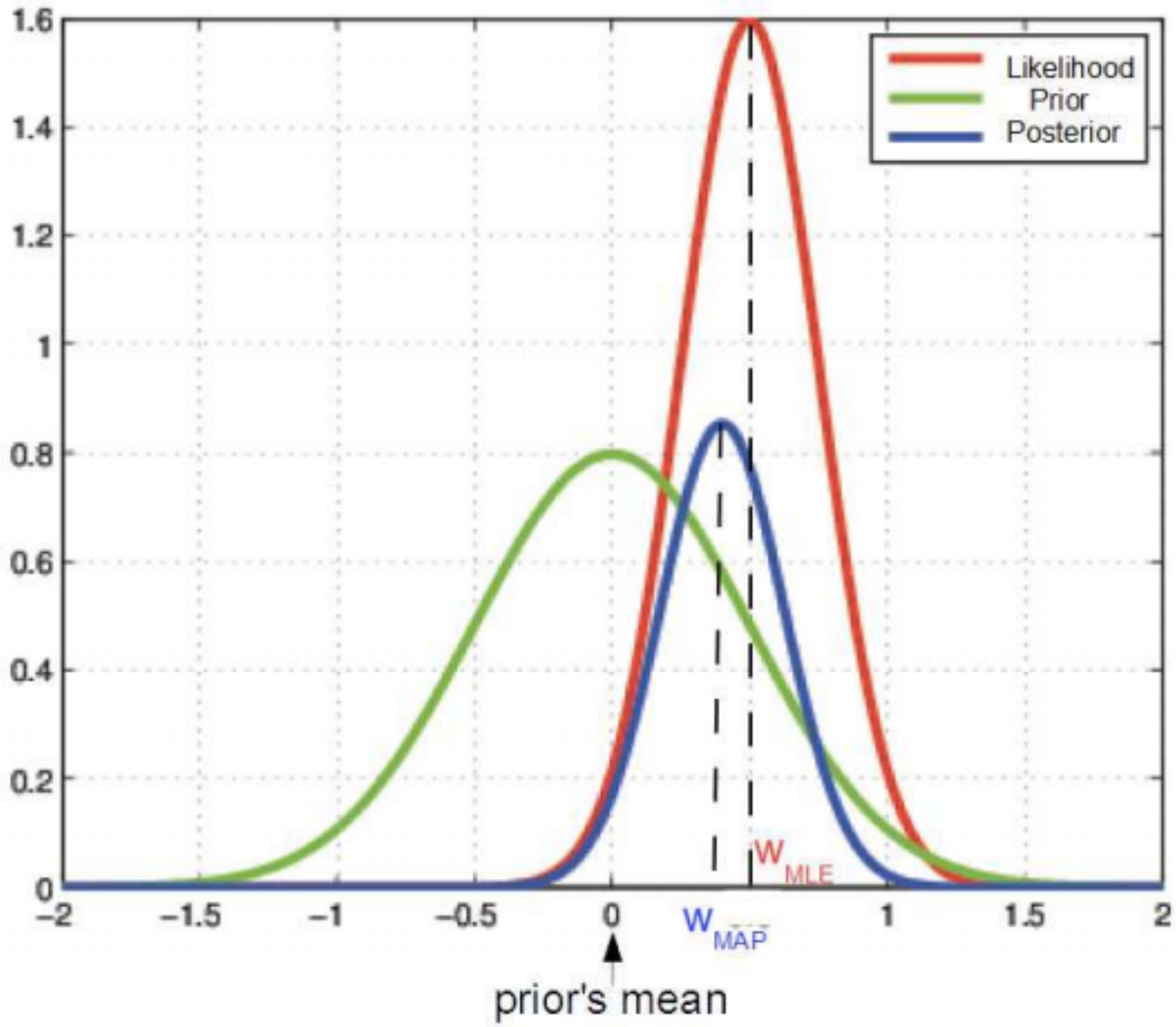
$P(\mathcal{D} \mid \theta)$: Likelihood

$P(\mathcal{D})$: Probability of the data (independent of θ)

$$P(\mathcal{D}) = \int P(\theta)P(\mathcal{D} \mid \theta)d\theta \quad \text{(sum over all } \theta\text{'s)}$$

The Bayes Rule lets us update our belief about θ in the light of observed data

While doing MAP, we usually maximize the log of the posterior probability



1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−2
−1.5
−1
−0.5
0
1
1.5
2
Likelihood
Prior
Posterior
W
MLE
W
MAP
prior's mean

Maximum-a-posteriori Estimation

Maximum-a-Posteriori parameter estimation

$$
\begin{aligned}
\hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta \mid \mathcal{D}) &= \arg\max_{\theta} \frac{P(\theta)P(\mathcal{D} \mid \theta)}{P(\mathcal{D})} \\
&= \arg\max_{\theta} P(\theta)P(\mathcal{D} \mid \theta) \\
&= \arg\max_{\theta} \log P(\theta)P(\mathcal{D} \mid \theta) \\
&= \arg\max_{\theta} \{\log P(\theta) + \log P(\mathcal{D} \mid \theta)\}
\end{aligned}
$$

$$
\boxed{\hat{\theta}_{MAP} = \arg\max_{\theta} \{\log P(\theta) + \sum_{i=1}^{N} \log P(\mathbf{d}_i \mid \theta)\}}
$$

Same as MLE except the extra log-prior-distribution term!

MAP allows incorporating our prior knowledge about θ in its estimation

Linear Regression

Each response generated by a linear model plus some Gaussian noise

$$y = \mathbf{w}^\top \mathbf{x} + \epsilon$$

Noise ϵ is drawn from a Gaussian distribution:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Each response y then becomes a draw from the following Gaussian:

$$y \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$$

Probability of each response variable

$$P(y \mid \mathbf{x}, \mathbf{w}) = \mathcal{N}(y \mid \mathbf{w}^\top \mathbf{x}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y - \mathbf{w}^\top \mathbf{x})^2}{2\sigma^2}\right]$$

Given data $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, we want to estimate the weight vector $\mathbf{w}$

Linear Regression : MLE

Log-likelihood:

$$\log \mathcal{L}(\mathbf{w}) = \log P(\mathcal{D} \mid \mathbf{w}) = \log P(\mathbf{Y} \mid \mathbf{X}, \mathbf{w}) = \log \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

$$= \sum_{i=1}^{N} \log P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

$$= \sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - \mathbf{w}^\top \mathbf{x}_i)^2}{2\sigma^2}\right]$$

$$= \sum_{i=1}^{N} \left\{-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y_i - \mathbf{w}^\top \mathbf{x}_i)^2}{2\sigma^2}\right\}$$

Maximum Likelihood Solution: $\hat{\mathbf{w}}_{MLE} = \arg\max_{\mathbf{w}} \log P(\mathcal{D} \mid \mathbf{w})$

$$= \arg\max_{\mathbf{w}} -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

$$= \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

For $\sigma = 1$ (or some constant) for each input, it's equivalent to the least-squares objective for linear regression

Linear Regression : MAP

Let's assume a Gaussian prior distribution over the weight vector $\mathbf{w}$

$$P(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid 0, \lambda^{-1}\mathbf{I}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{\lambda}{2}\mathbf{w}^\top \mathbf{w}\right)$$

Log posterior probability:

$$\log P(\mathbf{w} \mid \mathcal{D}) = \log \frac{P(\mathbf{w})P(\mathcal{D} \mid \mathbf{w})}{P(\mathcal{D})} = \log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w}) - \log P(\mathcal{D})$$

Maximum-a-Posteriori Solution: $\hat{\mathbf{w}}_{MAP} = \arg\max_{\mathbf{w}} \log P(\mathbf{w} \mid \mathcal{D})$

$$\begin{aligned}
&= \arg\max_{\mathbf{w}} \{\log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w}) - \log P(\mathcal{D})\} \\
&= \arg\max_{\mathbf{w}} \{\log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w})\} \\
&= \arg\max_{\mathbf{w}} \left\{-\frac{D}{2}\log(2\pi) - \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w} + \sum_{i=1}^{N}\left\{-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y_i - \mathbf{w}^\top \mathbf{x}_i)^2}{2\sigma^2}\right\}\right\} \\
&= \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w} \quad \text{(ignoring constants and changing max to min)}
\end{aligned}$$

For $\sigma = 1$ (or some constant) for each input, it's equivalent to the regularized least-squares objective

MLE vs MAP

MLE solution:

$$\boxed{\hat{\mathbf{w}}_{MLE} = \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2}$$

MAP solution:

$$\boxed{\hat{\mathbf{w}}_{MAP} = \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w}}$$

Take-home messages:

- MLE estimation of a parameter leads to unregularized solutions
- MAP estimation of a parameter leads to regularized solutions
- The prior distribution acts as a regularizer in MAP estimation

Note: For MAP, different prior distributions lead to different regularizers

- Gaussian prior on $\mathbf{w}$ regularizes the ℓ_2 norm of $\mathbf{w}$
- Laplace prior $\exp(-C||\mathbf{w}||_1)$ on $\mathbf{w}$ regularizes the ℓ_1 norm of $\mathbf{w}$