Mathematical Background

Vahid Tarokh

CEE 690/ECE 590, Fall 2019

Introduction

- In order to design and implement deep networks we need to know
    - Basics of Linear Algebra
    - Basics of Multivariable Calculus
    - Basics of Probability
- For writing research papers, you may need to know more.

Here, we will include some of the background for completeness, but will not teach them all.

- Source: **Dive into Deep Learning**
    - **Professor Smola's Slides**
    - **Professor David Carlson's Slides**
    - **Professor Lawrence Caron's slides**
    - **Professor Ruslan Salakhutdinov's slides (available online)**

Quick Review of Linear Algebra
(To Establish the Notation)

Scalars

- **Simple operations**

$$c = a + b$$

$$c = a \cdot b$$

$$c = \sin a$$

- **Length**

$$|a| = \begin{cases} a & \text{if} a > 0 \\ -a & \text{otherwise} \end{cases}$$

$$|a + b| \leq |a| + |b|$$

$$|a \cdot b| = |a| \cdot |b|$$

Vectors

- **Simple operations**

$$c = a + b \quad \text{where } c_i = a_i + b_i$$
$$c = \alpha \cdot b \quad \text{where } c_i = \alpha b_i$$
$$c = \sin a \quad \text{where } c_i = \sin a_i$$

Definition of a vector space

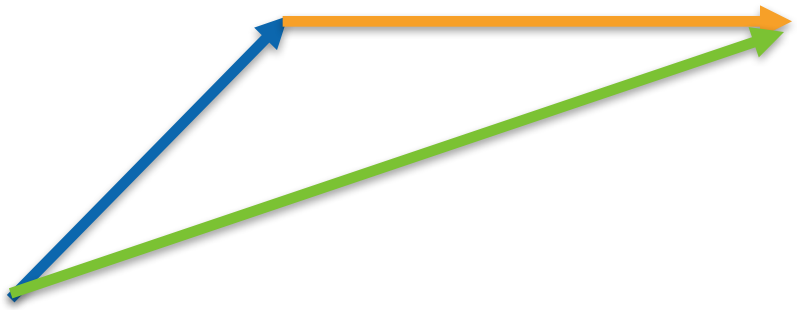
- **Length**

$$\|a\|_2 = \left[\sum_{i=1}^{m} a_i^2\right]^{\frac{1}{2}}$$

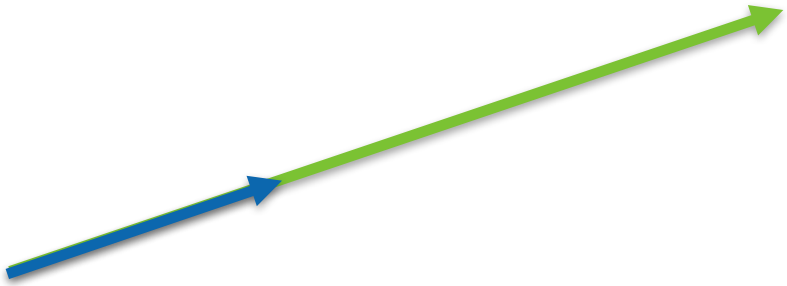
Definition of a norm

$$\|a\| \geq 0 \text{ for all } a$$
$$\|a + b\| \leq \|a\| + \|b\|$$
$$\|a \cdot b\| = |a| \cdot \|b\|$$

Vectors



$$c = a + b$$



$$c = \alpha \cdot b$$

Mathematician's **'parallel for all do'**

Vectors

- **Dot product**

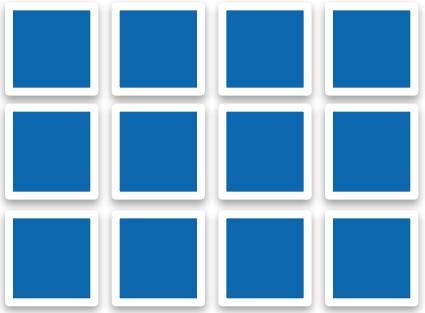
$$a^\top b = \sum_i a_i b_i$$

- **Orthogonality**

$$a^\top b = \sum_i a_i b_i = 0$$

(e.g. if we have two vectors that are orthogonal with a third, their linear combination is it, too)

Matrices



- **Simple operations**

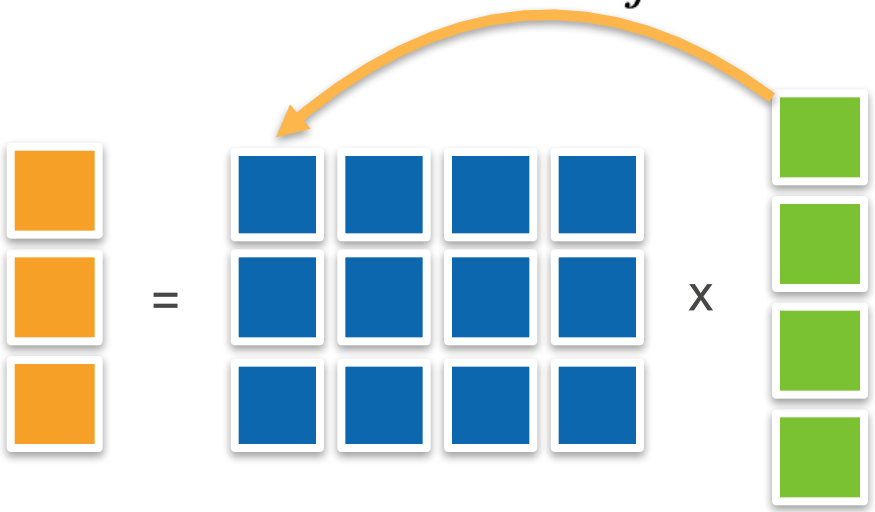
$$C = A + B \qquad \text{where } C_{ij} = A_{ij} + B_{ij}$$
$$C = \alpha \cdot B \qquad \text{where } C_{ij} = \alpha B_{ij}$$
$$C = \sin A \qquad \text{where } C_{ij} = \sin A_{ij}$$

- **Functional Analysis 101**
  vector = function, matrix = linear operator
  most theorems work sort-of in infinite dimensional spaces

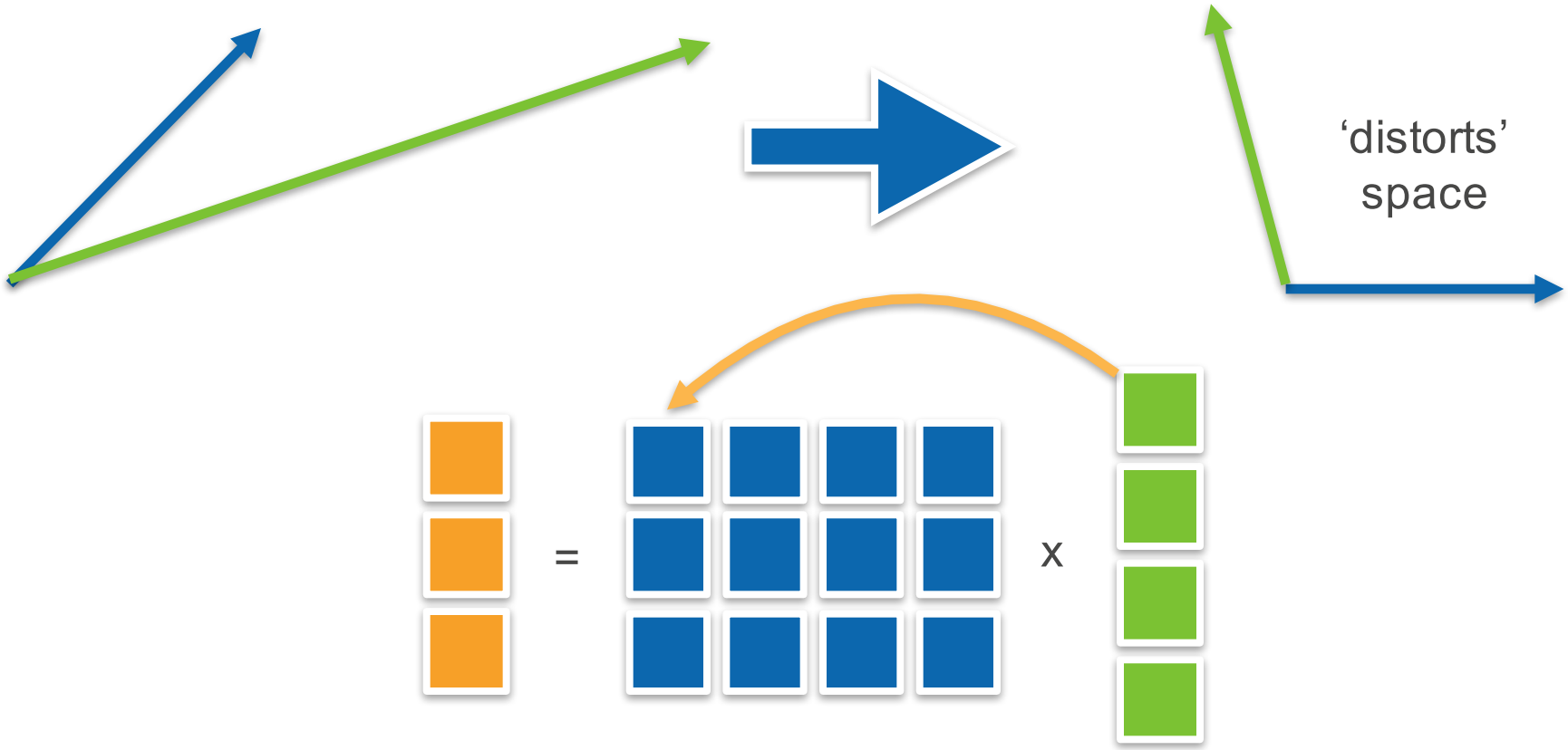
Matrices

- **Multiplications (matrix vector)**

$$c = Ab \text{ where } c_i = \sum_j A_{ij} b_j$$



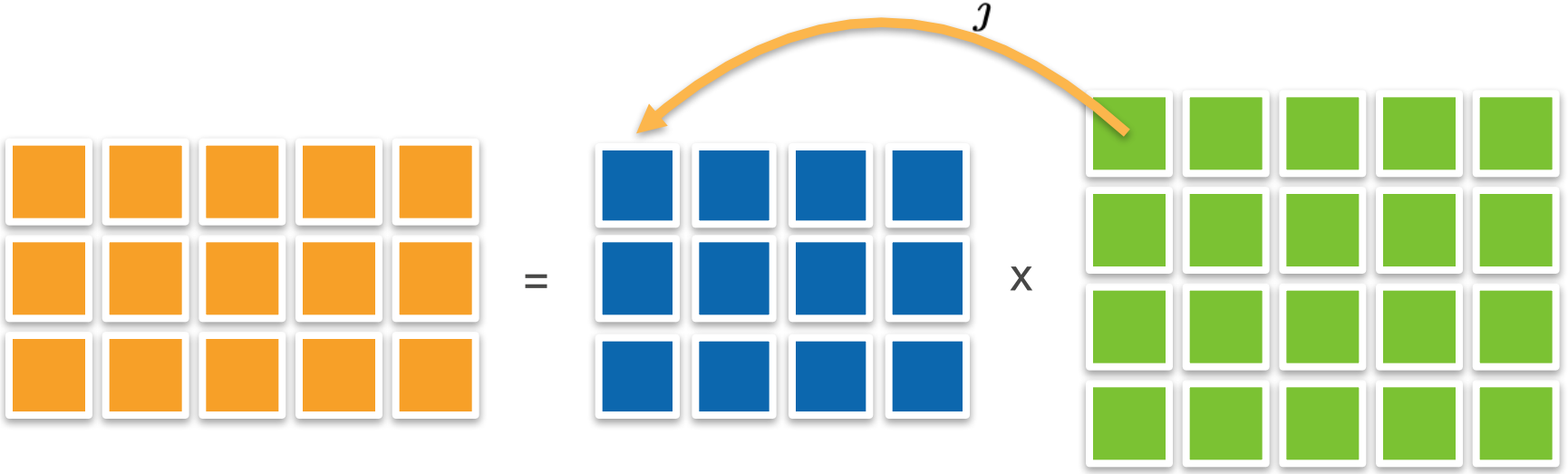
Matrices



Matrices

- **Multiplications (matrix matrix)**

$$C = AB \text{ where } C_{ik} = \sum_j A_{ij} B_{jk}$$



Matrices

- **Norms**

$$c = A \cdot b \text{ hence } \|c\| \leq \|A\| \cdot \|b\|$$

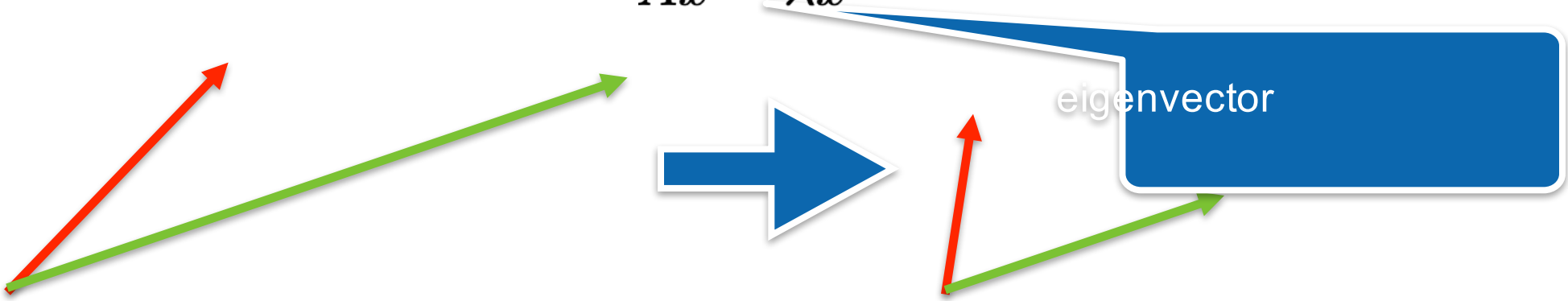
  - Choices depending on how to measure length of b and c
- **A Popular norm**
  - Frobenius norm

$$\|A\|_{\text{Frob}} = \left[\sum_{ij} A_{ij}^2 \right]^{\frac{1}{2}}$$

Matrices

- **Eigenvectors and eigenvalue**
  - Vectors that aren't changed by the matrix

$$Ax = \lambda x$$

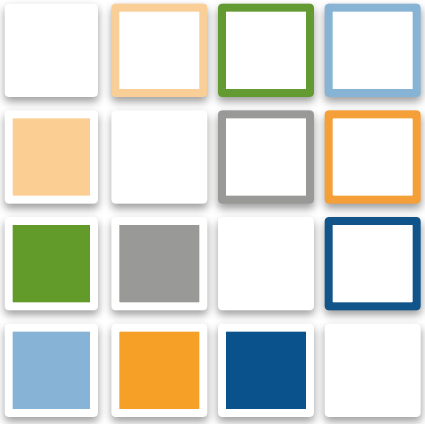


  - For symmetric matrices we can always find this

Special Matrices

- **Symmetric, antisymmetric** $A_{ij} = A_{ji}$ and $A_{ij} = -A_{ji}$





- **Non-negative definite**

$$\|x\|^2 = x^\top x \geq 0 \text{ generalizes to } x^\top A x \geq 0$$

(all non-negative eigenvalues)

Special Matrices

- **Orthogonal Matrices**
  - All rows of the matrix are orthogonal to each other
  - All rows of the matrix have unit length

$$U \text{ with } \sum_j U_{ij} U_{kj} = \delta_{ik}$$

Show that $U^\top U = \mathbf{1}$

  - Rewrite in matrix form

$$UU^\top = \mathbf{1}$$

- **Permutation Matrices**

$$P \text{ where } P_{ij} = 1 \text{ if and only if } j = \pi(i)$$

Show that P is orthogonal

Multidimensional Arrays

N-dimensional Array Examples

N-dimensional array, short for ndarray, is the main data structure for machine learning and neural networks

0-d (scalar)

```
1.0
```

A class label

1-d (vector)

```
[1.0, 2.7, 3.4]
```

A feature vector

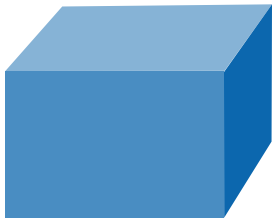
2-d (matrix)

```
[[1.0, 2.7, 3.4]
 [5.0, 0.2, 4.6]
 [4.3, 8.5, 0.2]]
```

A example-by-feature matrix

ND Array Examples, cont

3-d



```
[[[0.1, 2.7, 3.4]
  [5.0, 0.2, 4.6]
  [4.3, 8.5, 0.2]]
 [[3.2, 5.7, 3.4]
  [5.4, 6.2, 3.2]
  [4.1, 3.5, 6.2]]]
```

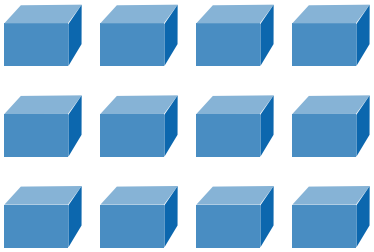
A RGB image (width x height x channels)

4-d

```
[[[[. . .
    . . .
    . . .]]]
```

A batch of RGB images (batch-size x width x height x channels)

5-d

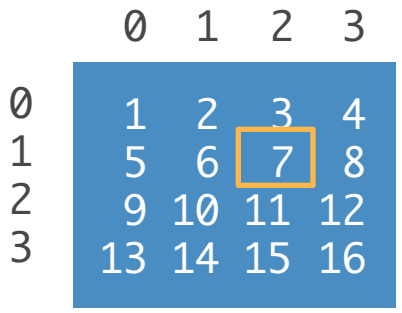


```
[[[[. . .
    . . .
    . . .]]]
```

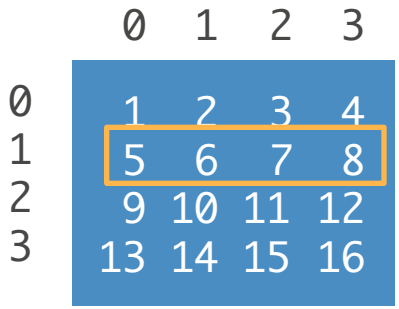
A batch of videos (batch-size x time x width x height x channels)

Access Elements

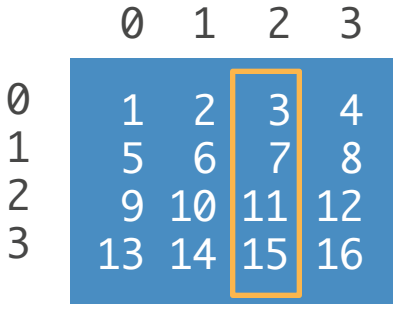
An element: [1, 2]

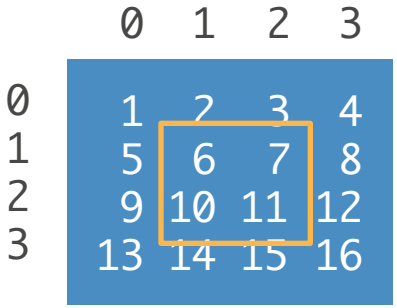


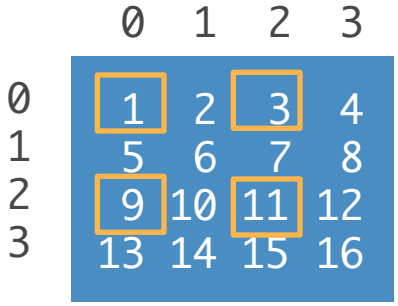
A row: [1, :]



A column: [1, :]







Review of Basic Probability

Probability

Space of events X

- server working; slow response; server broken
- income of the user (e.g. $95,000)
- query text for search (e.g. “statistics tutorial”)

Probability axioms

$$\Pr(X) \in [0,1],\ \Pr(\mathcal{X}) = 1$$
$$\Pr(\cup_i X_i) = \textstyle\sum_i \Pr(X_i) \text{ if } X_i \cap X_j = \emptyset$$

Example queries

- P(server working) = 0.999 — discrete
- P(90,000 < income < 100,000) = 0.1 — continuous

What you must know

- Definitions of random variable and random vector
- Conditional probability, Independence, and dependence
- Law of Total Probability
- Definition of PMF, PDF, and CDF
- Generation of an arbitrary random variable with a given pdf from the uniform random variable
- PMF and PDF of transforms of random vectors
- Mathematical Expectation
- Variance, Covariance, correlation, etc.
- Multivariable Calculus and Lagrange's multiplier method

(In)dependence

Independence

- Login behavior of two users (approximately)
- Disk crash in different colos (approximately)

$$\Pr(x, y) = \Pr(x) \cdot \Pr(y)$$

independent events

- Emails
- Queries
- News stream / Buzz / Tweets
- IM communication
- Russian Roulette

Everywhere

$$\Pr(x, y) \neq \Pr(x) \cdot \Pr(y)$$

Weak Low of Large Numbers

Let $X_1, X_2, \cdots X_n$ be i.i.d. random variable with mean μ. Then for any $\epsilon > 0$

$$P[\ |(X_1 + X_2 + \cdots + X_n)/n\ - \mu| > \epsilon\ ] \to 0$$

as $n \to \infty$.

Order Statistics

For $X_1, X_2, \ldots, X_n$ iid random variables X_k is the kth smallest X, usually called the kth order statistic.

$X_{(1)}$ is therefore the smallest X and

$$X_{(1)} = \min(X_1, \ldots, X_n)$$

Similarly, $X_{(n)}$ is the largest X and

$$X_{(n)} = \max(X_1, \ldots, X_n)$$

Order Statistics (density of maximum)

For $X_1, X_2, \ldots, X_n$ iid continuous random variables with pdf f and cdf F the density of the maximum is

$$
\begin{aligned}
P(X_{(n)} \in [x, x+\epsilon]) &= P(\text{one of the } X\text{'s} \in [x, x+\epsilon] \text{ and all others} < x) \\
&= \sum_{i=1}^{n} P(X_i \in [x, x+\epsilon] \text{ and all others} < x) \\
&= nP(X_1 \in [x, x+\epsilon] \text{ and all others} < x) \\
&= nP(X_1 \in [x, x+\epsilon])P(\text{all others} < x) \\
&= nP(X_1 \in [x, x+\epsilon])P(X_2 < x) \cdots P(X_n < x) \\
&= nf(x)\epsilon F(x)^{n-1}
\end{aligned}
$$

$$
f_{(n)}(x) = nf(x)F(x)^{n-1}
$$

Order Statistics (density of minimum)

For $X_1, X_2, \ldots, X_n$ iid continuous random variables with pdf f and cdf F the density of the minimum is

$$
\begin{aligned}
P(X_{(1)} \in [x, x+\epsilon]) &= P(\text{one of the } X\text{'s} \in [x, x+\epsilon] \text{ and all others} > x) \\
&= \sum_{i=1}^{n} P(X_i \in [x, x+\epsilon] \text{ and all others} > x) \\
&= nP(X_1 \in [x, x+\epsilon] \text{ and all others} > x) \\
&= nP(X_1 \in [x, x+\epsilon])P(\text{all others} > x) \\
&= nP(X_1 \in [x, x+\epsilon])P(X_2 > x) \cdots P(X_n > x) \\
&= nf(x)\epsilon(1 - F(x))^{n-1}
\end{aligned}
$$

$$
f_{(1)}(x) = nf(x)(1 - F(x))^{n-1}
$$

Order Statistics (density of k-th)

For $X_1, X_2, \ldots, X_n$ iid continuous random variables with pdf f and cdf F the density of the kth order statistic is

$$\begin{aligned}
P(X_{(k)} \in [x, x+\epsilon]) &= P(\text{one of the } X\text{'s} \in [x, x+\epsilon] \text{ and exactly } k-1 \text{ of the others} < x) \\
&= \sum_{i=1}^{n} P(X_i \in [x, x+\epsilon] \text{ and exactly } k-1 \text{ of the others} < x) \\
&= nP(X_1 \in [x, x+\epsilon] \text{ and exactly } k-1 \text{ of the others} < x) \\
&= nP(X_1 \in [x, x+\epsilon])P(\text{exactly } k-1 \text{ of the others} < x) \\
&= nP(X_1 \in [x, x+\epsilon]) \left(\binom{n-1}{k-1} P(X < x)^{k-1} P(X > x)^{n-k} \right)
\end{aligned}$$

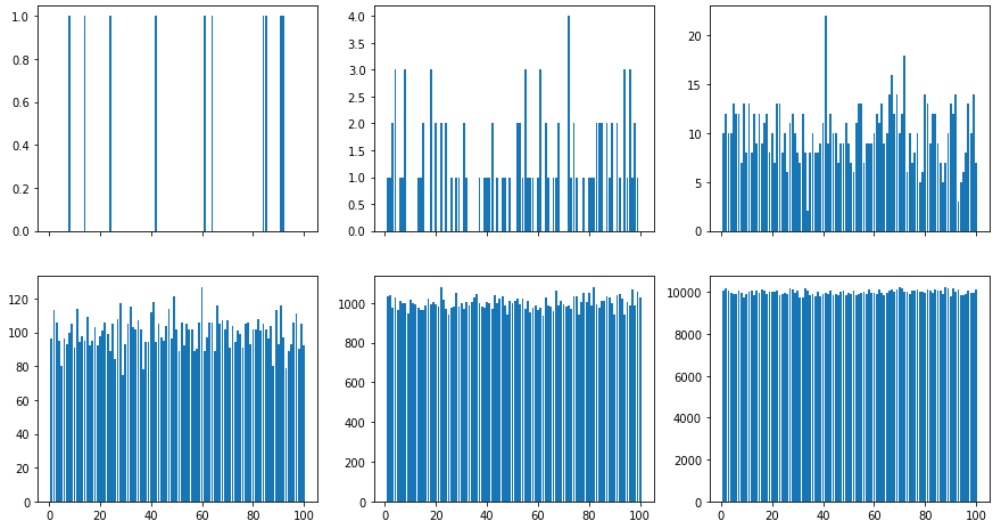
$$f_{(k)}(x) = nf(x) \binom{n-1}{k-1} F(x)^{k-1} (1 - F(x))^{n-k}$$

Uniform Distribution

- Constant within an interval, zero outside

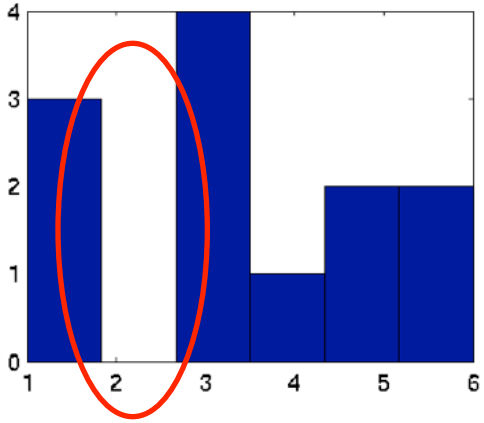
$$p(x) = \frac{1}{U - L} \text{ if } L \leq x \leq U$$

- Useful for initializing parameters or for load distribution

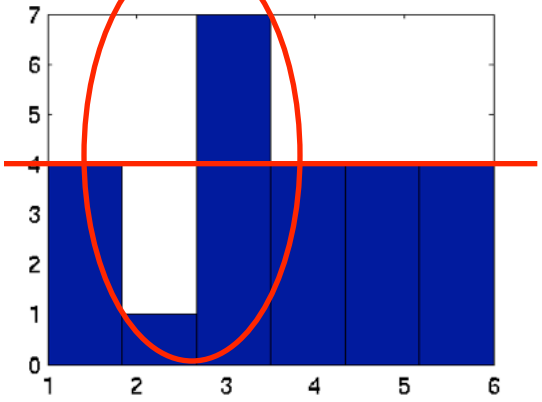


Tossing a Fair Dice

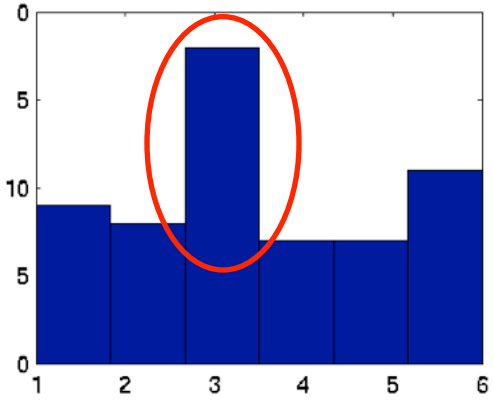
12



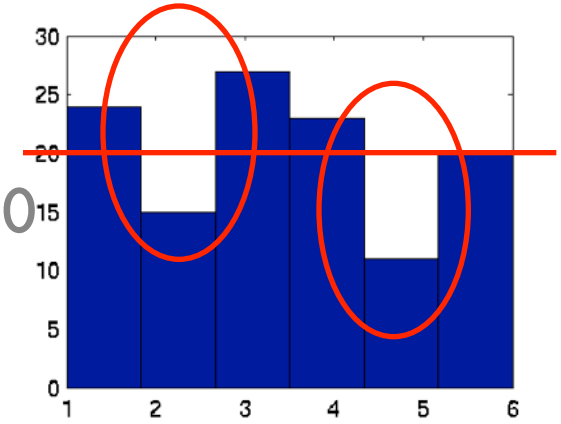
24



60



120



Euler's Gamma Function

For $x > 0$ The Euler's gamma function is defined as:

$$\Gamma(x) \equiv \int_0^\infty u^{x-1} e^{-u} du.$$

- The value $\Gamma(n) = (n-1)!$ When n > 0 is a positive integer.

- Show that $\Gamma(x) = (x-1)\Gamma(x-1)$ for $x > 1$.
- Calculate $\Gamma(3/2)$
- Calculate $\Gamma(1/2)$

Beta Distribution

- We define a distribution for a parameter $\mu \in [0, 1]$

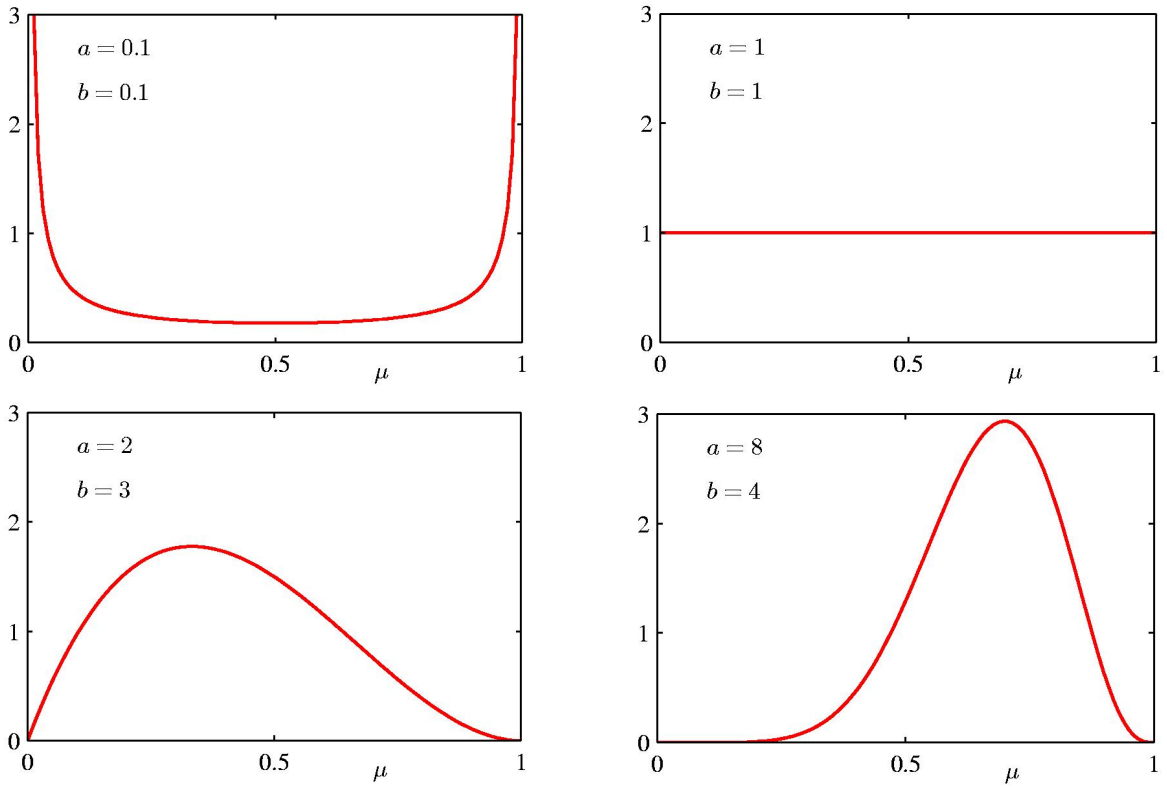
$$\begin{aligned} \text{Beta}(\mu|a, b) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \\ \mathbb{E}[\mu] &= \frac{a}{a+b} \\ \text{var}[\mu] &= \frac{ab}{(a+b)^2(a+b+1)} \end{aligned}$$

where the Euler's gamma function is defined as:

$$\Gamma(x) \equiv \int_0^\infty u^{x-1}e^{-u}du.$$

and ensures that the Beta distribution is normalized.

Beta Distribution



Relationship Between Beta and Uniform

Let $X_1, X_2, \ldots, X_n \overset{iid}{\sim} \text{Unif}(0,1)$ then the density of $X_{(n)}$ is given by

$$f_{(k)}(x) = nf(x)\binom{n-1}{k-1}F(x)^{k-1}(1-F(x))^{n-k}$$

$$= \begin{cases} n\binom{n-1}{k-1}x^{k-1}(1-x)^{n-k} & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

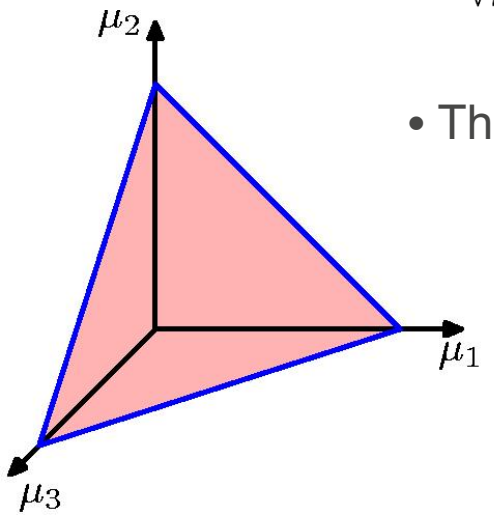
This is an example of the Beta distribution where $r = k$ and $s = n - k + 1$.

$$X_{(k)} \sim \text{Beta}(k, n-k+1)$$

Dirichlet Distribution

• Consider a distribution over the K-dimensional simplex, subject to constraints:

$$\forall k : \mu_k \geqslant 0 \quad \text{and} \quad \sum_{k=1}^{K} \mu_k = 1$$



• The Dirichlet distribution is defined as:

$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$$

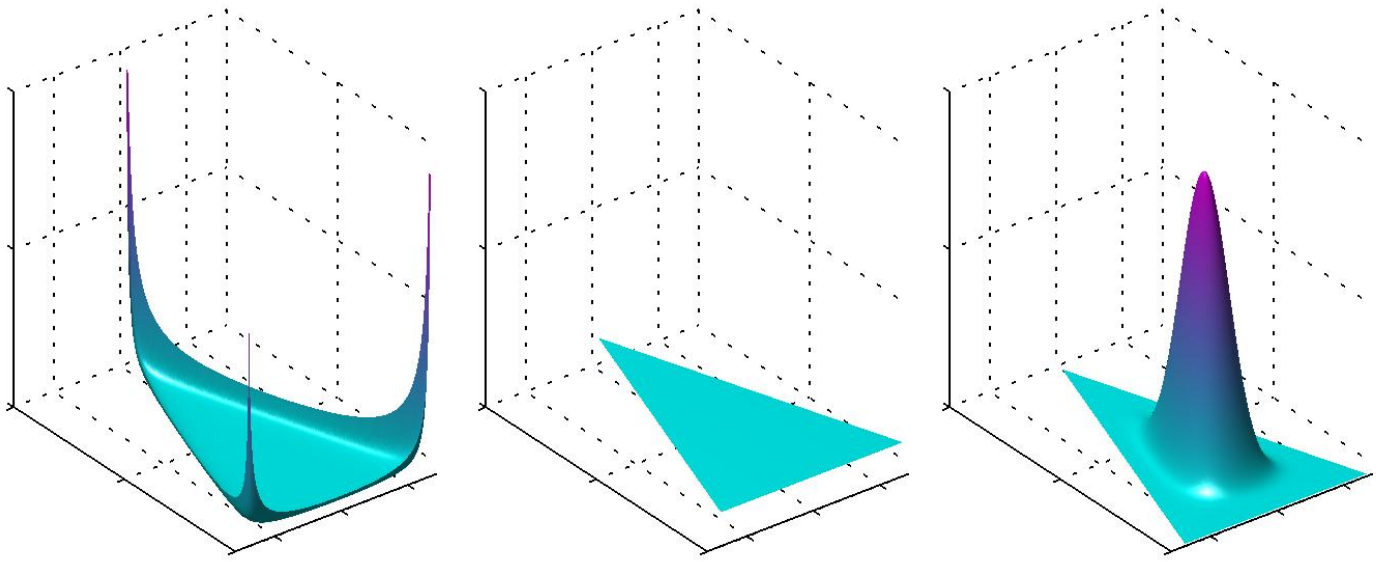
$$\alpha_0 = \sum_{k=1}^{K} \alpha_k$$

where $\alpha_{11},\ldots,\ \alpha_k$ are the parameters of the distribution, and $\Gamma(x)$ is the gamma function.

• The Dirichlet distribution is confined to a simplex as a consequence of the constraints.

Dirichlet Distribution

- Plots of the Dirichlet distribution over three variables.



$\alpha_k = 10^{-1}$ $\alpha_k = 10^{0}$ $\alpha_k = 10^{1}$

Generation of Dirichlet from Beta

- Consider a Stick of length 1.

Simulate a random variate $X_j \sim Beta(\alpha_j, \sum_{i=j+1}^{k} \alpha_i)$, where $j = 1, \ldots, k-1$. When $j = 1$, we have $X_1 \sim Beta(\alpha_1, \sum_{i=2}^{k} \alpha_i)$. The first piece of the stick has length $1 \cdot X_1$, such that the length of the remaining stick is $1 - X_1$. Also, set $Y_1 = X_1$.

Generation of Dirichlet from Beta

When $j = 2$, we have $X_2 \sim Beta(\alpha_2, \sum_{i=3}^{k} \alpha_i)$. The second piece of the stick has length $(1 - X_1)X_2$, such that the length of the remaining stick is $(1 - X_1) - (1 - X_1)X_2 = (1 - X_1)(1 - X_2)$. Also, set $Y_2 = (1 - X_1)X_2$.

:

Continue in this way.

Generation of Dirichlet from Beta

When $j = k-1$, we have $X_{k-1} \sim Beta(\alpha_{k-1}, \alpha_k)$. The $(k-1)^{th}$ piece of the stick has length $X_{k-1} \prod_{j=1}^{k-2}(1-X_j)$, such that the length of the remaining stick is $\prod_{j=1}^{k-1}(1-X_j)$. Also, set $Y_{k-1} = X_{k-1} \prod_{j=1}^{k-2}(1-X_j)$. Note that the k^{th} piece of the stick has length $\prod_{j=1}^{k-1}(1-X_j)$ and set $Y_k = \prod_{j=1}^{k-1}(1-X_j)$. We can conclude that $(Y_1, \ldots, Y_k) \sim Dir(\alpha_1, \ldots, \alpha_k)$.

Source: Bela A Frigyik, Amol Kapila, and Maya R Gupta. Introduction to the Dirichlet distribution and related processes. Technical report, UWEETR-2010-0006, 2010

Entropy

The entropy of a d-dimensional random vector $\mathbf{X} := \begin{bmatrix} X_1 & \cdots & X_d \end{bmatrix}^T$ is defined by the expectation of the self information

$$H(\mathbf{X}) := \mathbb{E}_{\mathbf{X}}\left[\log \frac{1}{p(\mathbf{X})}\right] = \sum_{\mathbf{x}\in\mathcal{X}_1\times\cdots\times\mathcal{X}_d} p(\mathbf{x}) \log \frac{1}{p(\mathbf{x})} = H(X_1, \ldots, X_d)\,.$$

The conditional entropy of X given Y is defined by

$$H(X|\,Y) := \sum_{y\in\mathcal{Y}} p(y) H(X|\,Y = y) = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p\,(x, y) \log \frac{1}{p_{X|\,Y}(x|y)}\,.$$

Kullback-Leibler Divergence

Let $p(\cdot)$ and $q(\cdot)$ are two p.m.f.'s of a random variable X. The relative entropy between p and q is $D(p||q) := \mathbb{E}_p\left[\log \frac{p(X)}{q(X)}\right]$

(the subscript "p" denotes that the expectation is taken over the distribution p.)

Please note that KL divergence is NOT symmetric: D (p || q) ≠ D (q || p).

Important Results:

$D(p||q) \geq 0$, *with equality iff* $p(x) = q(x)$ *for all* $x \in \mathcal{X}$.

Maximum Likelihood Estimator (MLE)

- Let $p_*(\cdot)$ be the true data generating distribution
- $E_*(\cdot)$ be expectation w.r.t. $p_*(\cdot)$
  - Suppose that iid samples (observations}

$$y_1, y_2, \cdots, y_n$$

are given.

  - Let $p \equiv p_\theta$ for $\theta \in \Theta$ denote our guesses for $p_*(\cdot)$
  - MLE: Choose the value of $\theta \in \Theta$ that achieves the maximum of

$$\frac{\Sigma_1^n \log p(y_i)}{n}$$

Maximum Likelihood Estimator (MLE)

- Why is it so popular?
  - It is an elementary function of the probability density function
  - Intimate relation with KL-divergence
  - Notice that

$$-\frac{\Sigma_1^n \log p(y_i)}{n} \rightarrow E_{p^*}[-\log\ p(y)]$$

  - Minimizing

$$-\frac{\Sigma_1^n \log p(y_i)}{n}$$

is asymptotically equivalent to minimizing

$$E_*\{-\log p(y)\} = D_{KL}(p_*||p) + \mathrm{H}(p_*)$$

or equivalently

$$D_{KL}(p_*||p).$$

Bernoulli Distribution

• Consider a single binary random variable $x \in \{0, 1\}$.

•For example, x can describe the outcome of flipping a coin:

Coin flipping: heads = 1, tails = 0.

• The probability of x=1 will be denoted by the parameter [1], so that:

$$p(x = 1|\mu) = \mu \qquad 0 \leq \mu \leq 1.$$

• The probability distribution, known as Bernoulli distribution, can be written as:

$$\begin{aligned} \text{Bern}(x|\mu) &= \mu^x(1-\mu)^{1-x} \\ \mathbb{E}[x] &= \mu \\ \text{var}[x] &= \mu(1-\mu) \end{aligned}$$

Parameter Estimation

- Suppose we observed a data $\mathcal{D} = \{x_1, ..., x_N\}$
- We can construct the likelihood function, which is a function of [1].

$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n}$$

- Equivalently, we can maximize the log of the likelihood function:

$$\ln p(\mathcal{D}|\mu) = \sum_{n=1}^{N} \ln p(x_n|\mu) = \sum_{n=1}^{N} \{x_n \ln \mu + (1-x_n)\ln(1-\mu)\}$$

- Note that the likelihood function depends on the N observations x_n only through the sum

$$\sum_n x_n$$

Sufficient Statistic

Parameter Estimation

- Suppose we observed a data $\mathcal{D} = \{x_1, ..., x_N\}$

$$\ln p(\mathcal{D}|\mu) = \sum_{n=1}^{N} \ln p(x_n|\mu) = \sum_{n=1}^{N} \{x_n \ln \mu + (1 - x_n) \ln(1 - \mu)\}$$

- Setting the derivative of the log-likelihood function w.r.t ^{1} to zero, we obtain:

$$\mu_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} x_n = \frac{m}{N}$$

where m is the number of heads.

Binomial Distribution

- We can also work out the distribution of the number m of observations of x=1 (e.g. the number of heads).

- The probability of observing m heads given N coin flips and a parameter μ is given by:

$$p(m \text{ heads}|N,\mu) =$$

$$\text{Bin}(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m}$$

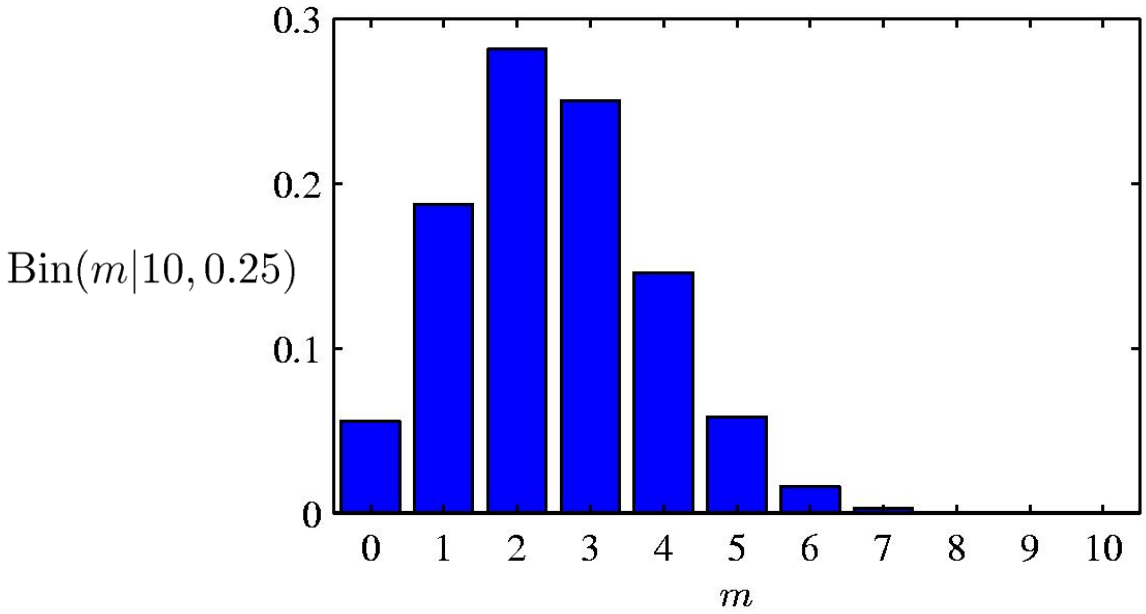
- The mean and variance can be easily derived as:

$$\mathbb{E}[m] \equiv \sum_{m=0}^{N} m\text{Bin}(m|N,\mu) = N\mu$$

$$\text{var}[m] \equiv \sum_{m=0}^{N} (m-\mathbb{E}[m])^2 \text{Bin}(m|N,\mu) = N\mu(1-\mu)$$

Example

• Histogram plot of the Binomial distribution as a function of m for N=10 and μ = 0.25.



Multinomial Variables

- Consider a random variable that can take on one of K possible mutually exclusive states (e.g. roll of a dice).
- We will use so-called 1-of-K encoding scheme.
- If a random variable can take on K=6 states, and a particular observation of the variable corresponds to the state x_3=1, then **x** will be resented as:

1-of-K coding scheme: $\mathbf{x} = (0, 0, 1, 0, 0, 0)^{\mathrm{T}}$

- If we denote the probability of x_k=1 by the parameter μ_k, then the distribution over **x** is defined as:

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \qquad \forall k : \mu_k \geqslant 0 \quad \text{and} \quad \sum_{k=1}^{K} \mu_k = 1$$

Multinomial Variables

• Multinomial distribution can be viewed as a generalization of Bernoulli distribution to more than two outcomes.

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$$

• It is easy to see that the distribution is normalized:

$$\sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu}) = \sum_{k=1}^{K} \mu_k = 1$$

and

$$\mathbb{E}[\mathbf{x}|\boldsymbol{\mu}] = \sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu})\mathbf{x} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}} = \boldsymbol{\mu}$$

Maximum Likelihood Estimation

- Suppose we observed a data $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$
- We can construct the likelihood function, which is a function of [1].

$$p(\mathcal{D}|\boldsymbol{\mu}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mu_k^{x_{nk}} = \prod_{k=1}^{K} \mu_k^{\left(\sum_n x_{nk}\right)} = \prod_{k=1}^{K} \mu_k^{m_k}$$

- Note that the likelihood function depends on the N data points only though the following K quantities:

$$m_k = \sum_n x_{nk}, \quad k = 1, ..., K.$$

which represents the number of observations of x_k=1.

- These are called the sufficient statistics for this distribution.

Maximum Likelihood Estimation

$$p(\mathcal{D}|\boldsymbol{\mu}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mu_k^{x_{nk}} = \prod_{k=1}^{K}\mu_k^{\left(\sum_n x_{nk}\right)} = \prod_{k=1}^{K}\mu_k^{m_k}$$

- To find a maximum likelihood solution for ^{1}, we need to maximize the log-likelihood taking into account the constraint that $\sum_k \mu_k = 1$

  - Forming the Lagrangian:

$$\sum_{k=1}^{K} m_k \ln \mu_k + \lambda\left(\sum_{k=1}^{K}\mu_k - 1\right)$$

$$\mu_k = -m_k/\lambda \qquad \mu_k^{\mathrm{ML}} = \frac{m_k}{N} \qquad \lambda = -N$$

which is the fraction of observations for which x_k=1.

Multinomial Distribution

• We can construct the joint distribution of the quantities $\{m_1, m_2, \ldots, m_k\}$ given the parameters [1] and the total number N of observations:

$$\begin{aligned}
\mathrm{Mult}(m_1, m_2, \ldots, m_K | \boldsymbol{\mu}, N) &= \binom{N}{m_1 m_2 \ldots m_K} \prod_{k=1}^{K} \mu_k^{m_k} \\
\mathbb{E}[m_k] &= N\mu_k \\
\mathrm{var}[m_k] &= N\mu_k(1-\mu_k) \\
\mathrm{cov}[m_j m_k] &= -N\mu_j\mu_k
\end{aligned}$$

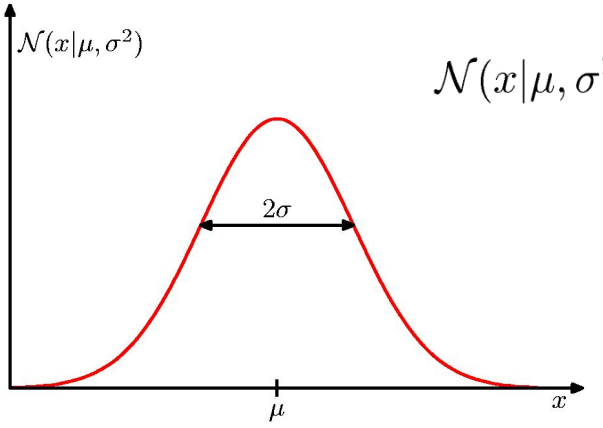
• The normalization coefficient is the number of ways of partitioning N objects into K groups of size $m_1, m_2, \ldots, m_K$.

• Note that

$$\sum_k m_k = N.$$

Gaussian Univariate Distribution

• In the case of a single variable x, the Gaussian distribution takes form:



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

which is governed by two parameters:

- μ (mean)
- σ^2 (variance)

- The Gaussian distribution satisfies:

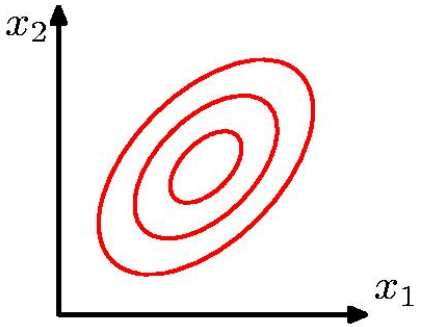
$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

Multivariate Gaussian Distribution

• For a D-dimensional vector **x**, the Gaussian distribution takes form:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$



which is governed by two parameters:

- μ is a D-dimensional mean vector.
- Σ is a D by D covariance matrix.

and $|\Sigma|$ denotes the determinant of Σ.

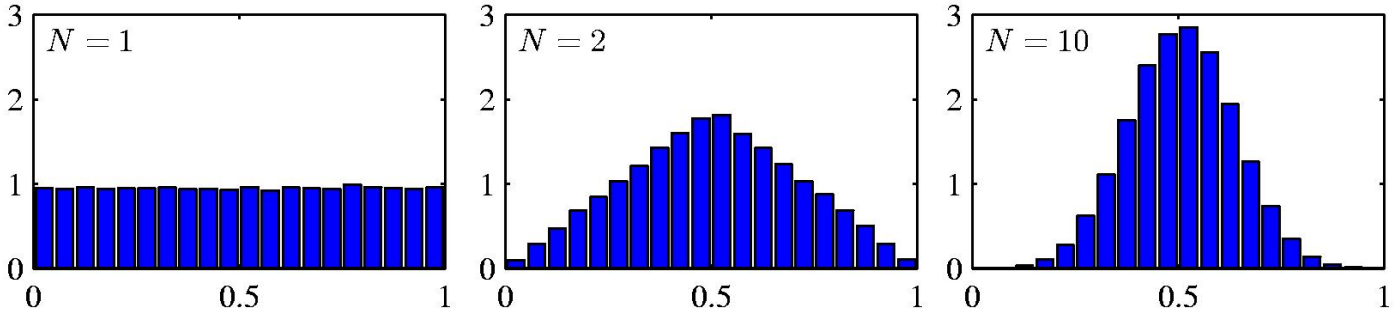
- Note that the covariance matrix is a symmetric positive definite matrix.

Central Limit Theorem

- The distribution of the sum of N i.i.d. random variables becomes increasingly Gaussian as N grows.
- Consider N variables, each of which has a uniform distribution over the interval [0,1].
- Let us look at the distribution over the mean:

$$\frac{x_1 + x_2 + ... + x_N}{N}.$$

- As N increases, the distribution tends towards a Gaussian distribution.



Moments of the Gaussian Distribution

- The expectation of **x** under the Gaussian distribution:

$$\begin{aligned}\mathbb{E}[\mathbf{x}] &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}\mathbf{x}\,\mathrm{d}\mathbf{x} \\ &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}\mathbf{z}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}\right\}(\mathbf{z}+\boldsymbol{\mu})\,\mathrm{d}\mathbf{z}\end{aligned}$$

The term in z in the factor $(z+^{1})$ will vanish by symmetry.

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

Moments of the Gaussian Distribution

- The second order moments of the Gaussian distribution:

$$\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{\Sigma}$$

- The covariance is given by:

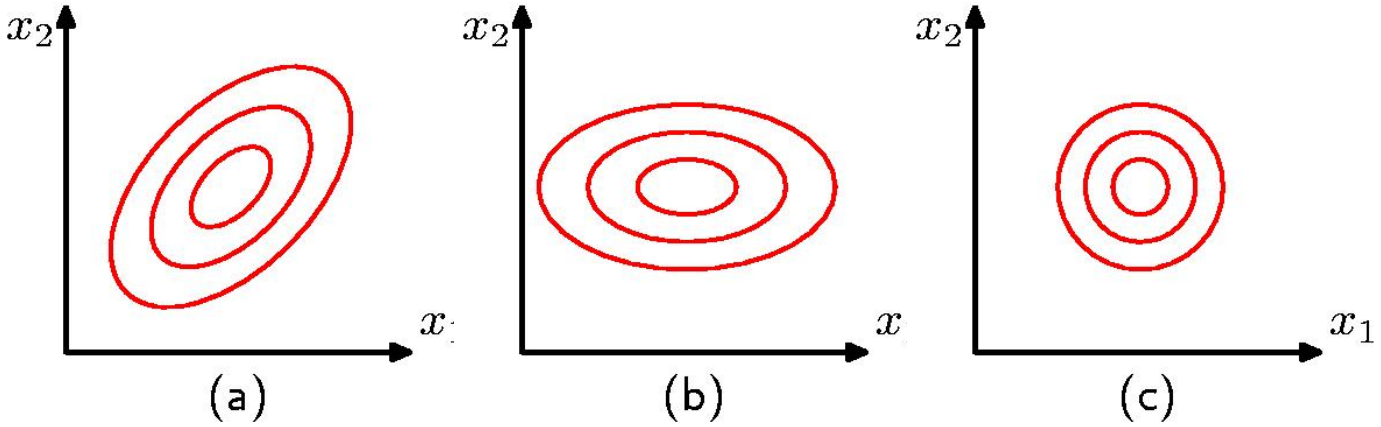
$$\mathrm{cov}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\mathrm{T}}\right] = \boldsymbol{\Sigma}$$

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

- Because the parameter matrix § governs the covariance of x under the Gaussian distribution, it is called the covariance matrix.

Moments of the Gaussian Distribution

- Contours of constant probability density:



Covariance matrix is of general form.

Diagonal, axis-aligned covariance matrix.

Spherical (proportional to identity) covariance matrix.

Partitioned Gaussian Distribution

- Consider a D-dimensional Gaussian distribution: $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Let us partition **x** into two disjoint subsets x_a and x_b:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

- In many situations, it will be more convenient to work with the precision matrix (inverse of the covariance matrix):

$$\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1} \qquad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}$$

- Note that Λ_{aa} is not given by the inverse of Σ_{aa}.

Marginal Distribution

• It turns out that the marginal distribution is also a Gaussian distribution:

$$\begin{aligned} p(\mathbf{x}_a) &= \int p(\mathbf{x}_a, \mathbf{x}_b)\, \mathrm{d}\mathbf{x}_b \\ &= \mathcal{N}(\mathbf{x}_a | \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}) \end{aligned}$$

• For a marginal distribution, the mean and covariance are most simply expressed in terms of partitioned covariance matrix.

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

Conditional Distribution

- It turns out that the conditional distribution is also a Gaussian distribution:

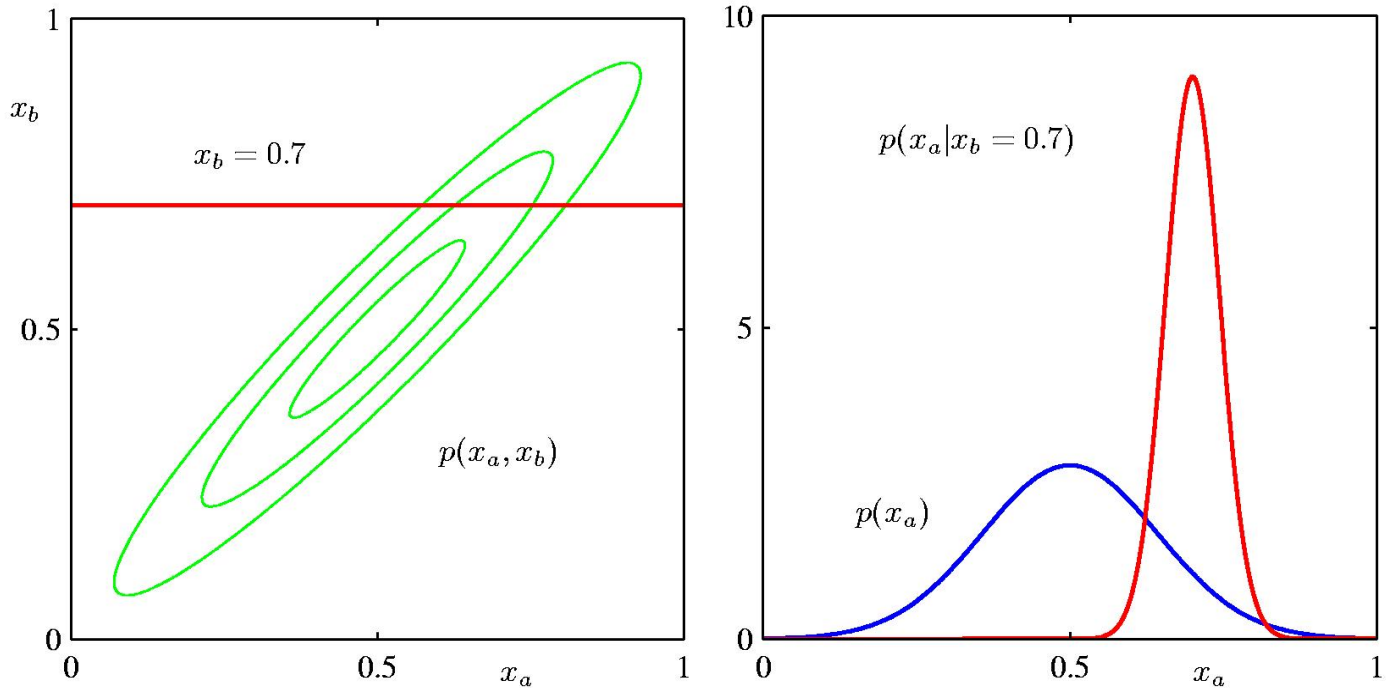
$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$$

Covariance does not depend on x_b.

$$\begin{aligned}
\boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Lambda}_{aa}^{-1} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \\
\boldsymbol{\mu}_{a|b} &= \boldsymbol{\Sigma}_{a|b}\left\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\right\} \\
&= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\
&= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b)
\end{aligned}$$

Linear function of x_b.

Conditional and Marginal Distributions



Maximum Likelihood Estimation

• To find a maximum likelihood estimate of the mean, we set the derivative of the log-likelihood function to zero:

$$\frac{\partial}{\partial \boldsymbol{\mu}} \ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}) = 0$$

and solve to obtain:

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n.$$

• Similarly, we can find the ML estimate of Σ :

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}}.$$

Maximum Likelihood Estimation

• Evaluating the expectation of the ML estimates under the true distribution, we obtain:

Unbiased estimate

$$\mathbb{E}[\boldsymbol{\mu}_{\mathrm{ML}}] = \boldsymbol{\mu}$$

$$\mathbb{E}[\boldsymbol{\Sigma}_{\mathrm{ML}}] = \frac{N-1}{N}\boldsymbol{\Sigma}.$$

Biased estimate

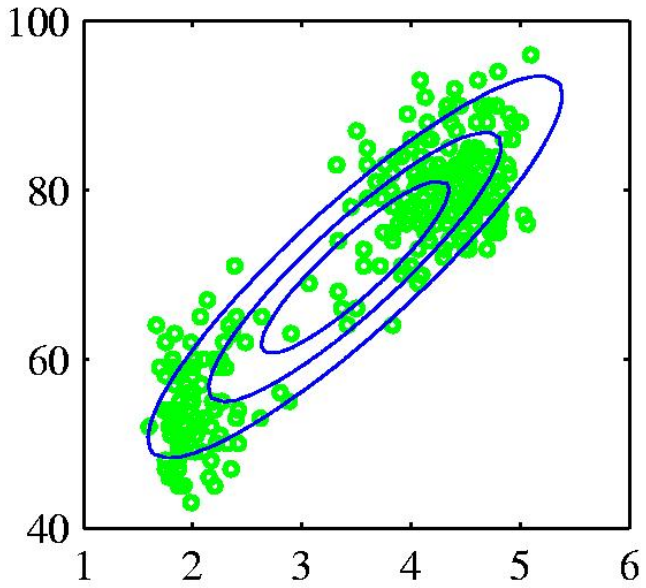
• Note that the maximum likelihood estimate of § is biased.

• We can correct the bias by defining a different estimator:

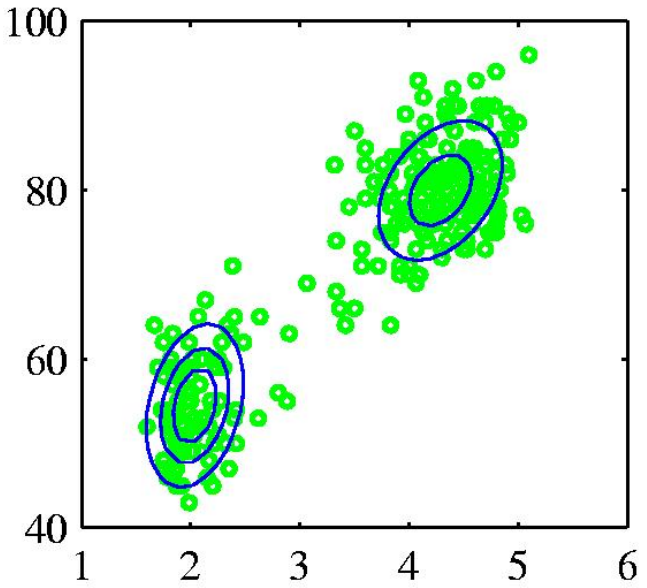
$$\widetilde{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}}.$$

Mixture of Gaussians

- When modeling real-world data, Gaussian assumption may not be appropriate.
- Consider the following example: Old Faithful Dataset



Single Gaussian



Mixture of two Gaussians

Mixture of Gaussians

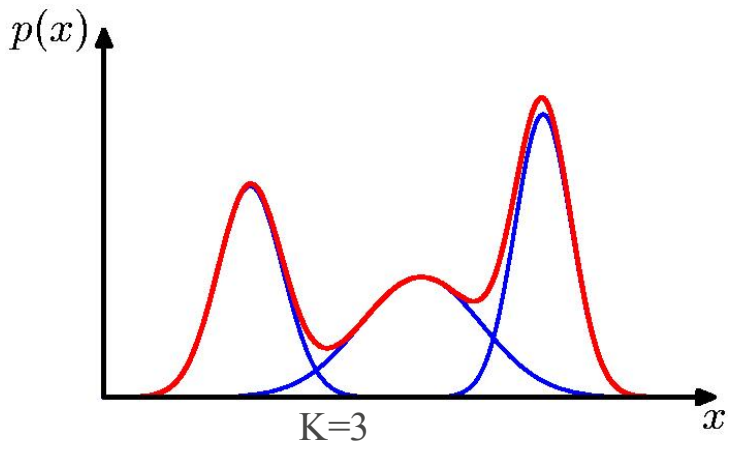
• We can combine simple models into a complex model by defining a superposition of K Gaussian densities of the form:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Component

Mixing coefficient

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

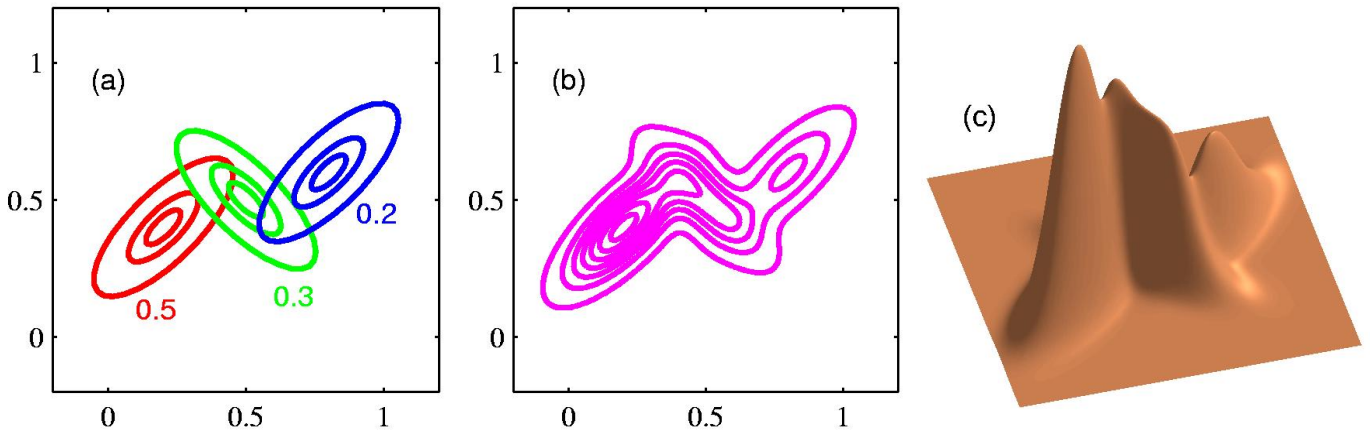


• Note that each Gaussian component has its own mean μ_k and covariance Σ_k. The parameters π_k are called mixing coefficients.

• More generally, mixture models can comprise linear combinations of other distributions.

Mixture of Gaussians

- Illustration of a mixture of 3 Gaussians in a 2-dimensional space:



(a) Contours of constant density of each of the mixture components, along with the mixing coefficients

(b) Contours of marginal probability density

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

(c) A surface plot of the distribution p(x).

Maximum Likelihood Estimation

• Given a dataset D, we can determine model parameters by maximizing the log-likelihood function:

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Log of a sum: no closed form solution

• **Solution**: use standard, iterative, numeric optimization methods or the Expectation Maximization algorithm.

The Exponential Distribution

The family of exponential distribution provides probability models that are very widely used.

Definition

X is said to have an **exponential distribution** with parameter $\lambda > 0$ if the pdf of X is

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

The Gamma Distribution

Definition

A continuous random variable X is said to have a **gamma distribution** if the pdf of X is

$$f(x;\alpha,\beta) = \begin{cases} \dfrac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where the parameters α and β satisfy $\alpha > 0$, $\beta > 0$. The **standard gamma distribution** has $\beta = 1$.

We may be lazy and write Gam(x| α , β) for the above pdf instead.

Generation of Dirichlet from Gamma

- Take $Y_1 \sim \mathrm{Gam}(\mathrm{x}|\ \alpha_1\ , \beta)$, $Y_2 \sim \mathrm{Gam}(\mathrm{x}|\ \alpha_2\ , \beta)$, $\cdots\ \ Y_n \sim \mathrm{Gam}(\mathrm{x}|\ \alpha_n\ , \beta)$,
- Let $V = \sum_1^n Y_i$.
- Let

$$X_i = \frac{Y_i}{V}$$

- Then $(X_1, \cdots, X_n)$ are distributed according to Dirichlet with parameters $\alpha_1, \cdots\ \alpha_n$.

Student's t-Distribution

- Consider Student's t-Distribution

$$
\begin{aligned}
p(x|\mu,a,b) &= \int_0^\infty \mathcal{N}(x|\mu,\tau^{-1})\mathrm{Gam}(\tau|a,b)\,\mathrm{d}\tau \\
&= \int_0^\infty \mathcal{N}\left(x|\mu,(\eta\lambda)^{-1}\right)\mathrm{Gam}(\eta|\nu/2,\nu/2)\,\mathrm{d}\eta \\
&= \frac{\Gamma(\nu/2+1/2)}{\Gamma(\nu/2)}\left(\frac{\lambda}{\pi\nu}\right)^{1/2}\left[1+\frac{\lambda(x-\mu)^2}{\nu}\right]^{-\nu/2-1/2} \\
&= \mathrm{St}(x|\mu,\lambda,\nu)
\end{aligned}
$$

Infinite mixture of Gaussians

where

$$\lambda = a/b \qquad \eta = \tau b/a \qquad \nu = 2a.$$

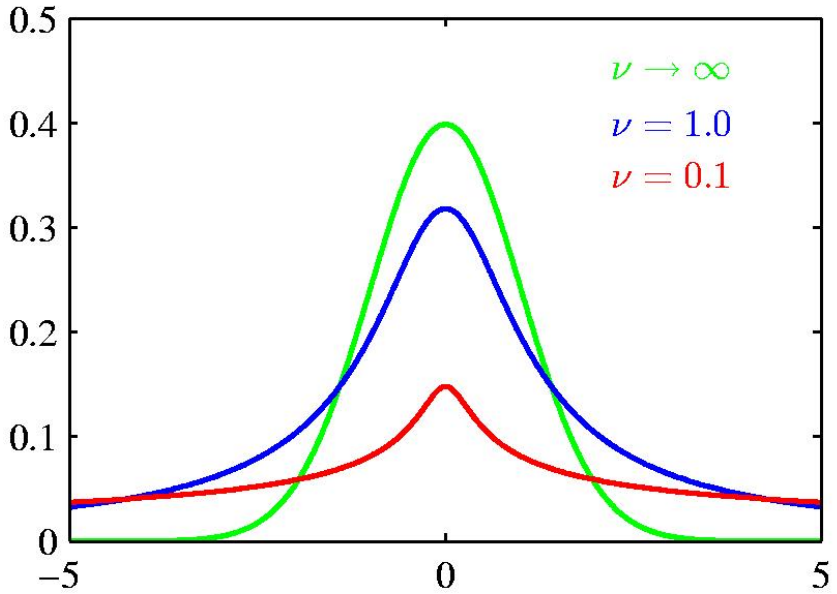
Sometimes called the precision parameter.

Degrees of freedom

Student's t-Distribution

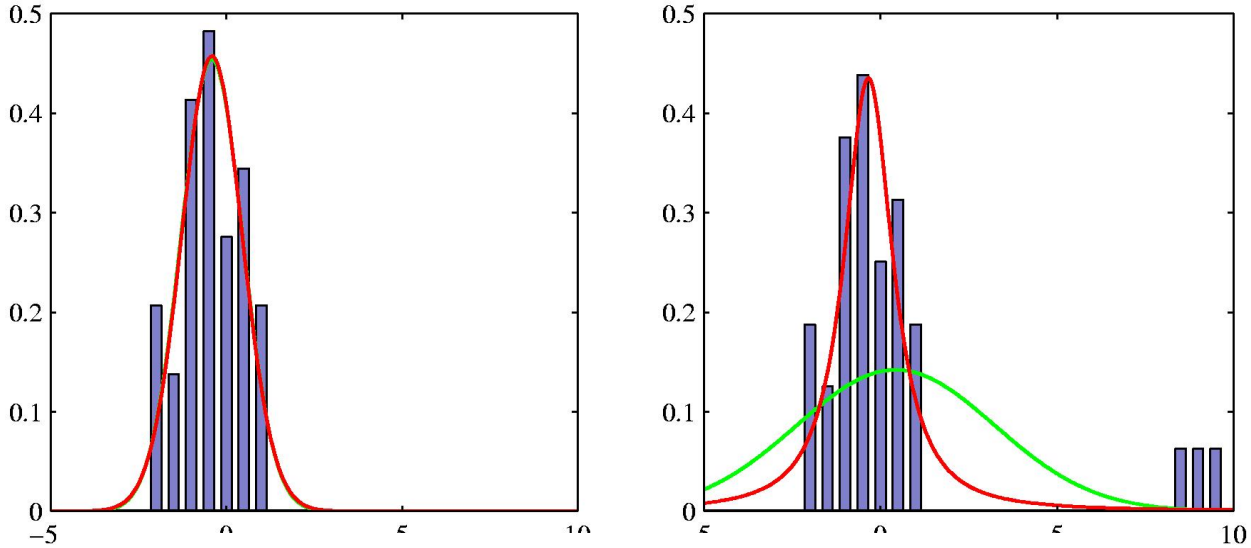
- Setting $\nu = 1$ recovers Cauchy distribution
- The limit $\nu \to \infty$ corresponds to a Gaussian distribution.

	$\nu = 1$	$\nu \to \infty$
$\mathrm{St}(x\|\mu, \lambda, \nu)$	Cauchy	$\mathcal{N}(x\|\mu, \lambda^{-1})$



Student's t-Distribution

- Robustness to outliners: Gaussian vs. t-Distribution.



Student's t-Distribution

- The multivariate extension of the t-Distribution of dimension D:

$$\begin{aligned}\mathrm{St}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Lambda},\nu) &= \int_0^\infty \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},(\eta\boldsymbol{\Lambda})^{-1})\mathrm{Gam}(\eta|\nu/2,\nu/2)\,\mathrm{d}\eta \\ &= \frac{\Gamma(D/2+\nu/2)}{\Gamma(\nu/2)}\frac{|\boldsymbol{\Lambda}|^{1/2}}{(\pi\nu)^{D/2}}\left[1+\frac{\Delta^2}{\nu}\right]^{-D/2-\nu/2}\end{aligned}$$

where $\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Lambda}(\mathbf{x}-\boldsymbol{\mu})$

- Properties:

$$\begin{aligned}\mathbb{E}[\mathbf{x}] &= \boldsymbol{\mu}, && \text{if } \nu > 1 \\ \mathrm{cov}[\mathbf{x}] &= \frac{\nu}{(\nu-2)}\boldsymbol{\Lambda}^{-1}, && \text{if } \nu > 2 \\ \mathrm{mode}[\mathbf{x}] &= \boldsymbol{\mu}\end{aligned}$$

The Exponential Family

• The exponential family of distributions over **x** is defined to be a set of distributions of the form:

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

where

- $\boldsymbol{\eta}$ is the vector of natural parameters
- $\mathbf{u}(\mathbf{x})$ is the vector of sufficient statistics

• The function g($\boldsymbol{\eta}$) can be iinterpreted as the coefficient that ensures that the distribution p($\mathbf{x}|\boldsymbol{\eta}$) is normalized:

$$g(\boldsymbol{\eta})\int h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}\,\mathrm{d}\mathbf{x} = 1$$

Bernoulli Distribution

- The Bernoulli distribution is a member of the exponential family:

$$\begin{aligned} p(x|\mu) &= \text{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x} \\ &= \exp\{x\ln\mu + (1-x)\ln(1-\mu)\} \\ &= (1-\mu)\exp\left\{\ln\left(\frac{\mu}{1-\mu}\right)x\right\} \end{aligned}$$

- Comparing with the general form of the exponential family:

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

we see that

$$\eta = \ln\left(\frac{\mu}{1-\mu}\right) \qquad \text{and so} \qquad \underbrace{\mu = \sigma(\eta) = \frac{1}{1+\exp(-\eta)}.}_{\text{Logistic sigmoid}}$$

Bernoulli Distribution

- The Bernoulli distribution is a member of the exponential family:

$$\begin{aligned} p(x|\mu) &= \text{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x} \\ &= \exp\{x\ln\mu + (1-x)\ln(1-\mu)\} \\ &= (1-\mu)\exp\left\{\ln\left(\frac{\mu}{1-\mu}\right)x\right\} \end{aligned}$$

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

- The Bernoulli distribution can therefore be written as:

$$p(x|\eta) = \sigma(-\eta)\exp(\eta x)$$

where

$$\begin{aligned} u(x) &= x \\ h(x) &= 1 \\ g(\eta) &= 1 - \sigma(\eta) = \sigma(-\eta). \end{aligned}$$

Multinomial Distribution

• The Multinomial distribution is a member of the exponential family:

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{M} \mu_k^{x_k} = \exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} = h(\mathbf{x}) g(\boldsymbol{\eta}) \exp\left(\boldsymbol{\eta}^{\mathrm{T}} \mathbf{u}(\mathbf{x})\right)$$

where $\mathbf{x} = (x_1, \ldots, x_M)^{\mathrm{T}}$ $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_M)^{\mathrm{T}}$

and

$$\begin{aligned} \eta_k &= \ln \mu_k \\ \mathbf{u}(\mathbf{x}) &= \mathbf{x} \\ h(\mathbf{x}) &= 1 \\ g(\boldsymbol{\eta}) &= 1. \end{aligned}$$

NOTE: The parameters μ_k are not independent since the corresponding μ_k must satisfy

$$\sum_{k=1}^{M} \mu_k = 1.$$

• In some cases it will be convenient to remove the constraint by expressing the distribution over the M-1 parameters.

Multinomial Distribution

- The Multinomial distribution is a member of the exponential family:

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{M} \mu_k^{x_k} = \exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} = h(\mathbf{x}) g(\boldsymbol{\eta}) \exp\left(\boldsymbol{\eta}^{\mathrm{T}} \mathbf{u}(\mathbf{x})\right)$$

- Let $\mu_M = 1 - \sum_{k=1}^{M-1} \mu_k$

- This leads to:

$$\eta_k = \ln\left(\frac{\mu_k}{1 - \sum_{j=1}^{M-1} \mu_j}\right) \quad \text{and} \quad \mu_k = \underbrace{\frac{\exp(\eta_k)}{1 + \sum_{j=1}^{M-1} \exp(\eta_j)}}_{\text{Softmax function}}.$$

- Here the parameters η_k are independent.

- Note that:

$$0 \leqslant \mu_k \leqslant 1 \quad \text{and} \quad \sum_{k=1}^{M-1} \mu_k \leqslant 1.$$

Multinomial Distribution

- The Multinomial distribution is a member of the exponential family:

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{M} \mu_k^{x_k} = \exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} = h(\mathbf{x}) g(\boldsymbol{\eta}) \exp\left(\boldsymbol{\eta}^{\mathrm{T}} \mathbf{u}(\mathbf{x})\right)$$

- The Multinomial distribution can therefore be written as:

$$p(\mathbf{x}|\boldsymbol{\mu}) = h(\mathbf{x}) g(\boldsymbol{\eta}) \exp\left(\boldsymbol{\eta}^{\mathrm{T}} \mathbf{u}(\mathbf{x})\right)$$

where

$$\begin{aligned} \boldsymbol{\eta} &= (\eta_1, \ldots, \eta_{M-1}, 0)^{\mathrm{T}} \\ \mathbf{u}(\mathbf{x}) &= \mathbf{x} \\ h(\mathbf{x}) &= 1 \\ g(\boldsymbol{\eta}) &= \left(1 + \sum_{k=1}^{M-1} \exp(\eta_k)\right)^{-1} . \end{aligned}$$

Gaussian Distribution

- The Gaussian distribution can be written as:

$$
\begin{aligned}
p(x|\mu,\sigma^2) &= \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}x^2+\frac{\mu}{\sigma^2}x-\frac{1}{2\sigma^2}\mu^2\right\} \\
&= h(x)g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(x)\right\}
\end{aligned}
$$

where

$$
\boldsymbol{\eta} = \begin{pmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{pmatrix} \qquad h(\mathbf{x}) = (2\pi)^{-1/2}
$$

$$
\mathbf{u}(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix} \qquad g(\boldsymbol{\eta}) = (-2\eta_2)^{1/2}\exp\left(\frac{\eta_1^2}{4\eta_2}\right).
$$

ML for the Exponential Family

- Recall the Exponential Family:

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

  - From the definition of the normalizer g(´):

$$g(\boldsymbol{\eta})\int h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}\,\mathrm{d}\mathbf{x} = 1$$

  - We can take a derivative w.r.t ´:

$$\nabla g(\boldsymbol{\eta})\underbrace{\int h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}\,\mathrm{d}\mathbf{x}}_{1/g(\boldsymbol{\eta})} + g(\boldsymbol{\eta})\underbrace{\int h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}\mathbf{u}(\mathbf{x})\,\mathrm{d}\mathbf{x}}_{\mathbb{E}[\mathbf{u}(\mathbf{x})]} = 0$$

  - Thus

$$-\nabla \ln g(\boldsymbol{\eta}) = \mathbb{E}[\mathbf{u}(\mathbf{x})]$$

ML for the Exponential Family

- Recall the Exponential Family:

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

  - We can take a derivative w.r.t ´:

$$\nabla g(\boldsymbol{\eta}) \underbrace{\int h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}\,\mathrm{d}\mathbf{x}}_{1/g(\boldsymbol{\eta})} + g(\boldsymbol{\eta}) \underbrace{\int h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}\mathbf{u}(\mathbf{x})\,\mathrm{d}\mathbf{x}}_{\mathbb{E}[\mathbf{u}(\mathbf{x})]} = 0$$

  - Thus

$$-\nabla \ln g(\boldsymbol{\eta}) = \mathbb{E}[\mathbf{u}(\mathbf{x})]$$

  - Note that the covariance of **u**(**x**) can be expressed in terms of the second derivative of g(´), and similarly for the higher moments.

ML for the Exponential Family

- Suppose we observed i.i.d da$\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$.
- We can construct the log-likelihood function, which is a function of the natural parameter ´.

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

$$p(\mathbf{X}|\boldsymbol{\eta}) = \left(\prod_{n=1}^{N} h(\mathbf{x}_n)\right) g(\boldsymbol{\eta})^N \exp\left\{\boldsymbol{\eta}^{\mathrm{T}} \sum_{n=1}^{N} \mathbf{u}(\mathbf{x}_n)\right\}.$$

- Therefore we have

$$-\nabla \ln g(\boldsymbol{\eta}_{\mathrm{ML}}) = \underbrace{\frac{1}{N} \sum_{n=1}^{N} \mathbf{u}(\mathbf{x}_n)}$$

Sufficient Statistic

Review of Multivariable Calculus

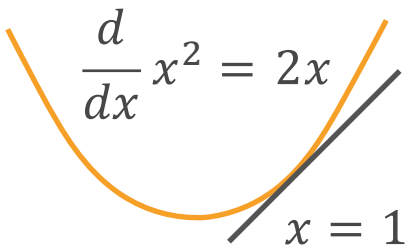
Review Scalar Derivative

y	a	x^n	$\exp(x)$	$\log(x)$	$\sin(x)$
$\frac{dy}{dx}$	0	nx^{n-1}	$\exp(x)$	$\frac{1}{x}$	$\cos(x)$

a is not a function of x

y	$u+v$	uv	$y=f(u), u=g(x)$
$\frac{dy}{dx}$	$\frac{du}{dx}+\frac{dv}{dx}$	$\frac{du}{dx}v+\frac{dv}{dx}u$	$\frac{dy}{du}\frac{du}{dx}$

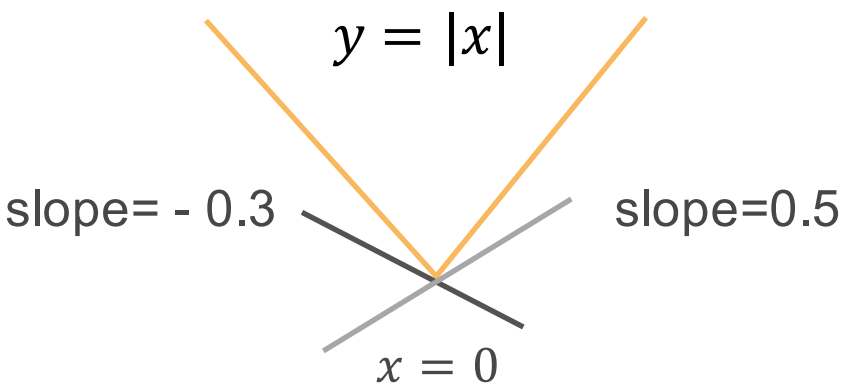
Derivative is the slope of the tangent line



The slope of the tangent line is 2

Subderivative

Extend derivative to non-differentiable cases



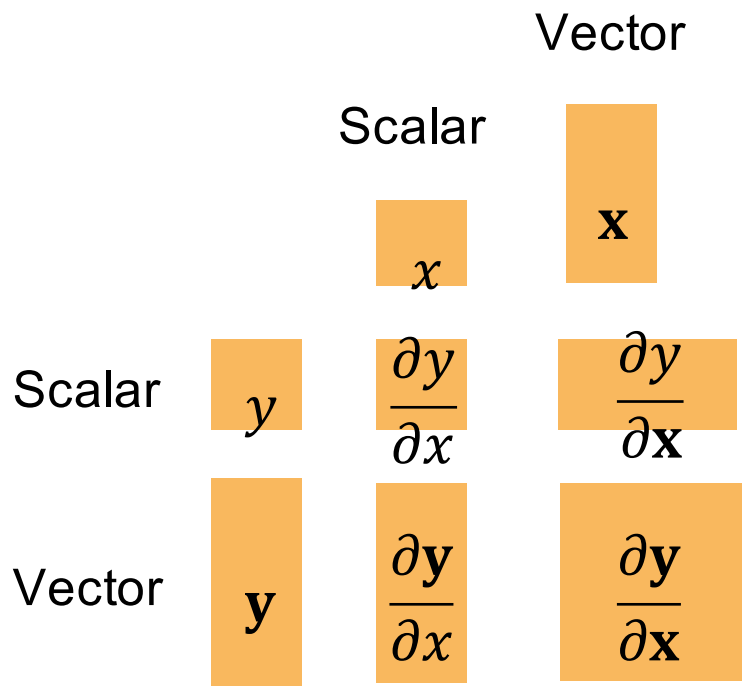
$$\frac{\partial |x|}{\partial x} = \begin{cases} 1 & if\ x > 0 \\ -1 & if\ x < 0 \\ a & if\ x = 0, a \in [-1,1] \end{cases}$$

Another example:

$$\frac{\partial}{\partial x} max(x, 0) = \begin{cases} 1 & if\ x > 0 \\ 0 & if\ x < 0 \\ a & if\ x = 0, a \in [0,1] \end{cases}$$

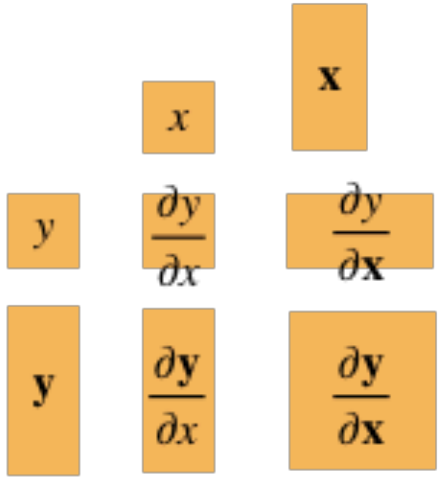
Gradients

Generalize derivatives into vectors



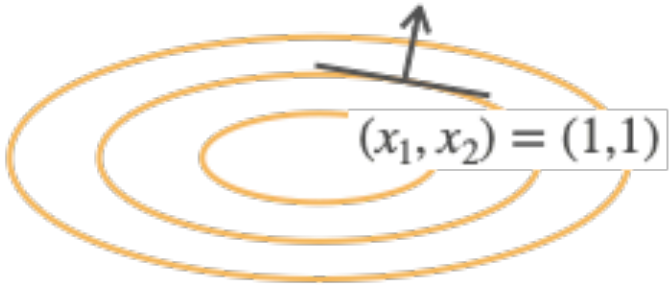
$\partial y / \partial \mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \frac{\partial y}{\partial \mathbf{x}} = \left[\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \cdots, \frac{\partial y}{\partial x_n} \right]$$



$$\frac{\partial}{\partial \mathbf{x}} x_1^2 + 2x_2^2 = [2x_1, 4x_2]$$

Direction (2, 4), perpendicular to the contour lines



Examples

y	a	au	$sum(\mathbf{x})$	$\| \mathbf{x} \|^2$
$\frac{\partial y}{\partial \mathbf{x}}$	$\mathbf{0}^T$	$a\frac{\partial u}{\partial \mathbf{x}}$	$\mathbf{1}^T$	$2\mathbf{x}^T$

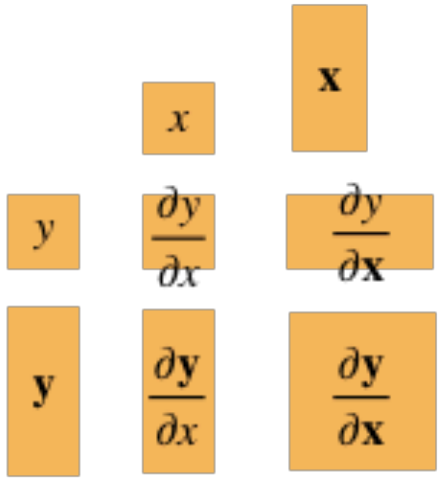
a is not a function of $\mathbf{x}$

$\mathbf{0}$ *and* $\mathbf{1}$ *are vectors*

y	$u+v$	uv	$\langle \mathbf{u}, \mathbf{v} \rangle$
$\frac{\partial y}{\partial \mathbf{x}}$	$\frac{\partial u}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}$	$\frac{\partial u}{\partial \mathbf{x}} v + \frac{\partial v}{\partial \mathbf{x}} u$	$\mathbf{u}^T \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^T \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$

$\partial \mathbf{y} / \partial x$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \qquad \frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \\ \vdots \\ \frac{\partial y_m}{\partial x} \end{bmatrix}$$



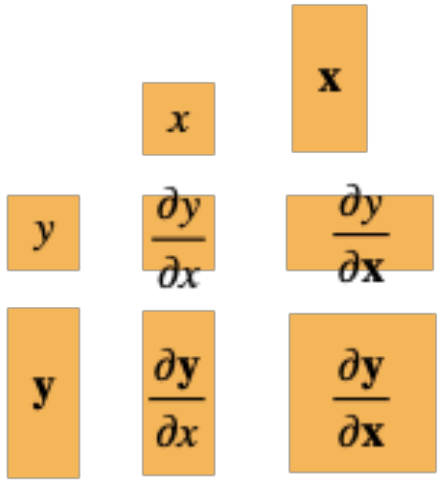
$\partial y / \partial \mathbf{x}$ is a row vector, while $\partial \mathbf{y} / \partial x$ is a column vector

It is called numerator-layout notation. The reversed version is called denominator-layout notation

$\partial \mathbf{y} / \partial \mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial \mathbf{x}} \\ \frac{\partial y_2}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial y_m}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1}, \frac{\partial y_1}{\partial x_2}, \cdots, \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1}, \frac{\partial y_2}{\partial x_2}, \cdots, \frac{\partial y_2}{\partial x_n} \\ \vdots \\ \frac{\partial y_m}{\partial x_1}, \frac{\partial y_m}{\partial x_2}, \cdots, \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$



Examples

$\mathbf{y}$	$\mathbf{a}$	$\mathbf{x}$	$\mathbf{Ax}$	$\mathbf{x}^T\mathbf{A}$
$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$	$\mathbf{0}$	$\mathbf{I}$	$\mathbf{A}$	$\mathbf{A}^T$

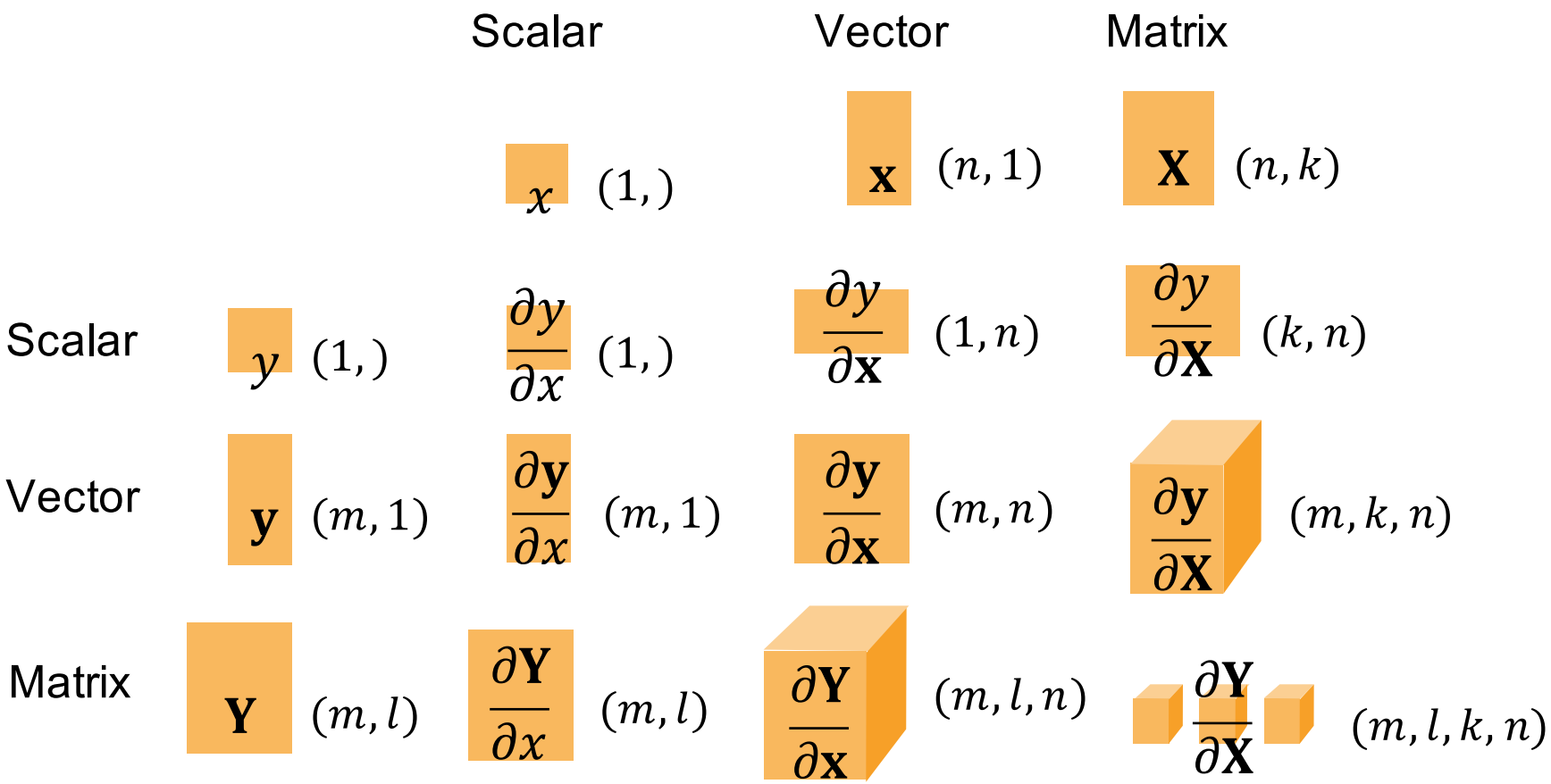
$\mathbf{y}$	$a\mathbf{u}$	$\mathbf{Au}$	$\mathbf{u}+\mathbf{v}$
$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}+\frac{\partial \mathbf{v}}{\partial \mathbf{x}}$

$\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m, \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{m \times n}$

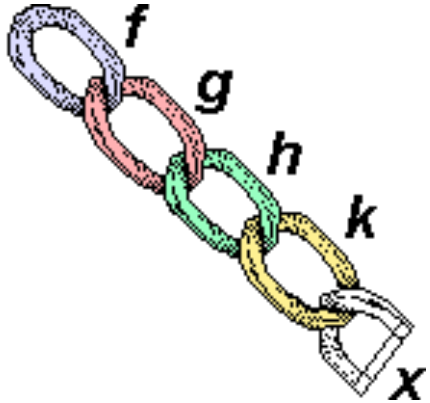
a, $\mathbf{a}$ and $\mathbf{A}$ are not functions of $\mathbf{x}$

$\mathbf{0}$ and $\mathbf{I}$ are matrices

Generalize to Matrices



Chain Rule



Generalize to Vectors

Chain rule for scalars:

$$y = f(u), u = g(x) \quad \frac{\partial y}{\partial x} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial x}$$

Generalize to vectors straightforwardly

$$\frac{\partial y}{\partial \mathbf{x}} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial \mathbf{x}} \qquad \frac{\partial y}{\partial \mathbf{x}} = \frac{\partial y}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \qquad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$$

$$(1,n) \quad (1,)(1,n) \qquad (1,n) \quad (1,k)\,(k,n) \qquad (m,n)\,(m,k)(k,n)$$

Example 1

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$$

Assume $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n, \quad y \in \mathbb{R}$

$$z = \left(\langle \mathbf{x}, \mathbf{w} \rangle - y\right)^2$$

Compute $\dfrac{\partial z}{\partial \mathbf{w}}$

$$\begin{aligned}
\frac{\partial z}{\partial \mathbf{w}} &= \frac{\partial z}{\partial b} \frac{\partial b}{\partial a} \frac{\partial a}{\partial \mathbf{w}} \\
&= \frac{\partial b^2}{\partial b} \frac{\partial a - y}{\partial a} \frac{\partial \langle \mathbf{x}, \mathbf{w} \rangle}{\partial \mathbf{w}} \\
&= 2b \cdot 1 \cdot \mathbf{x}^T \\
&= 2\left(\langle \mathbf{x}, \mathbf{w} \rangle - y\right) \mathbf{x}^T
\end{aligned}$$

Decompose

$$\begin{aligned}
a &= \langle \mathbf{x}, \mathbf{w} \rangle \\
b &= a - y \\
z &= b^2
\end{aligned}$$

Example 2

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$$

Assume $\mathbf{X} \in \mathbb{R}^{m \times n}, \quad \mathbf{w} \in \mathbb{R}^{n}, \quad \mathbf{y} \in \mathbb{R}^{m}$

$$z = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$$

Compute $\dfrac{\partial z}{\partial \mathbf{w}}$

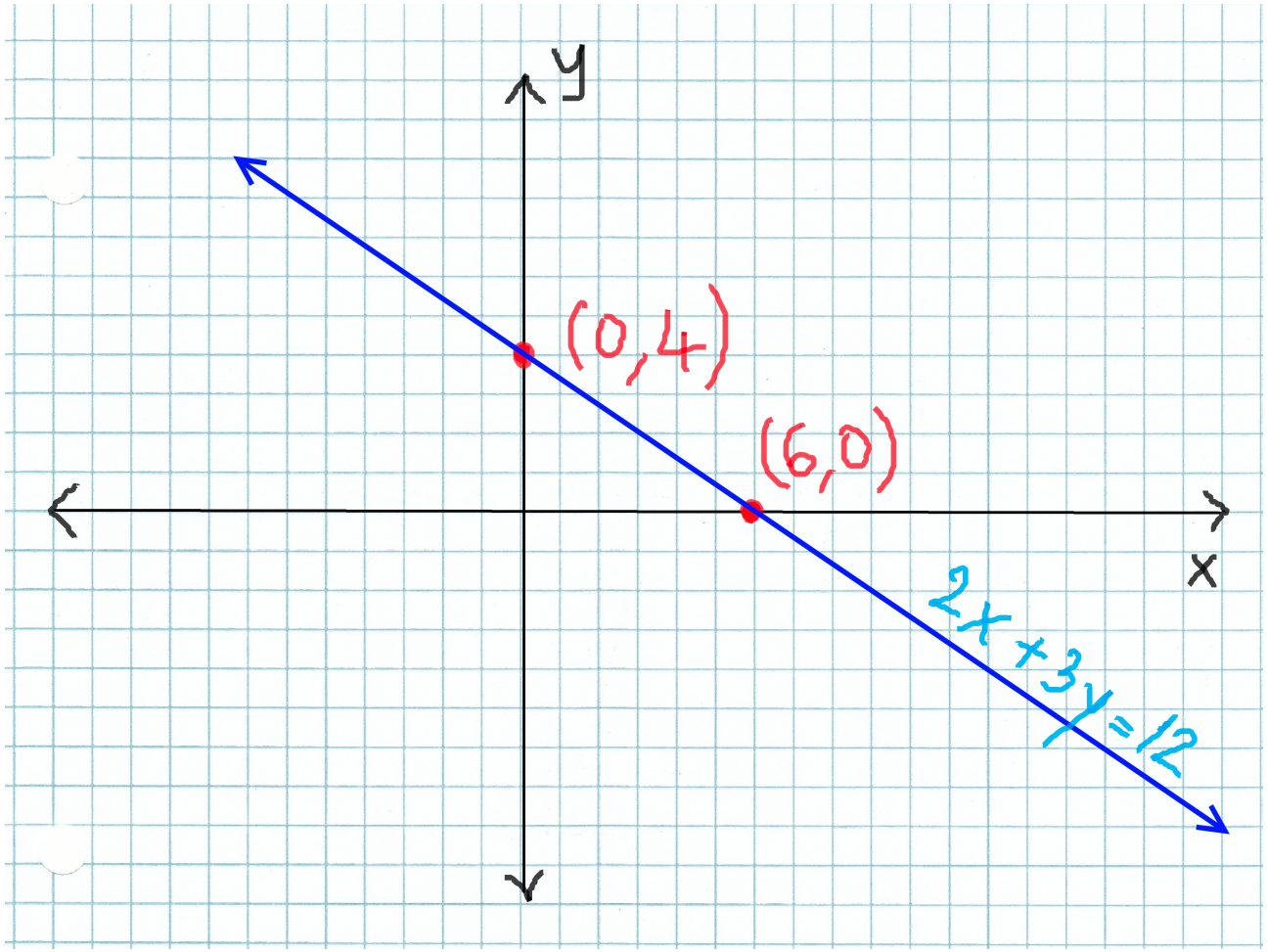
$$\begin{aligned}
\frac{\partial z}{\partial \mathbf{w}} &= \frac{\partial z}{\partial \mathbf{b}} \frac{\partial \mathbf{b}}{\partial \mathbf{a}} \frac{\partial \mathbf{a}}{\partial \mathbf{w}} \\
&= \frac{\partial \|\mathbf{b}\|^2}{\partial \mathbf{b}} \frac{\partial \mathbf{a} - \mathbf{y}}{\partial \mathbf{a}} \frac{\partial \mathbf{X}\mathbf{w}}{\partial \mathbf{w}} \\
&= 2\mathbf{b}^T \times \mathbf{I} \times \mathbf{X} \\
&= 2\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)^T \mathbf{X}
\end{aligned}$$

Decompose

$$\begin{aligned}
\mathbf{a} &= \mathbf{X}\mathbf{w} \\
\mathbf{b} &= \mathbf{a} - \mathbf{y} \\
z &= \|\mathbf{b}\|^2
\end{aligned}$$

Brief Superficial Review of Linear Regression

Linear Methods



A Simplified Model

Assumption 1

The key factors impacting y are denoted by x_1, x_2, x_3

Assumption 2

The value of y is a weighted sum over the key factors

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0$$

Weights and bias are determined later.

Linear Least Squares

Given a vector of d-dimensional inputs $\mathbf{x} = (x_1, x_2, ..., x_d)^T$, we want to predict the target (response) using the linear model:

$$y(x, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_d x_d = w_0 + \sum_{j=1}^{d} w_j x_j.$$

The term w_0 is the intercept, or often called bias term. It will be convenient to include the constant variable 1 in **x** and write:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T \mathbf{w}.$$

Observe a training set consisting of N observations

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)^T,$$

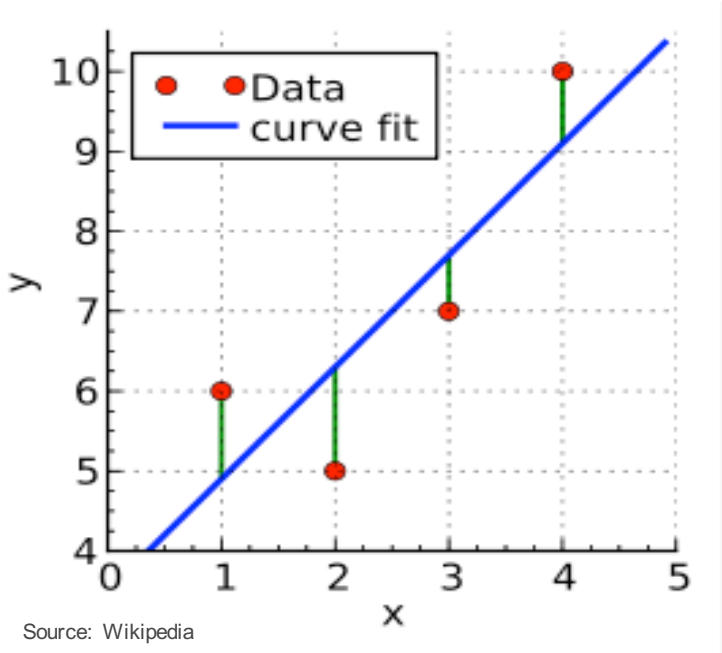
together with the corresponding target values

$$\mathbf{t} = (t_1, t_2, ..., t_N)^T.$$

Note that **X** is an $N \times (d+1)$ matrix.

Linear Least Squares

One option is to minimize **the sum of the squares of the errors** between the predictions $y(\mathbf{x}_n, \mathbf{w})$ for each data point x_n and the corresponding real-valued targets t_n.



Source: Wikipedia

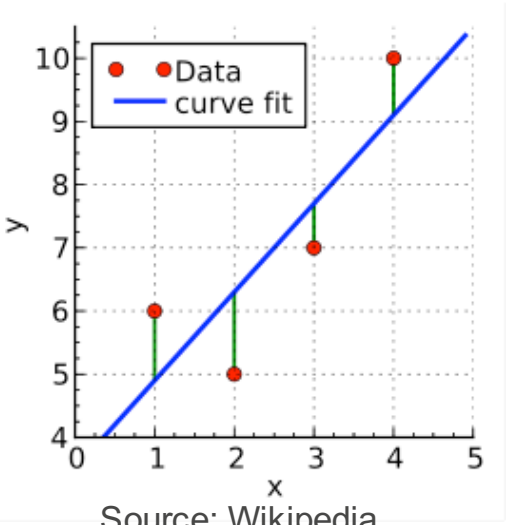
Loss function: sum-of-squared error function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n^T\mathbf{w} - t_n)^2$$

$$= \frac{1}{2}(\mathbf{Xw} - \mathbf{t})^{\mathbf{T}}(\mathbf{Xw} - \mathbf{t}).$$

Linear Least Squares

If $\mathbf{X}^{\mathbf{T}}\mathbf{X}$ is nonsingular, then the unique solution is given by:



Source: Wikipedia

optimal weights

vector of target values

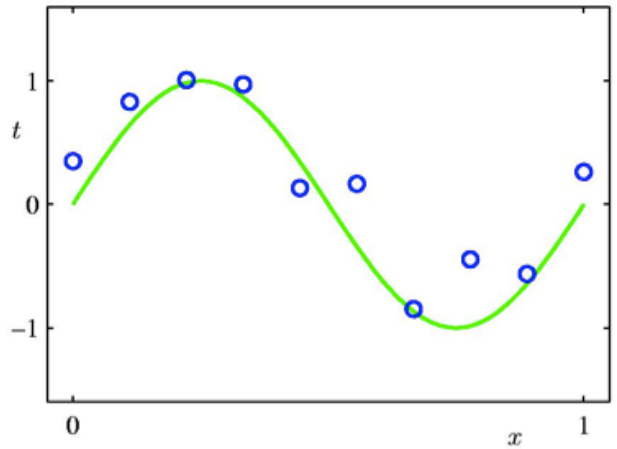
$$\mathbf{w}^* = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{t}$$

the design matrix has one input vector per row

- At an arbitrary input $\mathbf{x}_0$, the prediction is $y(\mathbf{x}_0, \mathbf{w}) = \mathbf{x}_0^T \mathbf{w}^*$.
- The entire model is characterized by d+1 parameters $\mathbf{w}^*$.

Example: Polynomial Curve Fitting

Consider observing a training set consisting of N 1-dimensional observations: $\mathbf{x} = (x_1, x_2, ..., x_N)^T$, together with corresponding real-valued targets: $\mathbf{t} = (t_1, t_2, ..., t_N)^T$.



- The green plot is the true function
- The training data was generated by $\sin(2\pi x)$. taking x_n spaced uniformly between [0 1].
- The target set (blue circles) was obtained by first computing the corresponding values of the sin function, and then adding a small Gaussian noise.

Goal: Fit the data using a polynomial function of the form:

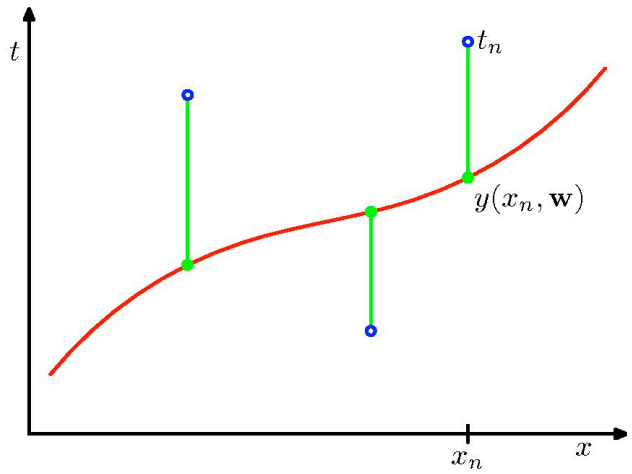
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j.$$

Note: the polynomial function is a nonlinear function of x, but it is a linear function of the coefficients **w** ! **Linear Models**.

Model Selection

Example: Polynomial Curve Fitting

• As for the least squares example: we can minimize the sum of the squares of the errors between the predictions $y(x_n, \mathbf{w})$ for each data point x_n and the corresponding target values t_n.



Loss function: sum-of-squared error function:

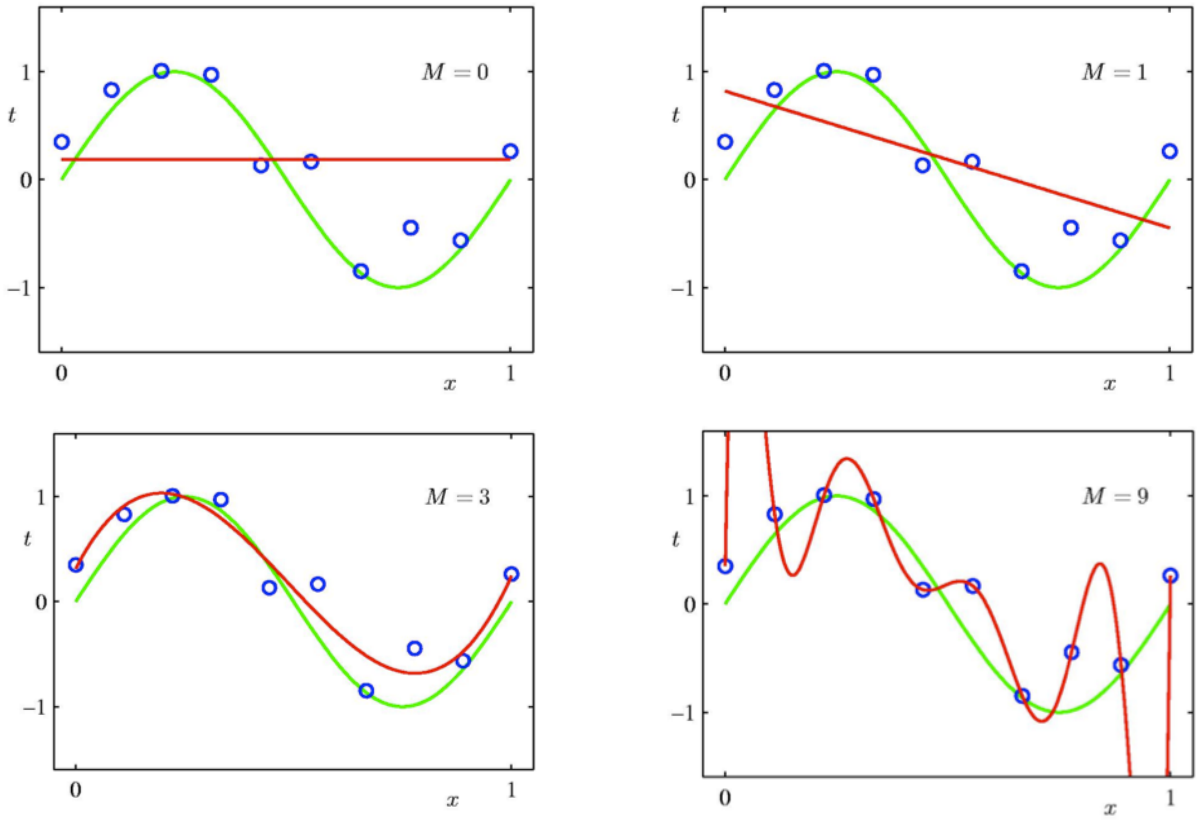
$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2.$$

• Similar to the linear least squares: Minimizing sum-of-squared error function has a unique solution $\mathbf{w}^*$.

• The model is characterized by M+1 parameters $\mathbf{w}^*$.

• How do we choose M? ! **Model Selection**.

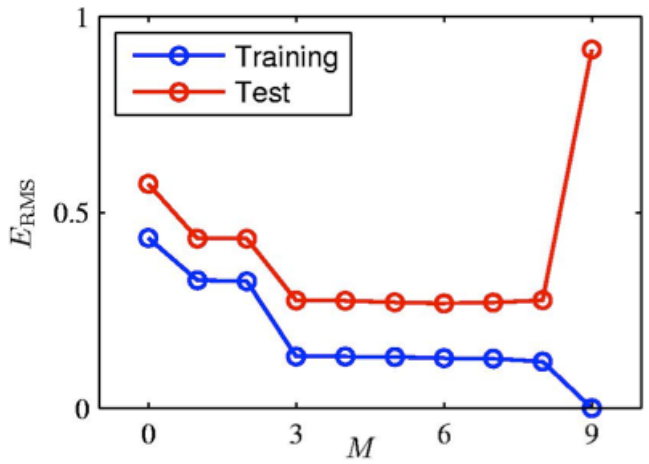
Some Fits to the Data

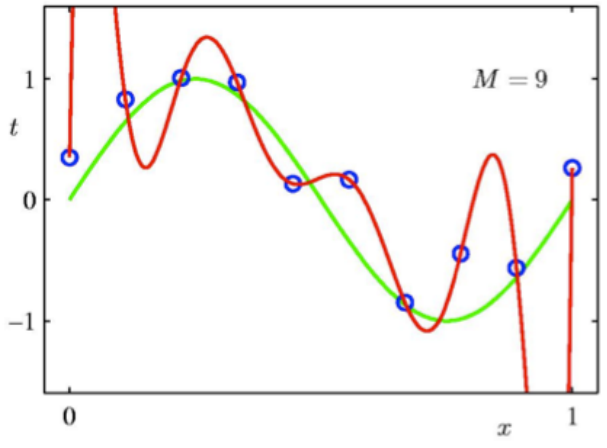


For M=9, we have fitted the training data perfectly.

Overfitting

• Consider a separate **test set** containing 100 new data points generated using the same procedure that was used to generate the training data.



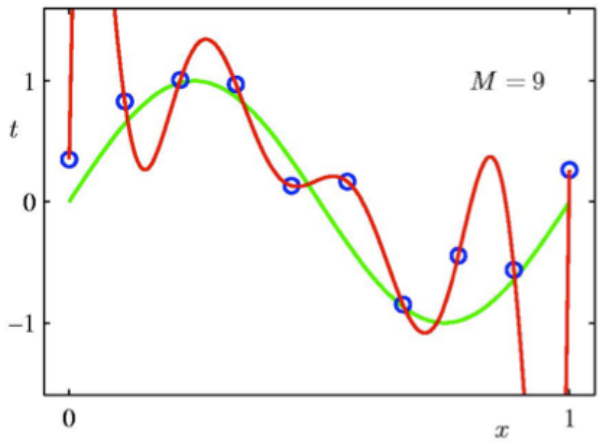


• For M=9, the training error is zero ! The polynomial contains 10 degrees of freedom corresponding to 10 parameters **w**, and so can be fitted exactly to the 10 data points.

• However, the test error has become very large. Why?

Overfitting

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

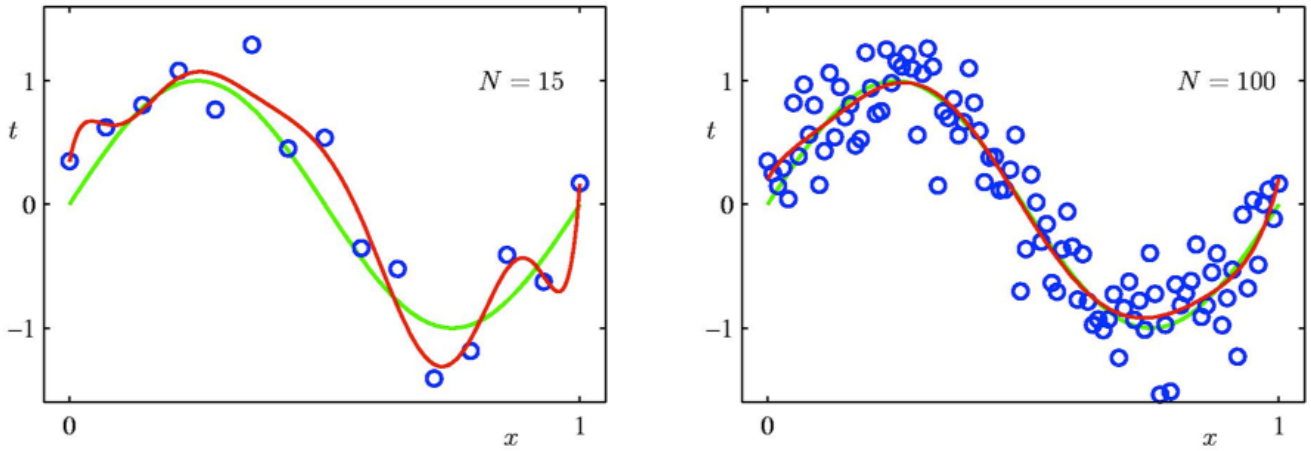


- As M increases, the magnitude of coefficients gets larger.
- For M=9, the coefficients have become finely tuned to the data.
- Between data points, the function exhibits large oscillations.

  More flexible polynomials with larger M tune to the random noise on the target values.

Varying the Size of the Data

9th order polynomial

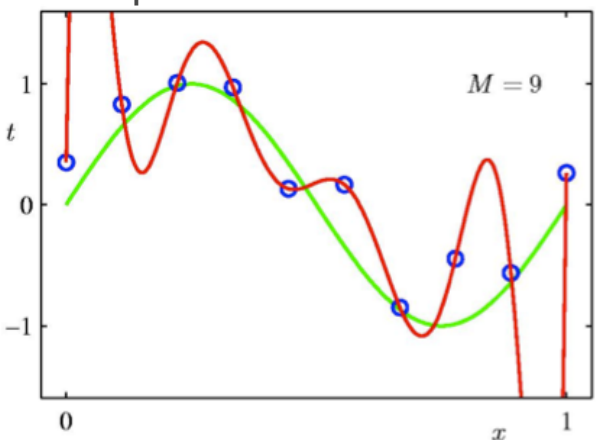


• For a given model complexity, the overfitting problem becomes less severe as the size of the dataset increases.

• However, the number of parameters is not necessarily the most appropriate measure of the model complexity.

Generalization

- The goal is achieve good **generalization** by making accurate predictions for new test data that is not known during learning.
- Choosing the values of parameters that minimize the loss function on the training data may not be the best option.
- We would like to model the true regularities in the data and ignore the noise in the data:
  - It is hard to know which regularities are real and which are accidental due to the particular training examples we happen to pick.



- Intuition: We expect the model to generalize if it explains the data well given the complexity of the model.
- If the model has as many degrees of freedom as the data, it can fit the data perfectly. But this is not very informative.
- Some theory on how to control model complexity to optimize generalization.

A Simple Way to Penalize Complexity

One technique for controlling over-fitting phenomenon is **regularization**, which amounts to adding a penalty term to the error function.

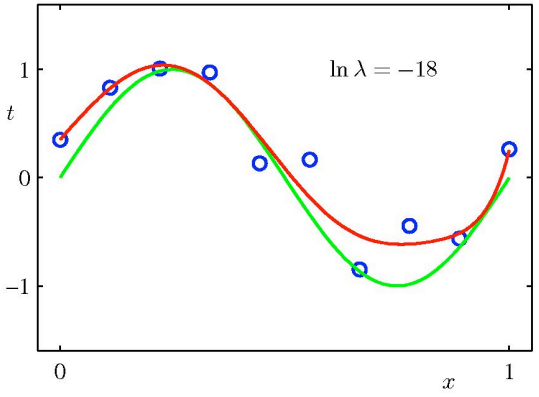
penalized error function

target value

regularization parameter

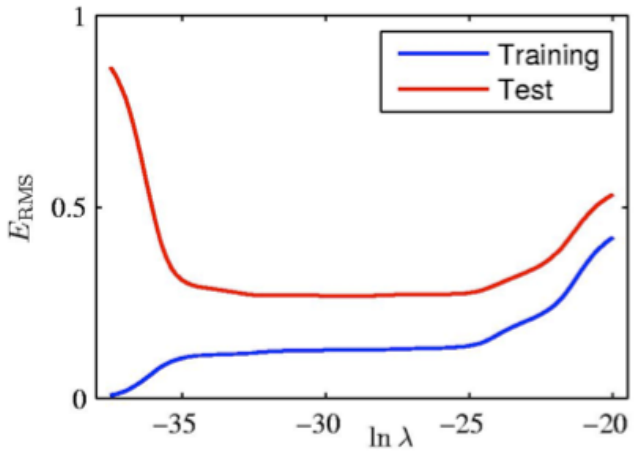
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where $||\mathbf{w}|| = \mathbf{w}^T\mathbf{w} = w_1^2 + w_2^2 + ... + w_M^2$ and , is called the regularization term. Note that we do not penalize the bias term w_0.



- The idea is to "shrink" estimated parameters towards zero (or towards the mean of some other weights).
- Shrinking to zero: penalize coefficients based on their size.
- For a penalty function which is the sum of the squares of the parameters, this is known as "**weight decay**", or "**ridge regression**".

Regularization



	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Graph of the root-mean-squared training and test errors vs. ln¸ for the M=9 polynomial.

How to choose λ?

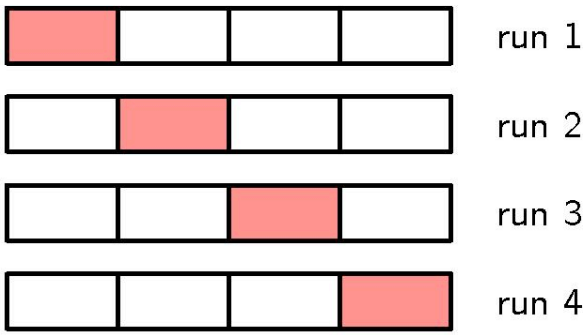
Cross Validation

If the data is plentiful, we can divide the dataset into three subsets:

- Training Data: used to fitting/learning the parameters of the model.
- Validation Data: not used for learning but for selecting the model, or choosing the amount of regularization that works best.
- Test Data: used to get performance of the final model.

For many applications, the supply of data for training and testing is limited.
To build good models, we may want to use as much training data as possible.
If the validation set is small, we get noisy estimate of the predictive performance.

S fold cross-validation



- The data is partitioned into S groups.
- Then S-1 of the groups are used for training the model, which is evaluated on the remaining group.
- Repeat procedure for all S possible choices of the held-out group.
- Performance from the S runs are averaged.

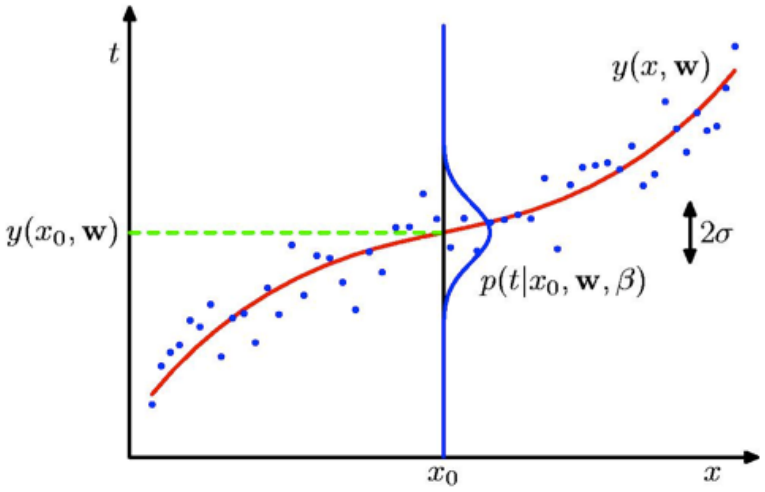
Probabilistic Perspective

• So far we saw that polynomial curve fitting can be expressed in terms of error minimization. We now view it from probabilistic perspective.

• Suppose that our model arose from a statistical model:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon,$$

where ϵ is a random error having Gaussian distribution with zero mean, and is independent of **x**.



Thus we have:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}),$$

where β is a precision parameter, corresponding to the inverse variance.

We will use probability distribution and probability density interchangeably. It should be obvious from the context.

Sampling Assumption

Assume that the training examples are drawn independently from the set of all possible examples, or from the same underlying distribution $p(\mathbf{x}, t)$.

We also assume that the training examples are identically distributed (i.i.d assumption).

Assume that the test samples are drawn in exactly the same way -- i.i.d from the same distribution as the training data.

These assumptions make it unlikely that some strong regularity in the training data will be absent in the test data.

Maximum Likelihood

If the data are assumed to be independently and identically distributed (*i.i.d assumption*), the likelihood function takes form:

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(t_n|y(\mathbf{x}_n, \mathbf{w}), \beta^{-1}).$$

It is often convenient to maximize the log of the likelihood function:

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2} \underbrace{\sum_{n=1}^{N} (y(\mathbf{x}_n, \mathbf{w}) - t_n)^2}_{\beta E(\mathbf{w})} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi).$$

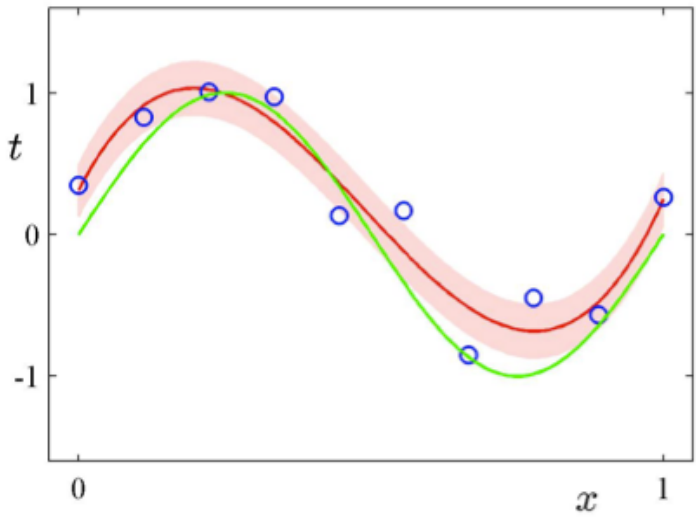
- Maximizing log-likelihood with respect to **w** (under the assumption of a Gaussian noise) is equivalent to minimizing the *sum-of-squared error* function.
- Determine $\mathbf{w}_{ML}$ by maximizing log-likelihood. Then maximizing w.r.t. β :

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n} (y(\mathbf{x}_n, \mathbf{w}_{ML}) - t_n)^2.$$

Predictive Distribution

Once we determined the parameters **w** and β, we can make prediction for new values of **x**:

$$p(t|\mathbf{x}, \mathbf{w}_{ML}, \beta_{ML}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}_{ML}), \beta_{ML}^{-1}).$$



Statistical Decision Theory

• We now develop a small amount of theory that provides a framework for developing many of the models we consider.

• Suppose we have a real-valued input vector **x** and a corresponding target (output) value t with joint probability $p(\mathbf{x}, t)$. distribution:

• Our goal is predict target t given a new value for **x**:

- for regression: t is a real-valued continuous target.
- for classification: t is a categorical variable representing class labels.

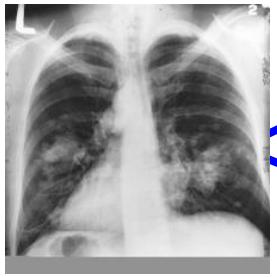
The joint probability distribution $p(\mathbf{x}, t)$ provides a complete summary of uncertainties associated with these random variables.

Determining $p(\mathbf{x}, t)$ from training data is known as the inference problem.

Example: Classification

Medical diagnosis: Based on the X-ray image, we would like determine whether the patient has cancer or not.

The input vector **x** is the set of pixel intensities, and the output variable t will represent the presence of cancer, class C_1, or absence of cancer, class C_2.



C_1: Cancer present

C_2: Cancer absent

x -- set of pixel intensities

Choose t to be binary: t=0 correspond to class C_1, and t=1 corresponds to C_2.

Inference Problem: Determine the joint distribution $p(\mathbf{x}, \mathcal{C}_k)$, or equivalently $p(\mathbf{x}, t)$. However, in the end, we must make a decision of whether to give treatment to the patient or not.

Example: Classification

Informally: Given a new X-ray image, our goal is to decide which of the two classes that image should be assigned to.

• We could compute conditional probabilities of the two classes, given the input image:

posterior probability of C_k given observed data.

probability of observed data given C_k

prior probability for class C_k

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x},\mathcal{C}_k)}{\sum_{k=1}^{K} p(\mathbf{x},\mathcal{C}_k)} = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

Bayes' Rule

• If our goal to minimize the probability of assigning **x** to the wrong class, then we should choose the class having the highest posterior probability.

Expected Loss

• Loss Function: overall measure of loss incurred by taking any of the available decisions.

Suppose that for **x**, the true class is C_k, but we assign **x** to class j ! incur loss of L_{kj} (k,j element of a loss matrix).

Consider medical diagnosis example: example of a loss matrix:

		Decision	
		cancer	normal
Truth	cancer	0	1000
	normal	1	0

Expected Loss: $$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}$$

Goal is to choose decision regions $\mathcal{R}_j$ as to minimize expected loss.

Regression

Let **x** ∈ R^d denote a real-valued input vector, and t ∈ R denote a real-valued random target (output) variable with joint the $p(\mathbf{x}, t)$. distribution

• The decision step consists of finding an estimate y(**x**) of t for each input **x**.

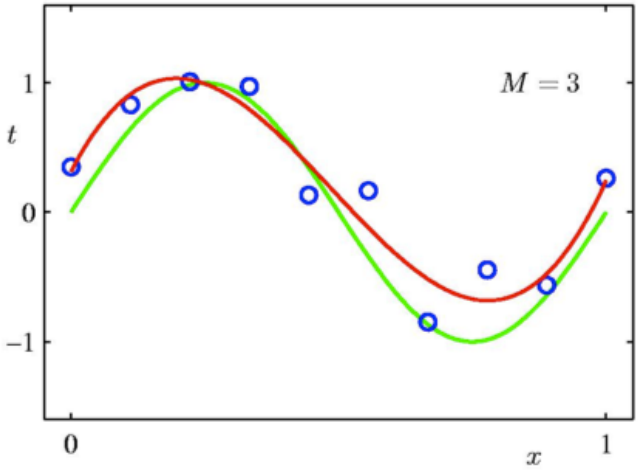
• To quantify what it means to do well or poorly on a task, we need to define a loss (error) function: $L(t, y(\mathbf{x}))$.

• The average, or expected, loss is given by:

$$\mathbb{E}[L] = \int \int L(t, y(\mathbf{x}))p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t.$$

• If we use squared loss, we obtain:

$$\mathbb{E}[L] = \int \int \big(t - y(\mathbf{x})\big)^2 p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t.$$



Squared Loss Function

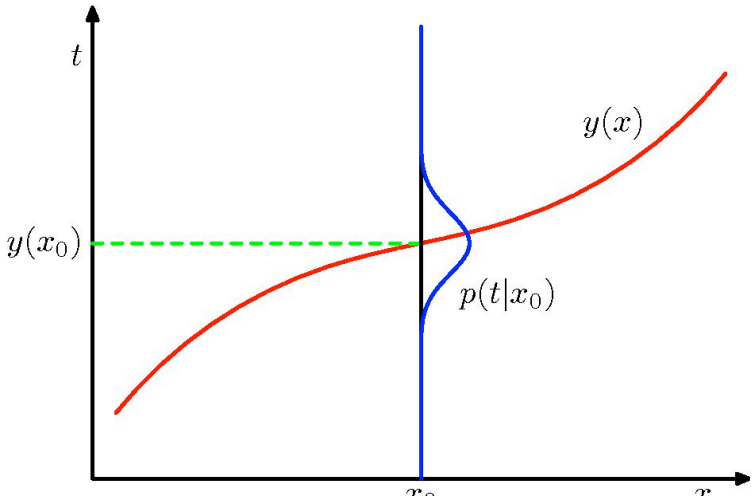
- If we use squared loss, we obtain:

$$\mathbb{E}[L] = \int \int \big(t - y(\mathbf{x})\big)^2 p(\mathbf{x}, t) \mathrm{d}\mathbf{x} \mathrm{d}t.$$

- Our goal is to choose y(x) so as to minimize the expected squared loss.

- The optimal solution (if we assume a completely flexible function) is the conditional average:

$$y(\mathbf{x}) = \int t p(t|\mathbf{x}) \mathrm{d}t = \mathbb{E}[t|\mathbf{x}].$$



The regression function y(**x**) that minimizes the expected squared loss is given by the mean of the conditional distribution $p(t|\mathbf{x})$.

Squared Loss Function

- If we use squared loss, we obtain:

$$\begin{aligned}(y(\mathbf{x}) - t)^2 &= (y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t)^2 \\ &= \big(y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\big)^2 + 2\big(y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\big)\big(\mathbb{E}[t|\mathbf{x}] - t\big) + \big(\mathbb{E}[t|\mathbf{x}] - t\big)^2.\end{aligned}$$

- Plugging into expected loss:

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 \, p(\mathbf{x}) \, \mathrm{d}\mathbf{x} + \int \mathrm{var}\,[t|\mathbf{x}] \, p(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

expected loss is minimized when $y(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$.

intrinsic variability of the target values.

Because it is independent noise, it represents an irreducible minimum value of expected loss.

Other Loss Function

• Simple generalization of the squared loss, called the *Minkowski* loss:

$$\mathbb{E}[L] = \int \int \left(t - y(\mathbf{x})\right)^q p(\mathbf{x}, t) \mathrm{d}\mathbf{x} \mathrm{d}t.$$

• The minimum of $\mathbb{E}[L]$ is given by:

- the conditional mean for q=2,
- the conditional median when q=1

Discriminative vs. Generative

- Generative Approach:

  Model the joint density: $p(\mathbf{x}, t) = p(\mathbf{x}|t)p(t)$,
  or joint distribution: $p(\mathbf{x}, \mathcal{C}_k) = p(\mathbf{x}|\mathcal{C}_k)p(C_k)$.

  Infer conditional density: $$p(t|\mathbf{x}) = \frac{p(\mathbf{x}|t)p(t)}{p(\mathbf{x})}.$$

- Discriminative Approach:

  Model conditional density $p(t|\mathbf{x})$ directly.

Linear Basis Function Models

- Remember, the simplest linear model for regression:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_d x_d = w_0 + \sum_{j=1}^{d} w_j x_j,$$

where $\mathbf{x} = (x_1, x_2, ..., x_d)^T$ is is a d-dimensional input vector (covariates).

Key property: linear function of the parameters $w_0, w_1, ..., w_d$

- However, it is also a linear function of the input variables. Instead consider:

$$y(\mathbf{x}, \mathbf{w}) = w_0 \phi_0(\mathbf{x}) + w_1 \phi_1(\mathbf{x}) + ... + w_{M-1} \phi_{M-1}(\mathbf{x}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}),$$

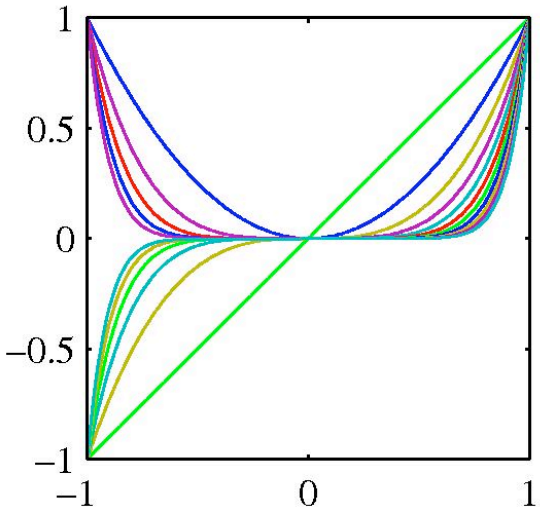
where $\phi_j(\mathbf{x})$ are known as basis functions.

- Typically $\phi_0(\mathbf{x}) = 1$ so that w_0 acts as a bias (or intercept).
- In the simplest case, we use linear bases functions: $\phi_j(\mathbf{x}) = x_j$.
- Using nonlinear basis allows the functions $y(\mathbf{x}, \mathbf{w})$ to be nonlinear functions of the input space.

Linear Basis Function Models

Polynomial basis functions:

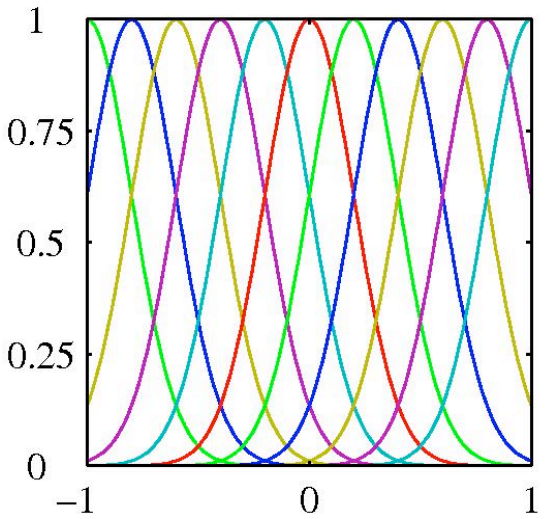
$$\phi_j(x) = x^j.$$



Basis functions are global: small changes in **x** affect all basis functions.

Gaussian basis functions:

$$\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right).$$

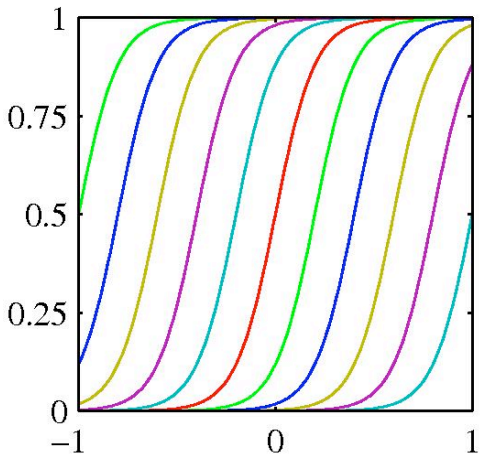


Basis functions are local: small changes in **x** only affect nearby basis functions. μ_j and s control location and scale (width).

Linear Basis Function Models

Sigmoidal basis functions

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right), \text{ where } \sigma(a) = \frac{1}{1 + \exp(-a)}.$$



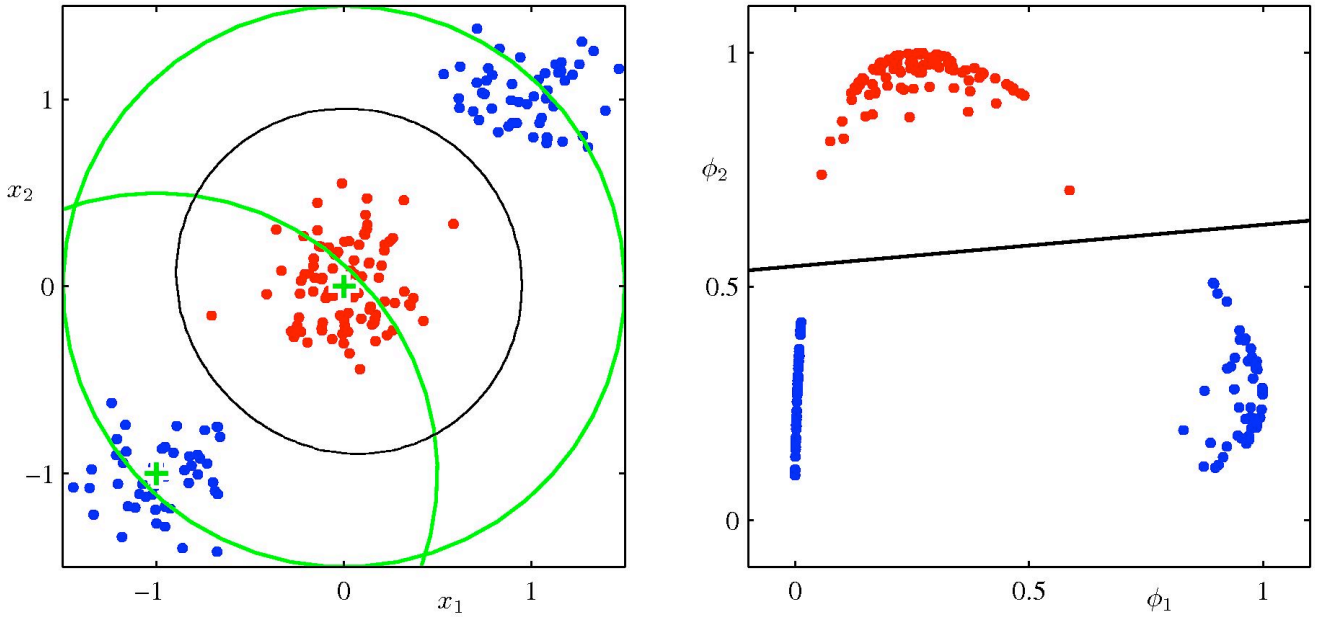
Basis functions are local: small changes in **x** only affect nearby basis functions.
μ_j and s control location and scale (slope).

- Decision boundaries will be linear in the feature space ϕ, but would correspond to nonlinear boundaries in the original input space x.
- Classes that are linearly separable in the feature space $\phi(\mathbf{x})$ need not be linearly separable in the original input space.

Linear Basis Function Models

Original input space

Corresponding feature space using two Gaussian basis functions



• We define two Gaussian basis functions with centers shown by green the crosses, and with contours shown by the green circles.

• Linear decision boundary (right) is obtained using logistic regression, and corresponds to nonlinear decision boundary in the input space (left, black curve).

Maximum Likelihood

- As before, assume observations arise from a deterministic function with an additive Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon,$$

which we can write as:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}).$$

- Given observed inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, and corresponding target values $\mathbf{t} = [t_1, t_2, ..., t_N]^T$, , under i.i.d assumption, we can write down the likelihood function:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta),$$

where $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), ..., \phi_{M-1}(\mathbf{x}))^T.$

Maximum Likelihood

Taking the logarithm, we obtain:

$$\ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \sum_{i=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n), \beta)$$

$$= \underbrace{-\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}_{\text{sum-of-squares error function}} + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi).$$

Differentiating and setting to zero yields:

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

Maximum Likelihood

Differentiating and setting to zero yields:

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

Solving for $\mathbf{w}$, we get:

$$\mathbf{w}_{\mathrm{ML}} = \underbrace{\left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \right)^{-1}}_{\text{Depends on Data}} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}$$

The Moore-Penrose pseudo-inverse of $\boldsymbol{\Phi}^{\dagger}$

where $\boldsymbol{\Phi}$ is known as the design matrix:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Sequential Learning

• The training data examples are presented one at a time, and the model parameters are updated after each such presentation (online learning):

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \bigtriangledown E_n$$

weights after seeing training case t+1

learning rate

vector of derivatives of the squared error w.r.t. the weights on the training case presented at time t.

• For the case of sum-of-squares error function, we obtain:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \left(t_n - \mathbf{w}^{(t)^T} \boldsymbol{\phi}(\mathbf{x}_n) \right) \boldsymbol{\phi}(\mathbf{x}_n).$$

• Stochastic gradient descent: The training examples are picked at random (dominant technique when learning with very large datasets).

• Care must be taken when choosing learning rate to ensure convergence.

Regularized Least Squares

• Let us consider the following error function:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

λ is called the regularization coefficient.

• Using sum-of-squares error function with a quadratic penalization term, we obtain:

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

which is minimized by setting:

Depends on Data

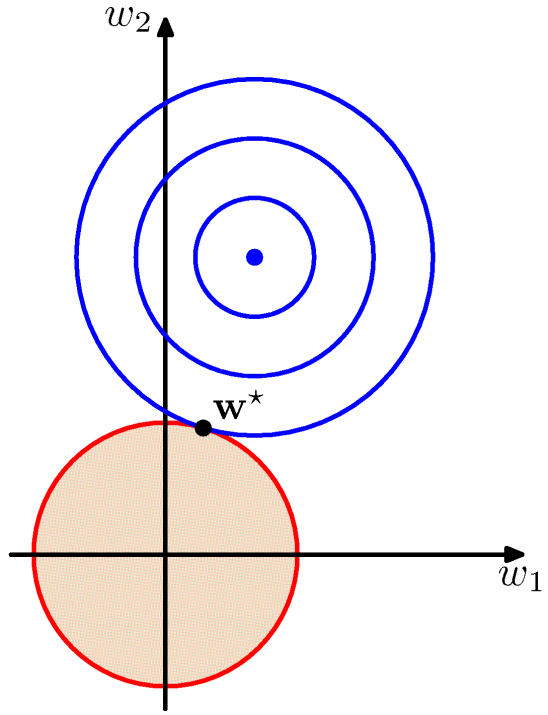
$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

Ridge regression

The solution adds a positive constant to the diagonal of $\boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Phi}$. This makes the problem nonsingular, even if $\boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Phi}$ is not of full rank (e.g. when the number of training examples is less than the number of basis functions).

Effect of Regularization

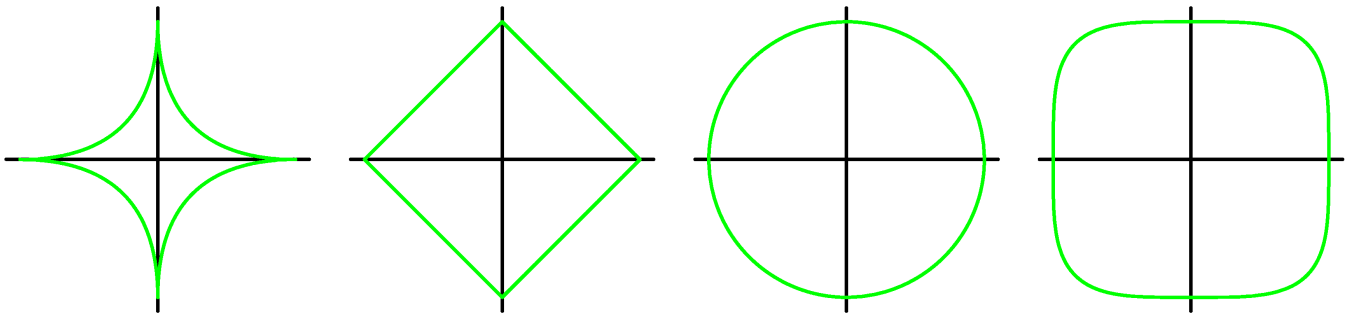
- The overall error function is the sum of two parabolic bowls.
- The combined minimum can be visualized using Lagrange Multiplier intuition.
- The regularizer shrinks model parameters to zero.



Other Regularizers

Using a more general regularizer, we get:

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Lasso Quadratic

The LASSO

- Penalize the absolute value of the weights:

$$\mathbf{w}^{lasso} = \underset{\mathbf{w}}{\operatorname{argmin}} \left[\frac{1}{2} \sum_{n=1}^{N} \left(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{M-1} |w_j| \right].$$

- For sufficiently large λ, some of the coefficients will be driven to exactly zero, leading to a sparse model.

- The above formulation is equivalent to:

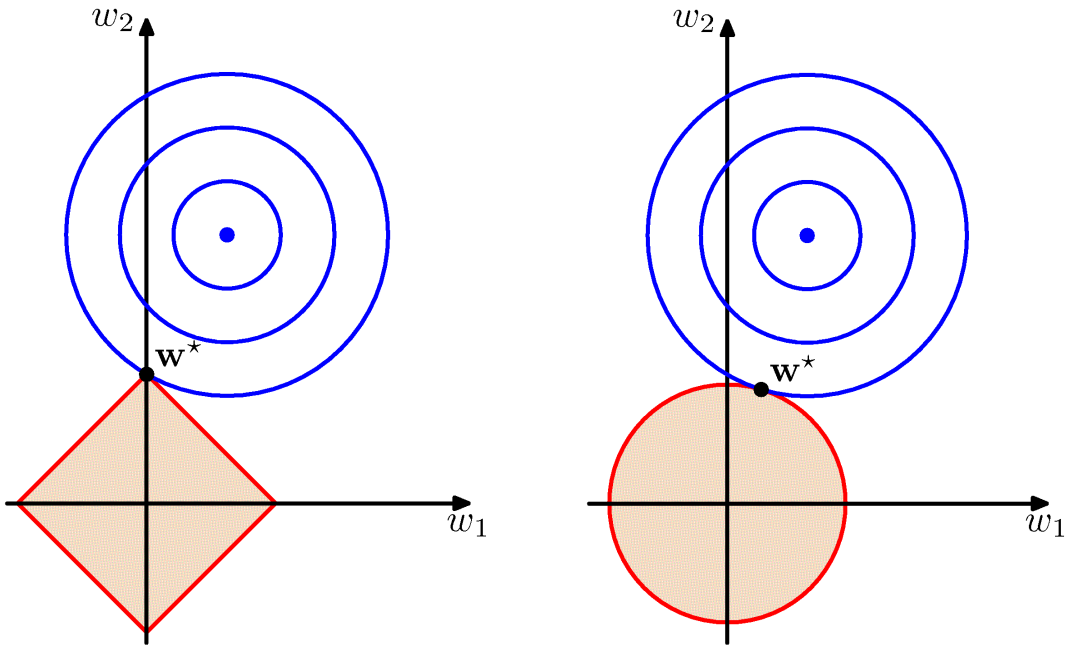
$$\mathbf{w}^{lasso} = \underset{\mathbf{w}}{\operatorname{argmin}} \underbrace{\frac{1}{2} \sum_{n=1}^{N} \left(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right)^2}_{\text{unregularized sum-of-squares error}}, \text{ subject to } \sum_{j=1}^{M-1} |w_j| \leq \tau.$$

- The two approaches are related using Lagrange multiplies.

- The LASSO solution is a quadratic programming problem: can be solved efficiently.

LASSO vs. Quadratic Penalty

LASSO tends to generate sparser solutions compared to a quadratic regualrizer (sometimes called L_1 and L_2 regularizers).



Bias-Variance Decomposition

• Introducing a regularization term can help us control overfitting. But how can we determine a suitable value of the regularization coefficient?

• Let us examine the expected squared loss function. Remember:

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

for which the optimal prediction is given by the conditional expectation:

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

intrinsic variability of the target values: The minimum achievable value of expected loss

• If we model $h(\mathbf{x})$ using a parametric function $y(\mathbf{x}, \mathbf{w})$, then from a Bayesian perspective, the uncertainly in our model is expressed through the posterior distribution over parameters $\mathbf{w}$.

• We first look at the frequentist perspective.

Bias-Variance Decomposition

• From a frequentist perspective: we make a point estimate of $\mathbf{w}^*$ based on the dataset D.

• We next interpret the uncertainly of this estimate through the following thought experiment:

- Suppose we had a large number of datasets, each of size N, where each dataset is drawn independently from $p(\mathbf{x}, t)$.
- For each dataset D, we can obtain a prediction function $y(\mathbf{x}; \mathcal{D})$.
- Different datasets will give different prediction functions.
- The performance of a particular learning algorithm is then assessed by taking the average over the ensemble of these datasets.

• Let us consider the expression:

$$\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2.$$

• Note that this quantity depends on a particular dataset D.

Bias-Variance Decomposition

- Consider:

$$\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2.$$

- Adding and subtracting the term $\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]$, we obtain

$$\begin{aligned}
&\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2 \\
&= \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&= \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad +2\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}.
\end{aligned}$$

- Taking the expectation over $\mathcal{D}$, the last term vanishes, so we get:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2\right] \\
&= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}.
\end{aligned}$$

Bias-Variance Trade-off

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

Average predictions over all datasets differ from the optimal regression function.

Solutions for individual datasets vary around their averages -- how sensitive is the function to the particular choice of the dataset.

Intrinsic variability of the target values.

$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

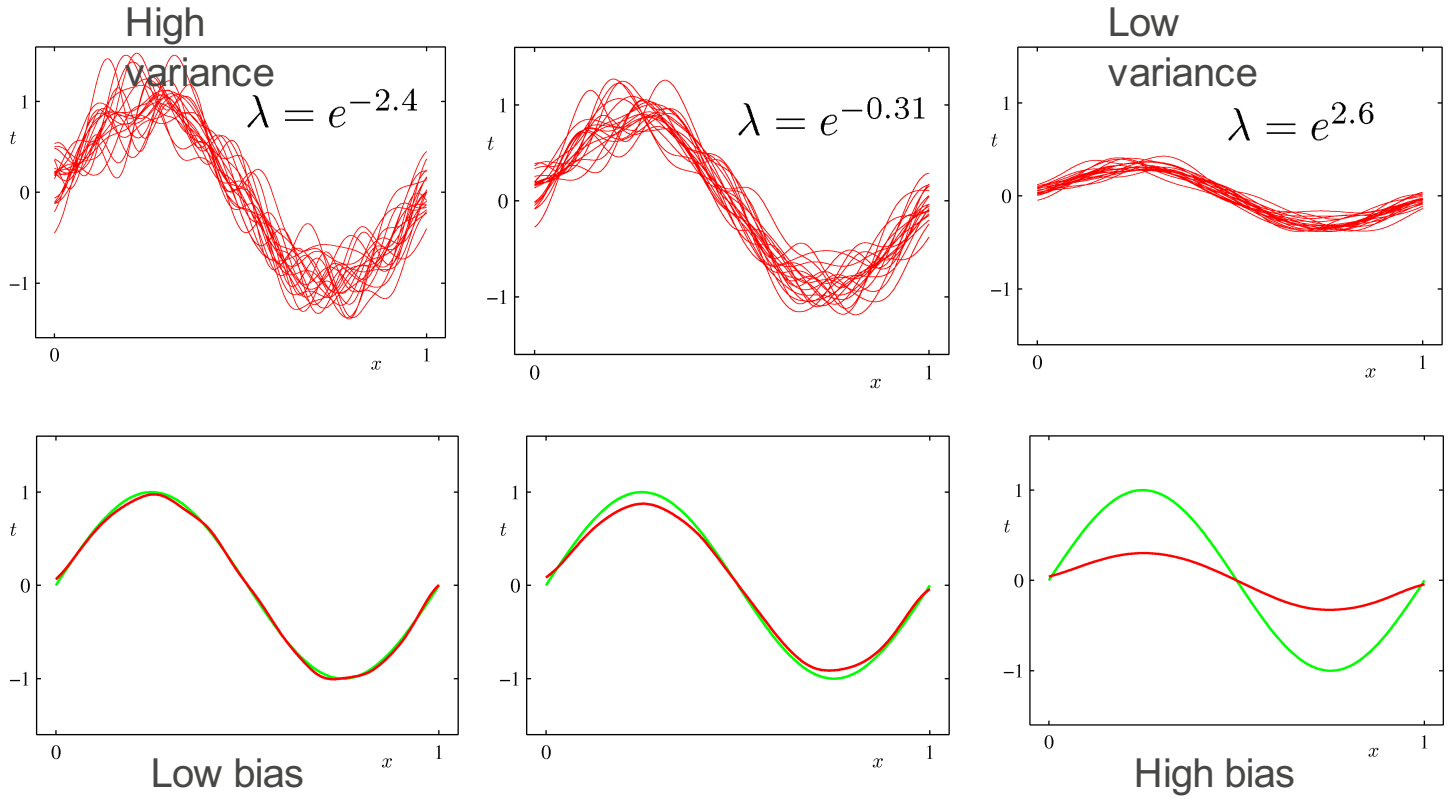
$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\text{noise} = \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

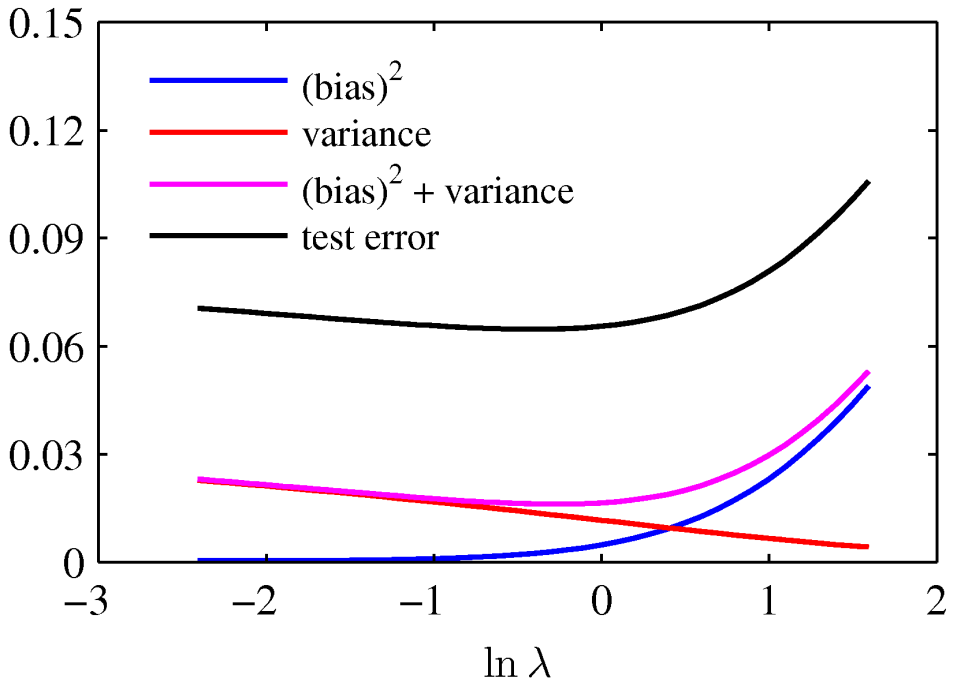
- Trade-off between bias and variance: With very flexible models (high complexity) we have low bias and high variance; With relatively rigid models (low complexity) we have high bias and low variance.
- The model with the optimal predictive capabilities has to balance between bias and variance.

Bias-Variance Trade-off

• Consider the sinusoidal dataset. We generate 100 datasets, each containing N=25 points, drawn independently from $h(x) = \sin 2\pi x$.



Bias-Variance Trade-off



From these plots note that over-regularized model (large λ) has high bias, and under-regularized model (low λ) has high variance.

Beating the Bias-Variance Trade-off

- We can reduce the variance by averaging over many models trained on different datasets:
  - In practice, we only have a single observed dataset. If we had many independent training sets, we would be better off combining them into one large training dataset. With more data, we have less variance.
- Given a standard training set D of size N, we could generate new training sets, of size N, by sampling examples from D uniformly and with replacement.
  - This is called bagging and it works quite well in practice (ad hoc).
- Given enough computation, we could also resort to the Bayesian framework:
  - Combine the predictions of many models using the posterior probability of each parameter vector as the combination weight.