Lecture 1-2: DL Basics

课程：机器学习与深度学习

Overview

- Linear Algebra
- Probability and Information Theory
- Machine Learning Basics

Linear Algebra

Overview

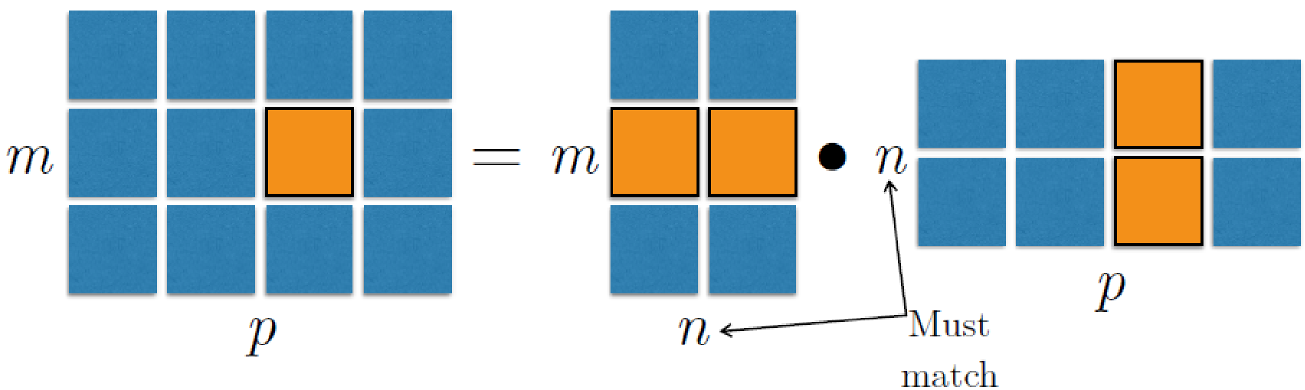
- Definitions – scalars, vectors, matrices, and tensors
- Basic Operations
- Special Kinds of Matrices and Vectors
- Norms
- Eigenvectors and Eigendecomposition
- Singular Value Decomposition
- The Moore-Penrose Pseudoinverse
- Principal Components Analysis (PCA*)

Definitions

- Scalars: x, y, a, b, …
- Vectors: **x**, **y**, **a**, **b**, …
- Matrices: **X**, **Y**, **A, B**,…
- Tensors

Basic Operations

- Matrix Transpose $(AB)^T = B^T A^T$
- Matrix Product



- Determinant: $\det(A)$
- Identity and Inverse Matrices

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix Inversion

- $A^{-1}A = I_n$
- Invertibility
  - Square matrix
  - Linearly independent
    - Span or range
    - If some columns linearly dependent, the matrix is called **singular**

Special Matrices and Vectors

- Unit vector: a vector with unit L^2 norm
- Diagonal matrices
- Symmetric Matrix

$$A = A^T$$

- Orthogonal matrix

$$A^T A = AA^T = I, A^{-1} = A^T$$

Norm

- A norm is any function f that satisfies the properties:
  - $f(x) \geq 0, and\ f(x) = 0 \rightarrow x = 0$
  - $f(x + y) \leq f(x) + f(y)$
  - $\forall\ \alpha \in \mathbb{R}, f(\alpha x) = |\alpha| f(x)$

Norms of Vectors

- Functions that measure the size of a vector
- L^p norm ($p \in \mathbb{R}, p \geq 1$)

$$||x||_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$$

- L1 norm, $||x||_1 = \sum |x_i|$
- L2 norm, Euclidean distance
- Max norm, $||x||_\infty = \max |x_i|$

Norms of Matrices

- The size of a matrix
  - Frobenius norm

$$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

  - **Nuclear norm: sum of singular values**

Eigendecomposition

- Eigenvector and eigenvalue

$$Av = \lambda v$$

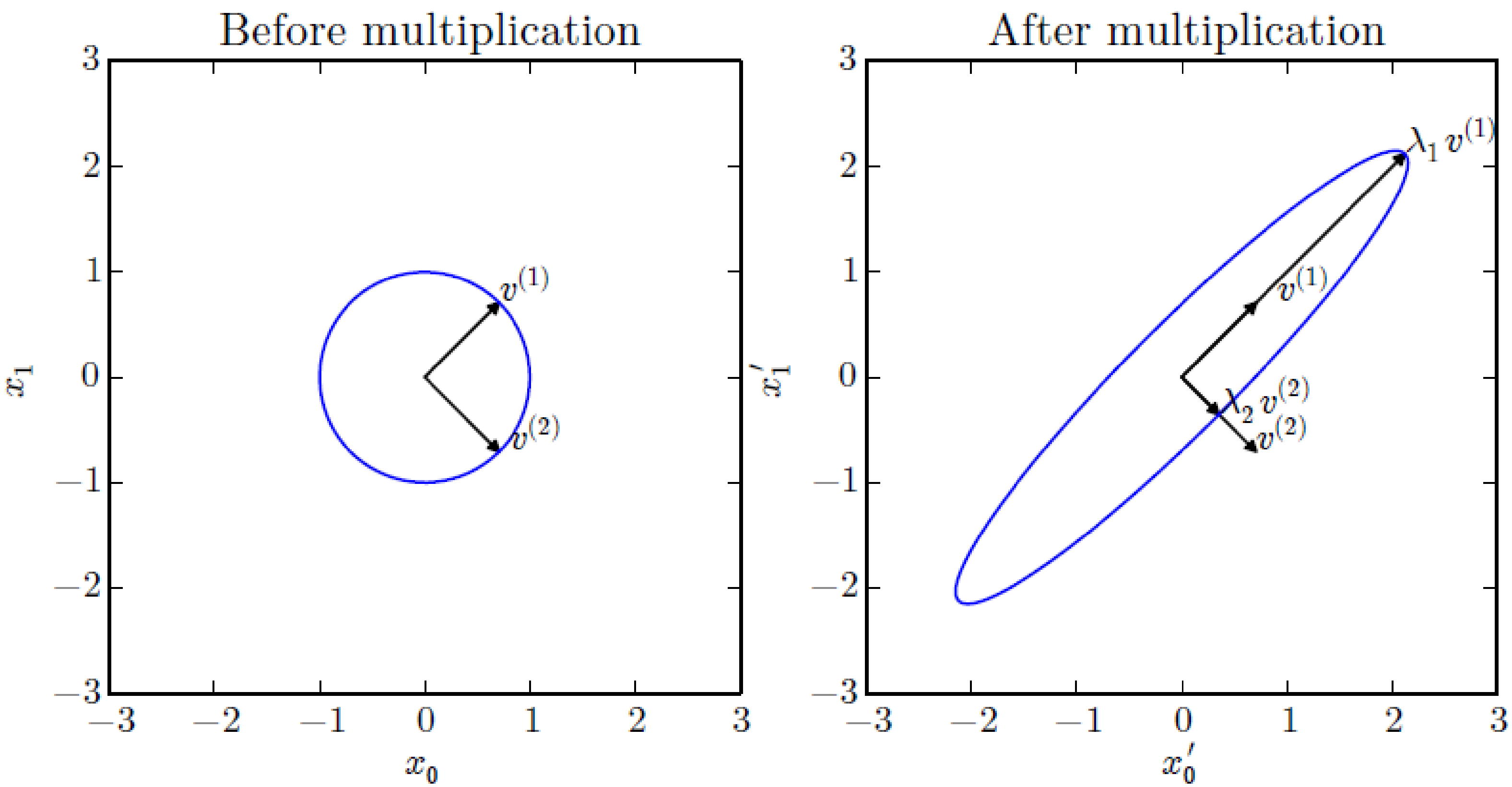
- Eigendecomposition of a nonsingular square matrix, where $V = \left[v^{(1)}, v^{(2)}, \dots, v^{(n)}\right]$, $\lambda = [\lambda_1, \lambda_2, \dots \lambda_n]^T$

$$A = V diag(\boldsymbol{\lambda}) V^{-1}$$

- Every real symmetric matrix has a real, orthogonal eigendecomposition

$$A = Q \Lambda Q^T$$

Effect of Eigenvalues



Eigendecomposition

- Positive definite
- Positive semidefinite $\forall x, x^t A x \geq 0$
- Negative definite
- Negative semidefinite

Singular Value Decomposition

- Similar to eigendecomposition
- More general; matrix need not be square

$$A = UDV^T$$

A: $m \times n$ matrix

U: $m \times m$ **orthogonal** (left singular vectors, eigenvectors of AA^T)

D: $m \times n$ **diagonal** (singular values)

V: n $\times$ n **orthogonal** (right singular vectors, eigenvectors of $A^T A$)

SVD vs. Eigenvector

- Interpret the SVD of A in terms of eigendecomposition related to A
- For $A \in \mathbb{R}^{m \times n}$, $A^T A \in \mathbb{R}^{n \times n}$
- $A = UDV^T$
- $A^T A = VD^T U^T UDV^T = V\left(D^T D\right)V^T$
- $AA^T = UDV^T VD^T U^T = U\left(D^T D\right)U^T$

The Trace Operator

- $Tr(A) = \sum_i A_{i,i}$
- $Tr(ABC) = Tr(CBA) = Tr(BCA)$
- $\|A\|_F = \sqrt{Tr(AA^T)} = \sqrt{Tr(A^TA)}$

Principal Components Analysis

- Applied to feature dimensional reduction in machine learning
- Suppose we have a collection of m points $\{x^1, x^2, \ldots, x^m\}$ *in* $\mathbb{R}^n$, transform them to $\{c^1, c^2, \ldots, c^m\}$ *in* $\mathbb{R}^l (l < n)$ by losing some precision. To find D, where $x \approx Dc$, $D \in \mathbb{R}^{n \times \mathrm{l}}$
- $c^* = \arg\min_c \|x - Dc\|_2^2$ -> $c = D^T x$
- Objective function:

$D^* = \arg\min_D \sqrt{\sum_{i,j} \left(x_j^i - (DD^T x)_{i,j} \right)^2}$, subject to $DD^T = I_l$

Probability and Information Theory

Overview

- Definitions
- Chain Rule of Conditional Probabilities
- Independence and Conditional Independence
- Expectation, Variance and Covariance
- Common Probability Distributions
- Common Functions
- Information Theory
- Structured Probabilistic Models

Why Probability?

- Uncertainty exist
  - Stochasticity
  - Incomplete observability
  - Incomplete modeling
- Interpretations
  - Frequentist
  - Subjective degrees of belief (Bayesian)

Basic definitions

- Random variables (continuous or discrete)
- Probability distributions
  - PMF, the domain of P
  - PDF, the domain of p
- Conditional probability

$$P(\mathrm{y} = y | \mathrm{x} = x) = \frac{P(\mathrm{y} = y, \mathrm{x} = x)}{P(\mathrm{x} = x)}$$

$$P(\mathrm{x} = x) > 0$$

Chain Rule of Conditional Probability

- $P(\mathrm{x}^1, \ldots, \mathrm{x}^\mathrm{n}) = P(\mathrm{x}^1) \prod_{i=2}^{n} P(\mathrm{x}^\mathrm{i}, | \mathrm{x}^1, \ldots, \mathrm{x}^{\mathrm{i}-1})$

- Bayes' Rule (prior and posterior)

$$P(\mathrm{x}|\mathrm{y}) = \frac{P(\mathrm{x})P(\mathrm{y}|\mathrm{x})}{P(\mathrm{y})}$$

  - Consider $P(GradeA|smart) = 0.6$
    - If $P(smart) = 0.3, P(GradeA) = 0.2$
    - If $P(smart) = 0.3, P(GradeA) = 0.4$
    - To compute $P(smart|GradeA)$

Independence and Conditional Independence

- Independence

$$\forall x \in \mathrm{x}, y \in \mathrm{y}, p(\mathrm{x} = x, \mathrm{y} = y) = p(\mathrm{x} = x)p(\mathrm{y} = y)$$

$$p(x|y) = p(x)$$

- Conditional independence

$$\forall x \in \mathrm{x}, y \in \mathrm{y}, \mathrm{z} \in z, p(\mathrm{x} = x, \mathrm{y} = y|\mathrm{z} = z)$$
$$= p(\mathrm{x} = x|\mathrm{z} = z)p(\mathrm{y} = y|\mathrm{z} = z)$$

Expectation, Variance and Covariance

- Expectation

$$\mathbb{E}_{x\sim p}[f(x)]$$

- Variance

$$\mathrm{Var}\big(f(x)\big) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$$

- Covariance

$$\mathrm{Cov}(f(x), g(y)) \\ = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])(g(y) - \mathbb{E}[g(y)])]$$

- Chebyshev inequality

$$P(|f(x) - \mathbb{E}_{x\sim p}[f(x)]| \geq t) \leq \frac{\mathrm{Var}\big(f(x)\big)}{t^2}$$

Bernoulli Distribution

- $P(\mathrm{x} = x) = \phi^x (1 - \phi)^{1-x}$ where $\phi \in [0,1]$
- $\mathbb{E}_{\mathrm{x}}[\mathrm{x}] = \phi$
- $\mathrm{Var}_{\mathrm{x}}[\mathrm{x}] = \phi(1 - \phi)$
- Related Distributions
    - Multinoulli Distribution
    - Binomial Distribution
    - Multinomial Distribution

Parameter Estimation

- Suppose we observed a dataset $D = \{x_1, \cdots, x_N\}$
- We can construct the likelihood function, which is a function of ϕ

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

- Equivalently we can maximize the log of the likelihood function

$$\ln p(D|\phi) = \sum \ln p(x_n|\phi)$$

Parameter Estimation

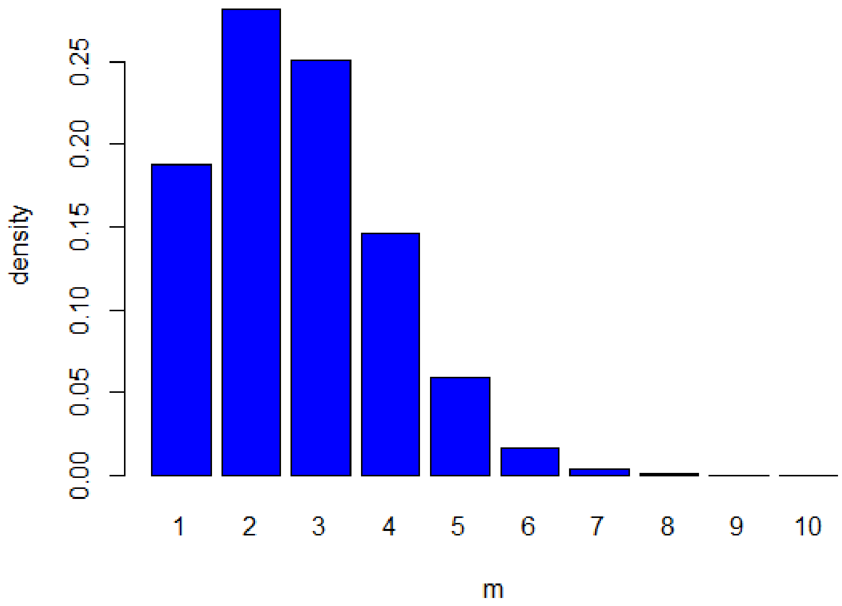
- Set the derivative of the log-likelihhood function w.r.t. ϕ to zero, we obtain $\phi_{ML} = \frac{1}{\mathrm{N}} \sum x_n$ (can compute yourself)

Binomial Distribution

- Work out the distribution of the number m of observations of $x = 1$
  - The probability of observing m ($x = 1$) given N trials and a parameter ϕ is given by:
  - $Bin(m|N, \phi) = \binom{N}{m} \phi^m (1 - \phi)^{N-m}$
  - $\mathbb{E}[m] = N\phi$
  - $Var[m] = N\phi(1 - \phi)$

Binomial Distribution: Example

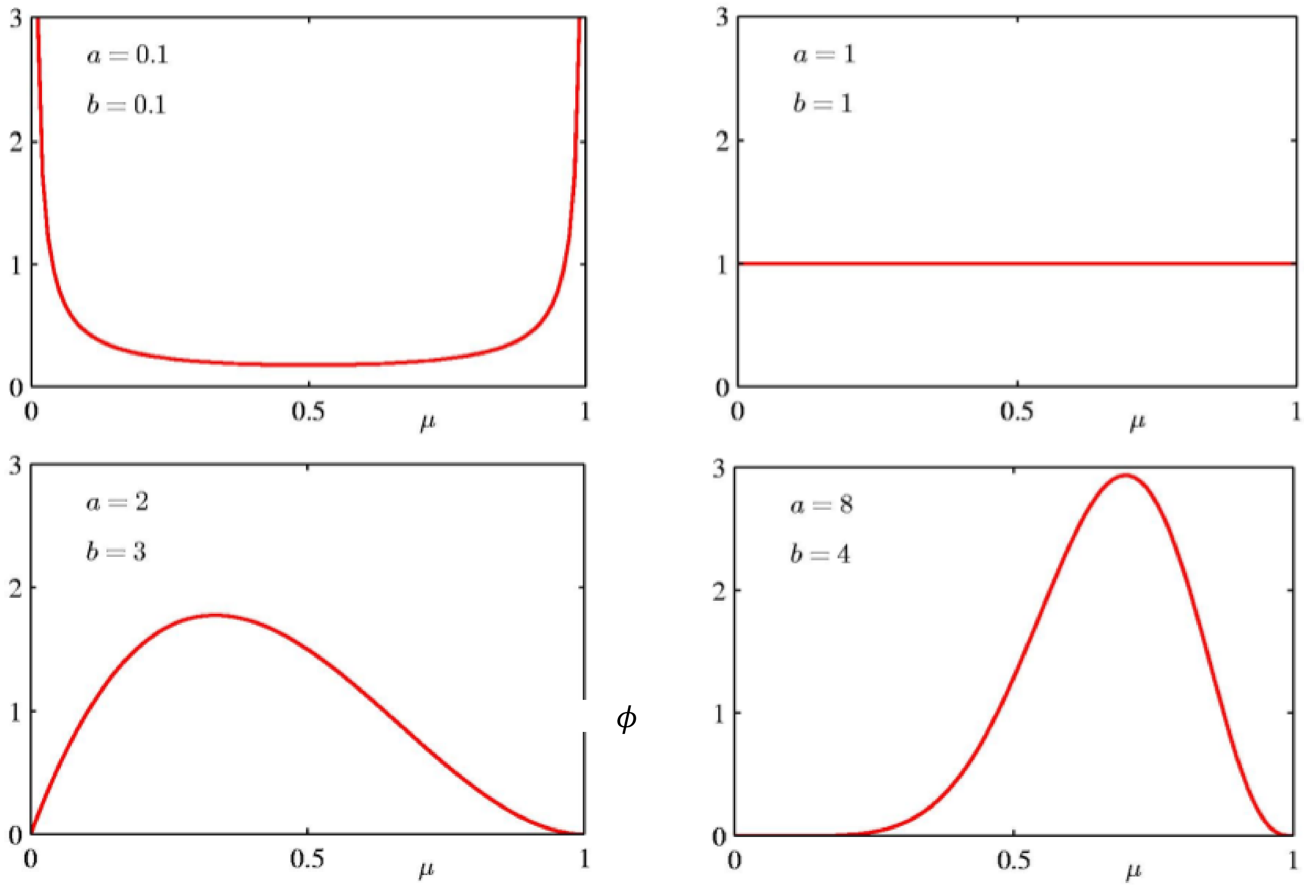
- $\text{Bin}(m|10, 0.25)$



Beta Distribution

- Define a distribution over $\phi \in [0,1]$
  - $Beta(\phi|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \phi^{a-1}(1-\phi)^{b-1}$
  - $\mathbb{E}[\phi] = \frac{a}{a+b}$
  - $Var[\phi] = \frac{ab}{(a+b)^2(a+b+1)}$
  - Where the 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 can be used as a prior over the parameter ϕ of the Bernoulli distribution

Beta Distribution: Example



Multinoulli Variables

- A generalization of Bernoulli distribution to more than two outcomes, i.e. K outcomes

$$p(x|\boldsymbol{\phi}) = \prod_{k=1}^{K} \phi_k^{x_k}$$

$$\forall k: \phi_k \geq 0 \; and \; \sum_k \phi_k = 1$$

- 1-of-K encoding schema
  - $x = (0,0,1,0,0,0)^T$, i.e. $x_3 = 1$
  - $\phi = (\phi_1, \ldots, \phi_K)$

Multinomial Distribution

- Multinoulli with N trials

$$Multi(m_1, \ldots, m_K|\boldsymbol{\phi}, N) = \binom{N}{m_1 \ldots m_K} \prod_{k=1}^{K} \phi_k^{m_k},$$

s.t. $\sum_k m_k = N$

- $\mathbb{E}[m_k] = N\phi_k$
- $Var[m_k] = N\phi_k(1 - \phi_k)$
- $Cov[m_j, m_k] = -N\phi_j\phi_k$

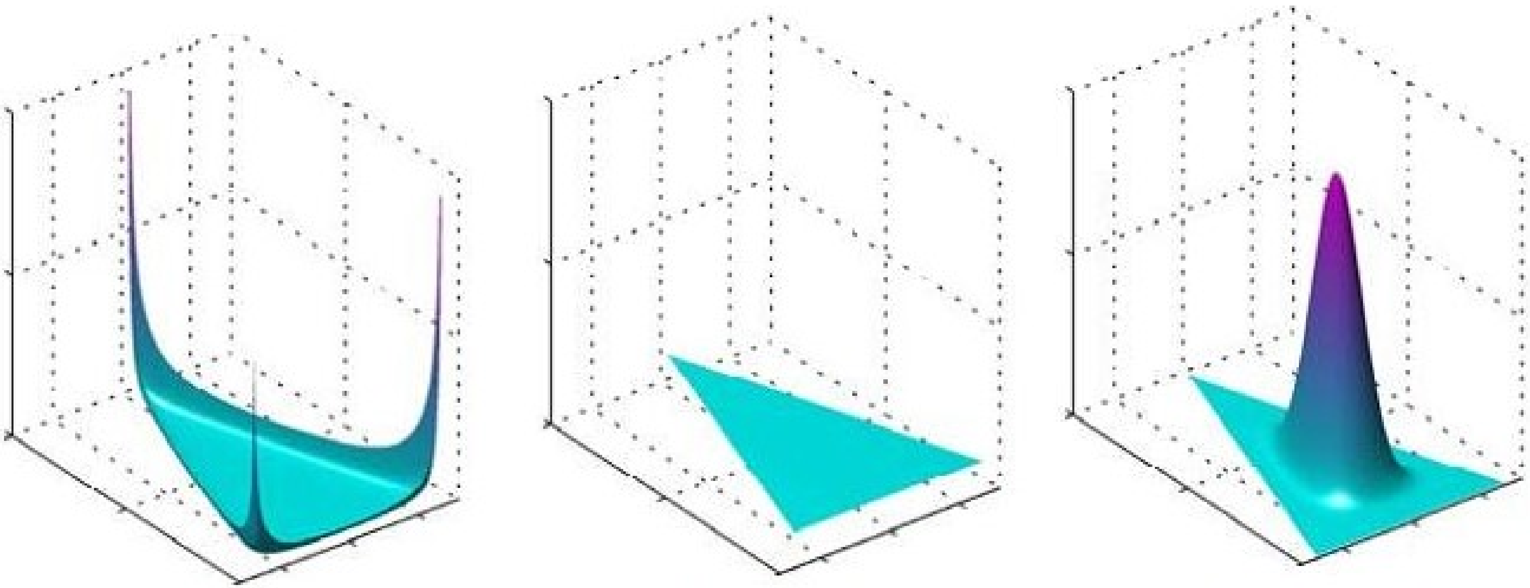
Dirichlet Distribution

- Consider a distribution over ϕ_k, s.t. constraints:

$$\forall k: \phi_k \geq 0 \ and \ \sum_k \phi_k = 1$$

- The Dirichlet distribution is defined as:
  - $Dir(\boldsymbol{\phi}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\ldots\Gamma(\alpha_K)} \prod_{k=1}^{K} \phi_k^{\alpha_k - 1}$, s. t. $\alpha_0 = \sum_k \alpha_k$
  - $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$
  - A generalization of Beta
  - $\mathbb{E}[\phi] = \frac{\alpha}{\alpha_0}$; $Var[\boldsymbol{\phi}] = \frac{\alpha .* (\alpha_0 - \alpha)}{\alpha_0^2(\alpha_0 + 1)}$; $Cov[\phi_j, \phi_k] = \frac{-\alpha_j \alpha_k}{\alpha_0^2(\alpha_0 + 1)}$

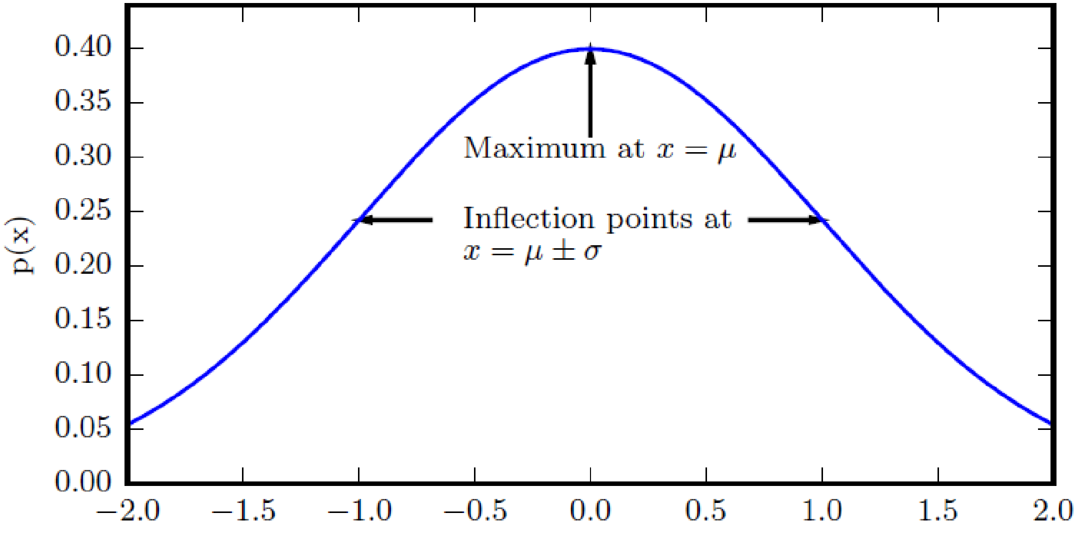
Dirichlet Distribution: Example



$\{\alpha_k\} = 0.1$ $\{\alpha_k\} = 1$ $\{\alpha_k\} = 10$

Gaussian Distribution

- $N(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
  - $\mu\ (mean)$
  - $\sigma^2 (variance)$
  - Precision: $\beta = 1/\sigma^2$



Multivariate Gaussian

$$N(x;\mu,\Sigma) = \sqrt{\frac{1}{(2\pi)^n \det(\Sigma)}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$

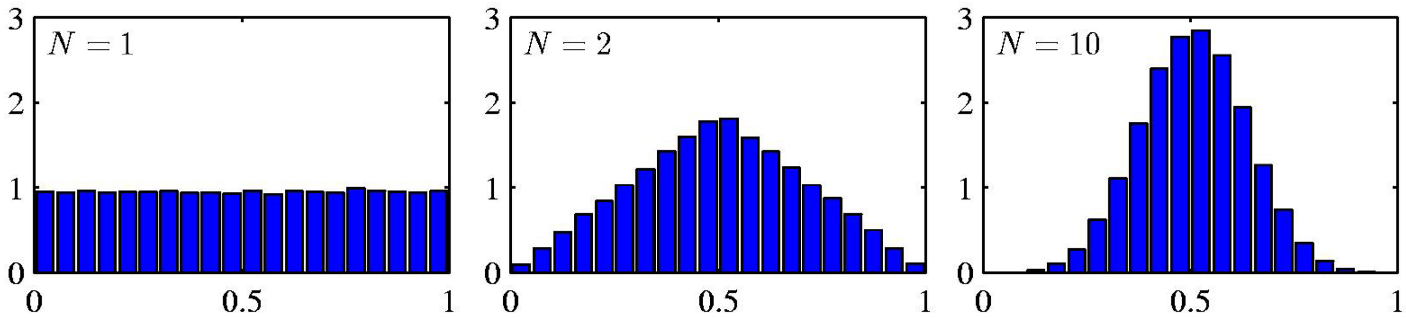
- μ is a n-dimensional mean vector
- Σ is a n by n covariance matrix

Gaussian Distribution

- A default choice
  - Many real cases approximate to: central limit theorem
  - Encode the maximum uncertainty with the same variance (textbook p638-639)

Central Limit Theorem

- The Distribution of the sum (or mean) 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 $\frac{x_1+x_2+\cdots+x_N}{N}$



Exponential and Laplace Distributions

- Exponential distribution
  - A sharp point at $x = 0$

$$p(x; \lambda) = \lambda 1_{x \geq 0} \exp(-\lambda x)$$

- Laplace distribution
  - A sharp point at $x = \mu$

$$Laplace(x; \mu, \gamma) = \frac{1}{2\gamma} \exp\left(-\frac{|x - \mu|}{\gamma}\right)$$

Dirac and Empirical Distributions

- Dirac distribution
  - $p(x) = \delta(x - \mu)$
- Empirical distribution

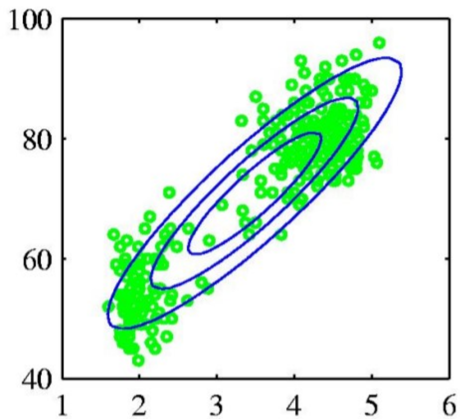
$$\hat{p}(x) = \frac{1}{m} \sum_{i=1}^{m} \delta(x - x^{i})$$

Mixture Distributions

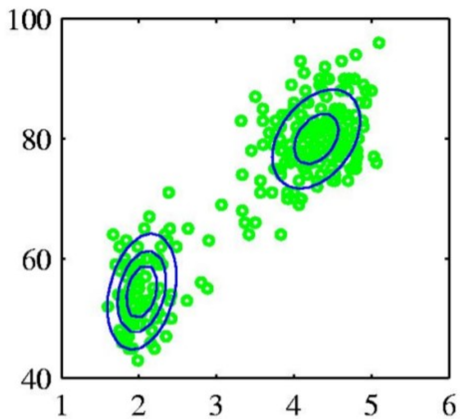
- A common way of combine distributions

$$P(\mathrm{x}) = \sum_i P(c = i) P(\mathrm{x} | c = i)$$

- Gaussian mixture distributions
  - Universal approximator of densities



Single Gaussian



Mixture of two Gaussians

Conjugate Distributions

- Bayesian probability

$$P(\theta|\mathrm{x}) = \frac{P(\mathrm{x}|\theta)P(\theta)}{P(\mathrm{x})}$$

  - The prior distribution $P(\theta)$ (conjugate prior) and the posterior distribution $P(\theta|\mathrm{x})$ are in the same family
- Typical examples
  - Bernoulli (or binomial), Beta
  - Multinoulli (or multinomial), Dirichlet
  - Possion, Gamma
  - Gaussian, Gaussian
  - …..

The Exponential Family

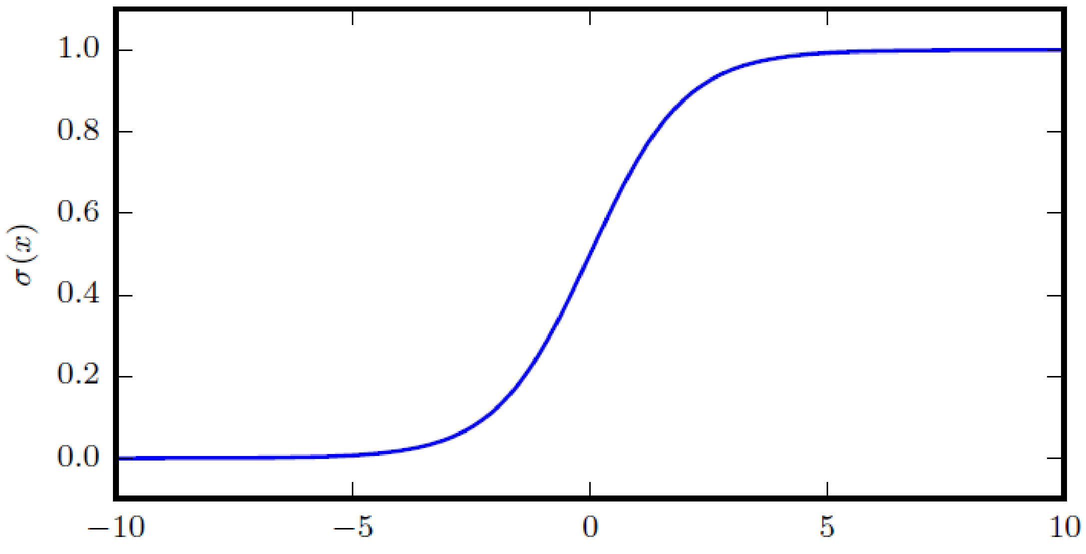
- The exponential family of distributions over x is in the form of
  - $p(x|\eta) = h(x)g(\eta)\exp\{\eta^T u(x)\}$
  - η: vector of natural parameters, $u(x)$: vector of sufficient statistics; g(η): normalizer, ensure p is normalized
  - $g(\eta)\int h(x)g(\eta)\exp\{\eta^T u(x)\}\,dx = 1$
    - (Try to re-write the aforementioned distributions to the form)
    - (Conduct the MLE for exponential family distributions)

Parameters Estimation

- MLE
- Bayesian
- MAP
- …

Logistic Sigmoid

- $\sigma(x) = \frac{1}{1+\exp(-x)}$
- Parameterize Bernoulli distribution

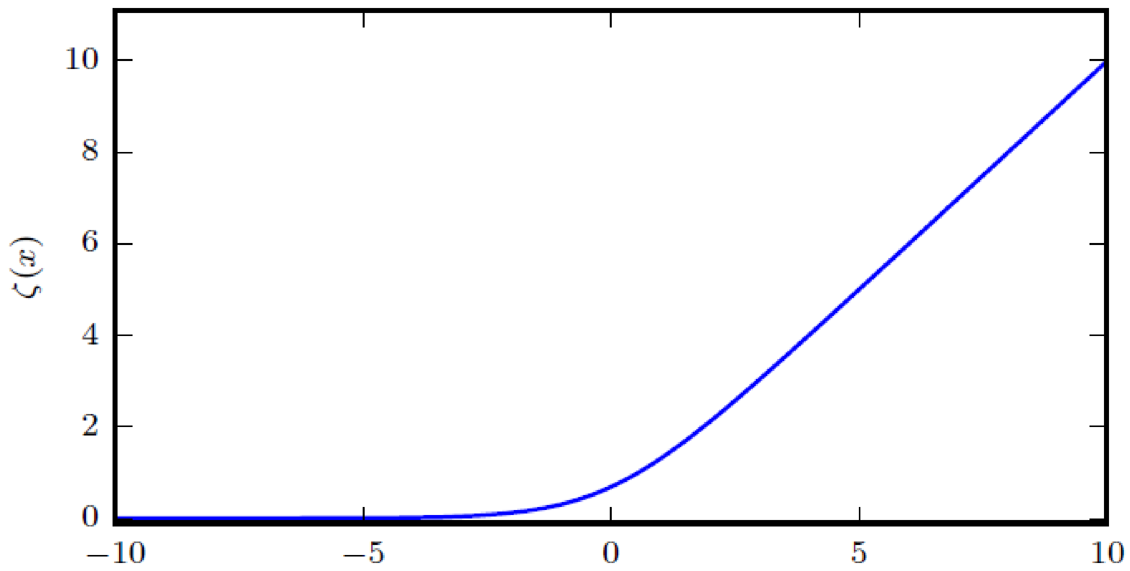


Softmax Function

- $\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$
- Parameterize the multinoulli distribution

Softplus Function

- $\zeta(x) = \log(1 + \exp(x))$
- Parameterize the variance of normal distribution



Properties of Sigmoid and Softplus Functions

- $\left(\sigma(x)\right)' = \sigma(x)\left(1 - \sigma(x)\right)$
- $1 - \sigma(x) = \sigma(-x)$
- $\log\left(\sigma(x)\right) = -\zeta(-x)$
- $\left(\zeta(x)\right)' = \sigma(x)$
- $\zeta(x) - \zeta(-x) = x$
- $\zeta(x) = \int_{-\infty}^{x} \sigma(y)dy$
- $\forall x > 0, \zeta^{-1}(x) = \log(\exp(x) - 1)$
- $\forall x \in (0,1), \sigma^{-1}(x) = \log(\frac{x}{1-x})$

Information Theory

- Likely event -> lower information
- Unlikely event -> higher information
- Independent events -> additive information

- Information

$$I(x) = -\log P(x)$$

- Entropy

$$H(x) = \mathbb{E}_{x \sim P}[I(x)]$$

- KL divergence

$$D_{KL}(P \parallel Q) = \mathbb{E}_{x \sim P}[\log \frac{P(x)}{Q(x)}]$$

- nonnegative

  - Asymmetric
    - $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$
  - Cross-entropy

$$H(P, Q) = H(P) + D_{KL}(P \parallel Q) \Longrightarrow H(P, Q) = -\mathbb{E}_{x \sim P} \log Q(x)$$

Example: Entropy

- Four cases
- $P(C1) = \frac{0}{6} = 0; P(C2) = \frac{6}{6} = 1$
- $\text{Entropy} = -0 \log 0 - 1 \log 1 = 0$
- $P(C1) = \frac{1}{6}; P(C2) = \frac{5}{6}$
- $\text{Entropy} = -\frac{1}{6} \log \frac{1}{6} - \frac{5}{6} \log \frac{5}{6} = 0.65$
- $P(C1) = \frac{2}{6}; P(C2) = \frac{4}{6}$
- $\text{Entropy} = -\frac{2}{6} \log \frac{2}{6} - \frac{4}{6} \log \frac{4}{6} = 0.92$
- $P(C1) = \frac{3}{6}; P(C2) = \frac{3}{6}$
- $\text{Entropy} = -\frac{3}{6} \log \frac{3}{6} - \frac{3}{6} \log \frac{3}{6} = 1$

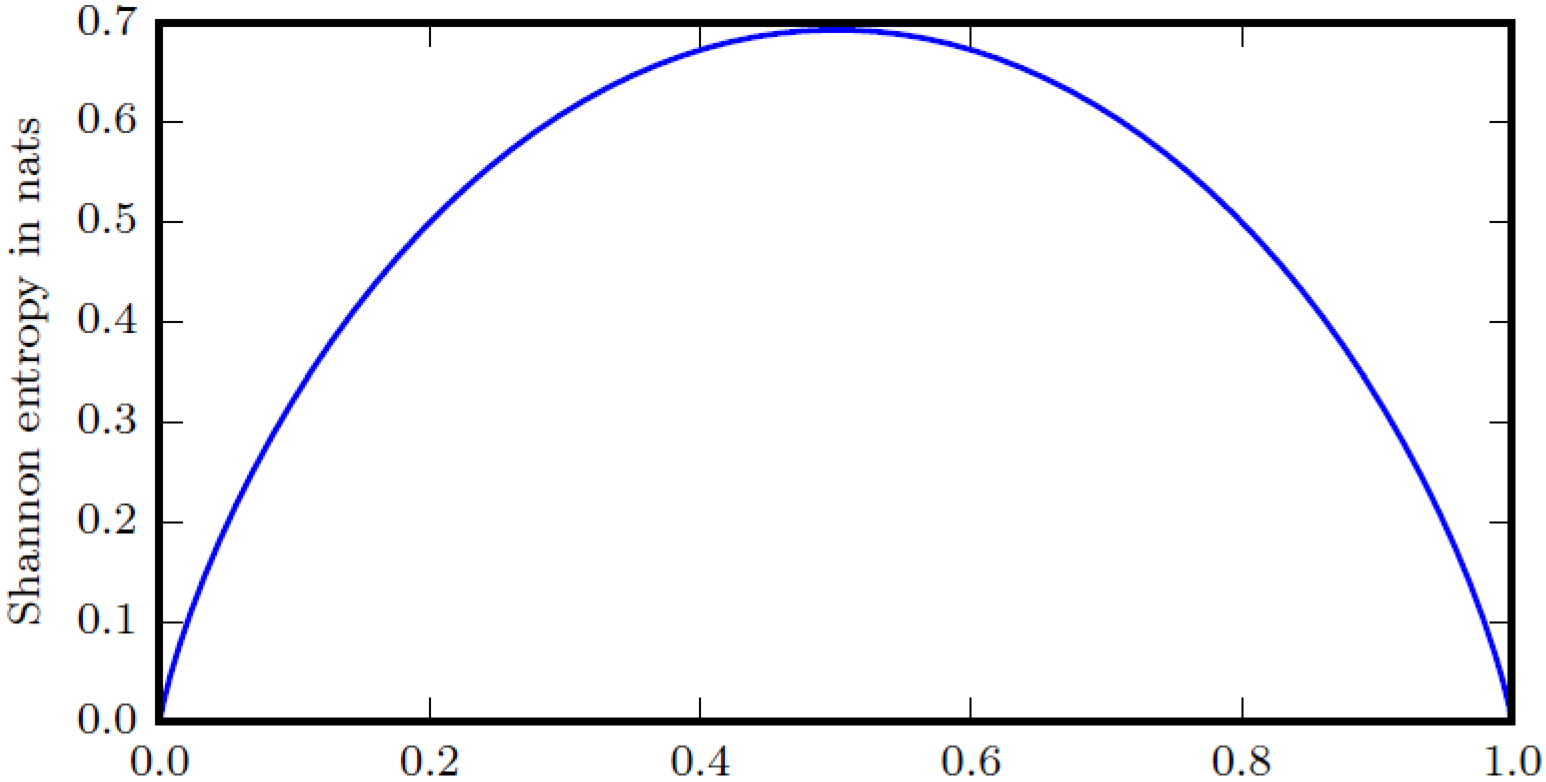
C1	**0**
C2	**6**

C1	**1**
C2	**5**

C1	**2**
C2	**4**

C1	**3**
C2	**3**

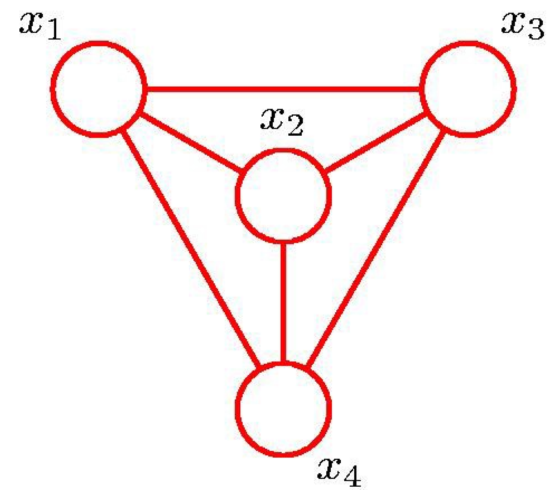
Entropy of a Bernoulli Variable



Structural Probabilistic Models

- Also called graphical models
- Provide a powerful framework for representing dependency structure between random variables
- Properties
  - A simple way to visualize the structure
  - Various insights into the properties of model, e.g. conditional independence
  - Express complex computations in terms of graphical manipulations

Structural Probabilistic Models

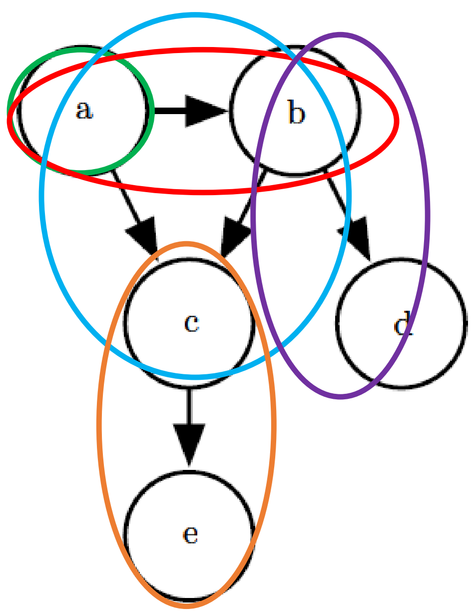


- A set of nodes and links
  - Node: random variable
  - Link: probabilistic dependency
- Two types
  - Directed graphical model: Bayesian network
  - Undirected graphical model: Markov random fields

Directed Graphical Models

- Useful for expressing causal relationships between random variables

$$p(x) = \prod_i p\big(x_i \big| Pa_{\mathcal{G}}(x_i)\big)$$



If fully connected: $p(a,b,c,d,e) =$
$p(a)p(b|a)p(c|b,a)p(d|c,b,a)p(e|a,b,c,d)$

Not fully connected$\Longrightarrow$ $p(a,b,c,d,e) =$
$p(a)p(b|a)\ (c|b,a)p(d|b)p(e|c)$

No directed cycles

Directed Graphical Model: Example

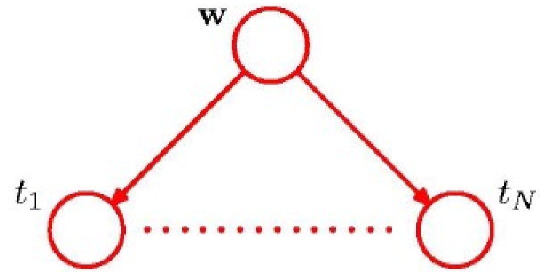
- Consider Bayesian polynomial regression

$$y(x, w) = \sum_j w_j x^j$$

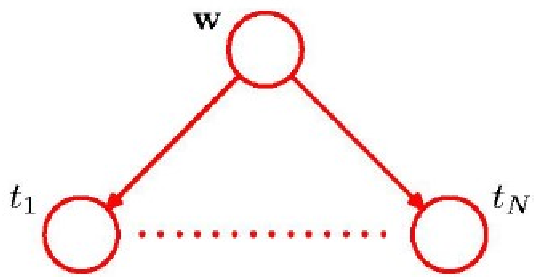
- Given inputs $X = \{x_1, x_2, \dots, x_N\}$, $\mathrm{t} = \{\mathrm{t}_1, \dots, \mathrm{t}_\mathrm{N}\}$

$$P(t, w|X) = p(w) \prod_{n=1}^{N} p(t_n|w, x_n)$$

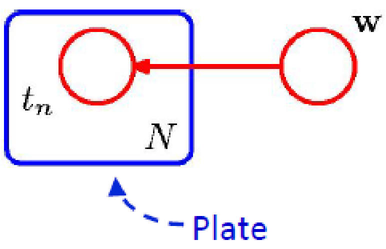
- Can be represented as:

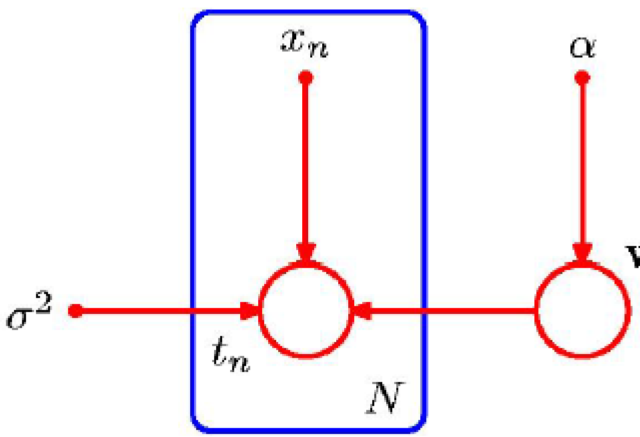


Directed Graphical Model: Example

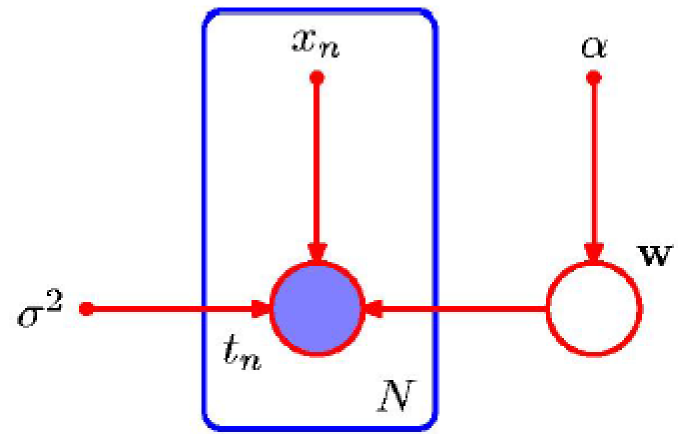


Same representation using plate notation





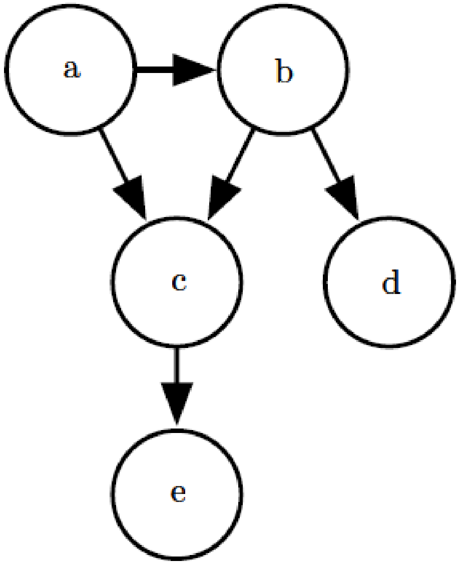
$$p(w|\alpha) = \mathcal{N}(w|0, \alpha I)$$
$$p\left(t_n \middle| w, x_n, \sigma^2\right) = \mathcal{N}(t_n|y(w, x_n), \sigma^2)$$



$$p(w|t) \propto p(w) \prod_n p(t_n|w)$$

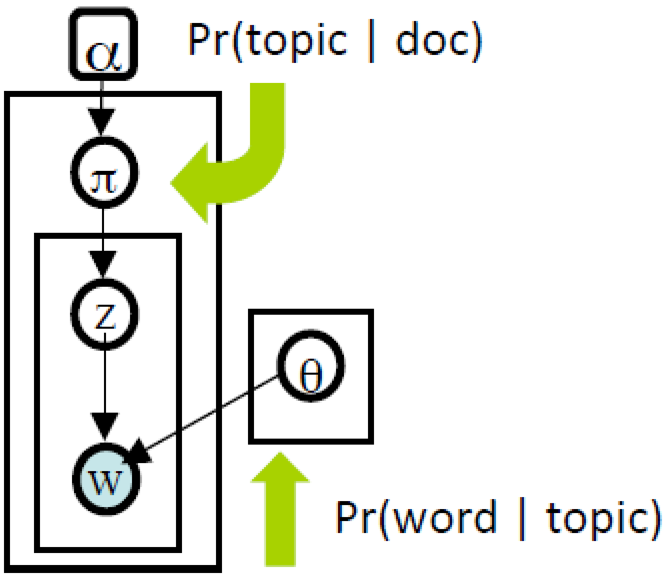
Conditional Independence

- Common parent
  - $c \leftarrow b \rightarrow d, c \perp d$ if b observed; if b unobserved, then not
- Cascade
  - $a \rightarrow c \rightarrow e, a \perp e$ if c observed;if c unobserved, then not
- V-structure
  - $a \rightarrow \mathrm{c} \leftarrow \mathrm{b}, a \perp \mathrm{b}$ if c unobserved

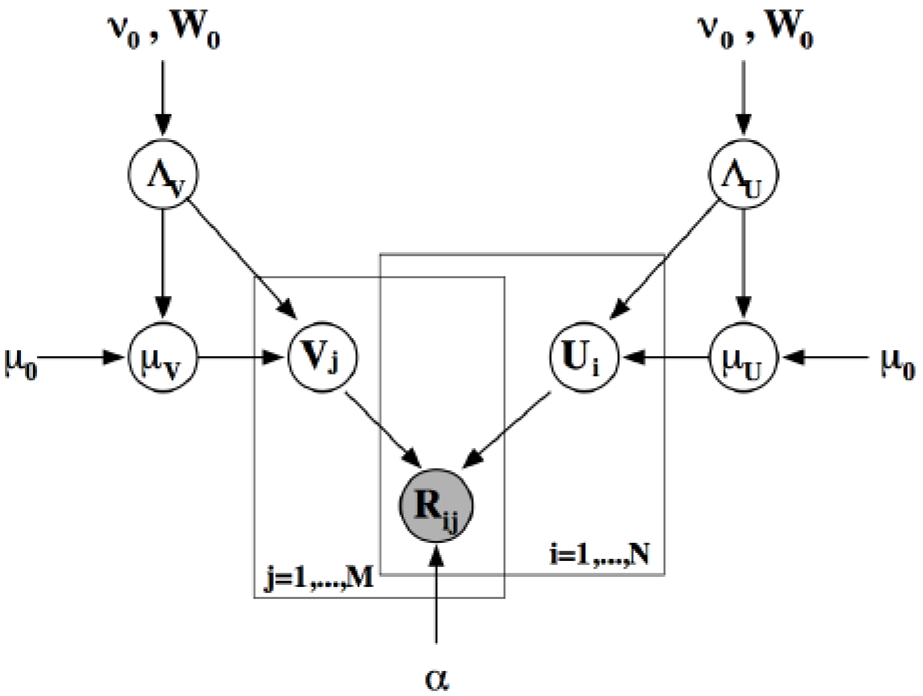


Popular Models

Latent Dirichlet Allocation



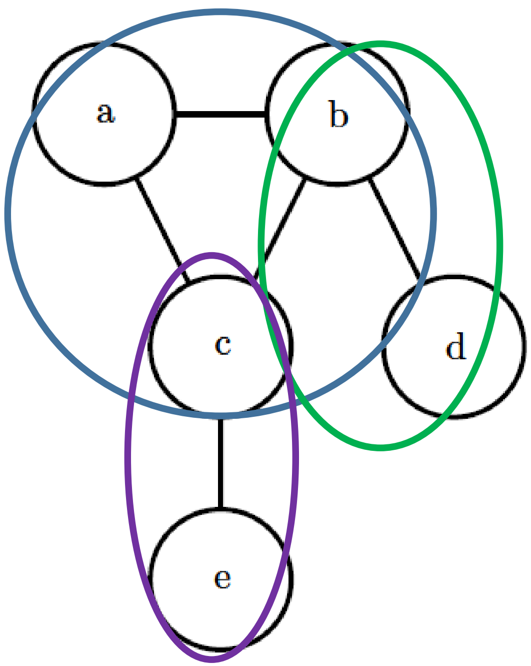
Bayesian Probabilistic Matrix Factorization



Undirected Graphical Models

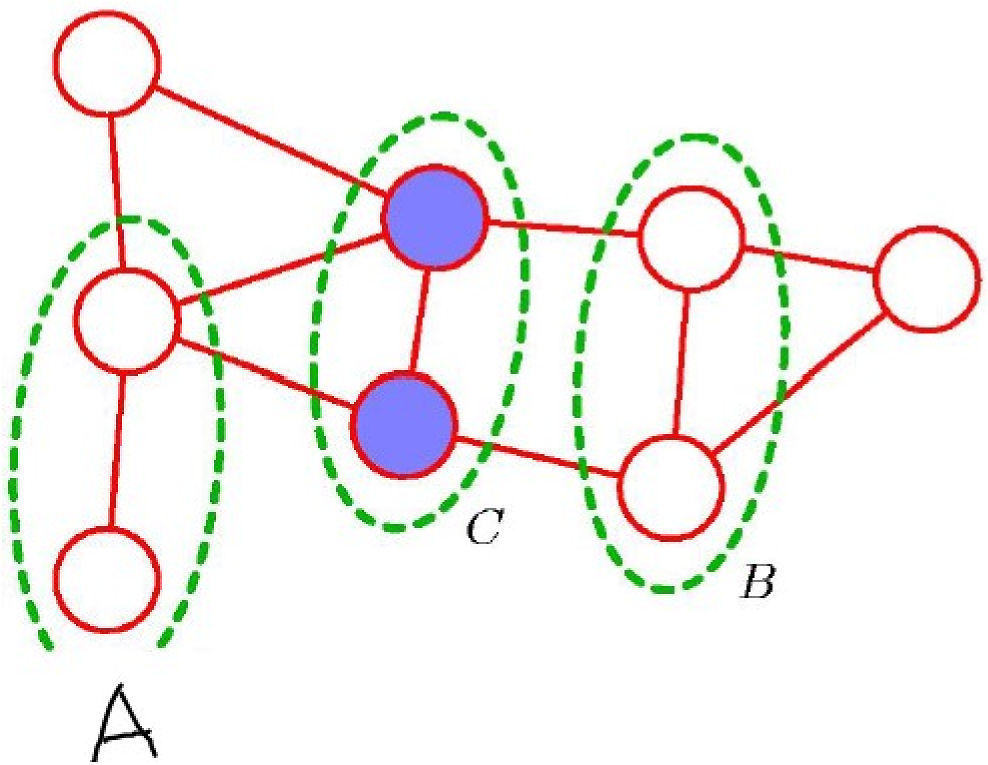
- Useful for expressing soft constraints between random variables

$$p(x) = \frac{1}{Z} \prod_i \phi^i(\mathcal{C}^i)$$



$$p(a, b, c, d, e) = \frac{1}{Z} \phi^1(a, b, c) \phi^2(b, d) \phi^3(c, e)$$

Conditional Independence



Compared to Directed Model

- Undirected model
  - Advantages
    - Wider applications
    - Succinctly express certain dependencies that directed model cannot easily describe
  - Disadvantages
    - Computing Z is NP-hard, need approximation
    - Difficult to interpret
    - Much easier to generate data from directed model (also called generated model)

Reading Material

- Koller, D. and Friedman, N., *Probabilistic graphical models: Principles and Techniques*, MIT Press
- Kevin Murphy (2013)., *Machine Learning: A Probabilistic Perspective.*

Summary

- Probability
  - Bayesian interpretation of probability
  - We care about conditional probability in real world
  - Important probability distribution
    - Bernoulli, binomial, multinoulli, multinomial, beta, dirichlet, Gaussian
      - Parameter estimation: MLE, MAP, etc.
    - Conjugate distribution: likelihood and prior
    - Sigmoid function, etc.
  - Entropy and cross-entropy
  - Structured graphical model
    - Directed vs. undirected
    - Casual relationships vs. dependency relationship
    - Conditional independency

Machine Learning Basics

Learning Algorithms

- An algorithm that is able to learn from data
- Mitchell (1997)
    - "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measures *P*, if its performance at tasks in *T*, as measured by *P*, improved with experience *E*."

Framework

Sentiment analysis

$f(\text{"I love the restaurant"}) = \text{"+"}$ (positive)

A set of function

Model

$f_1, f_2 \cdots$

Goodness of function f

Training Data

$f1(\text{"I love the restaurant"}) = \text{"+"}$

$f2(\text{"I love the restaurant"}) = \text{"−"}$

Better!

$f1(\text{"the quality is bad"}) = \text{"−"}$

$f2(\text{"the quality is bad"}) = \text{"+"}$

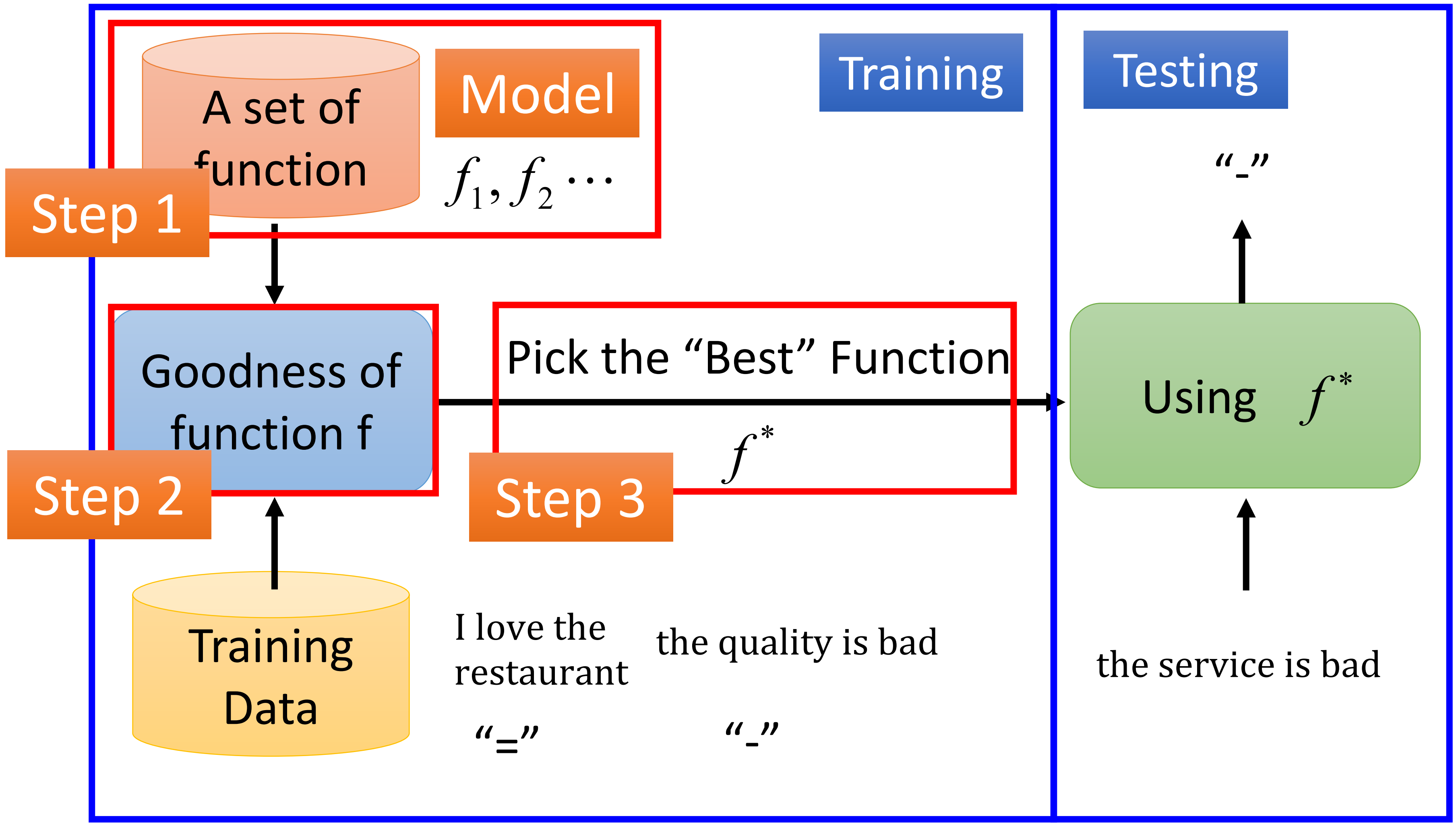
Supervised Learning

function input:	I love the restaurant	the quality is bad
function output:	“+”	“-”

Framework

Sentiment analysis:

f("I love the restaurant") = "+" (positive)



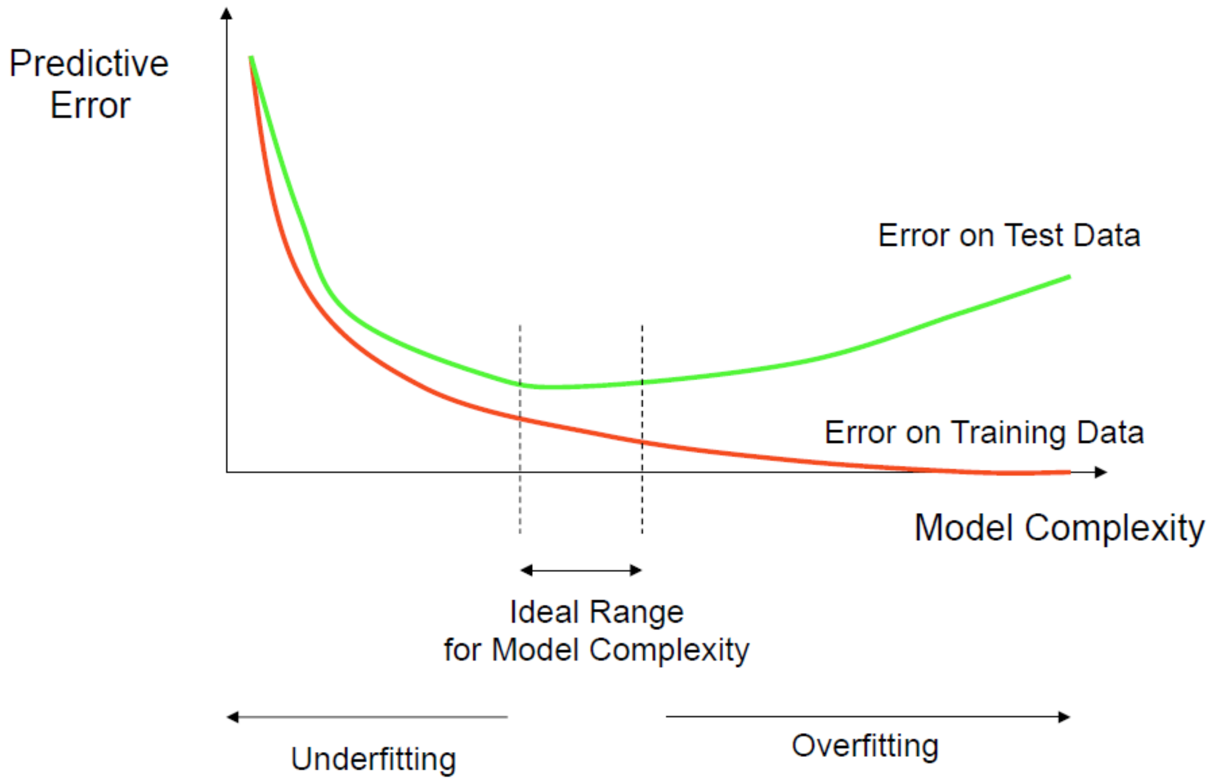
Three Steps for Machine Learning

Step 1: define a set of function	→	Step 2: goodness of function	→	Step 3: pick the best function
model		loss function		optimization

Capacity, Overfitting and Underfitting

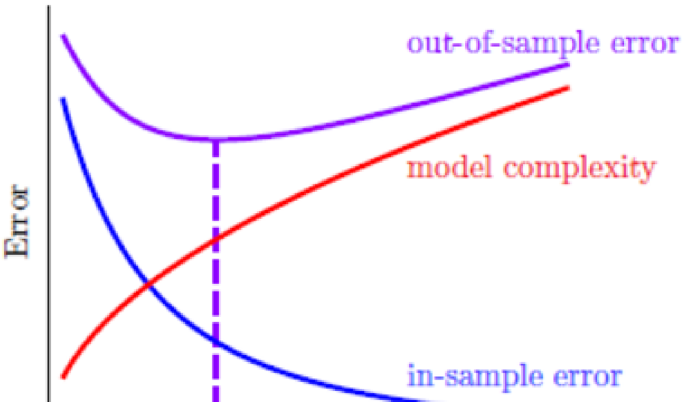
- Generalization
  - The ability to perform well on previously unobserved inputs (i.e. out-of-sample)
- Data generating process
  - $i.i.d.$ assumptions = independently and identically distributed
  - Data-generating distribution, p_{data}
  - Expected [Generalization error (or test error)] = Expected (training error)
- Goal of ML algorithms
  - Make the training error small
    - If not, underfitting
  - Make the gap between training and test error small
    - If not, overfitting

How Overfitting affects Prediction

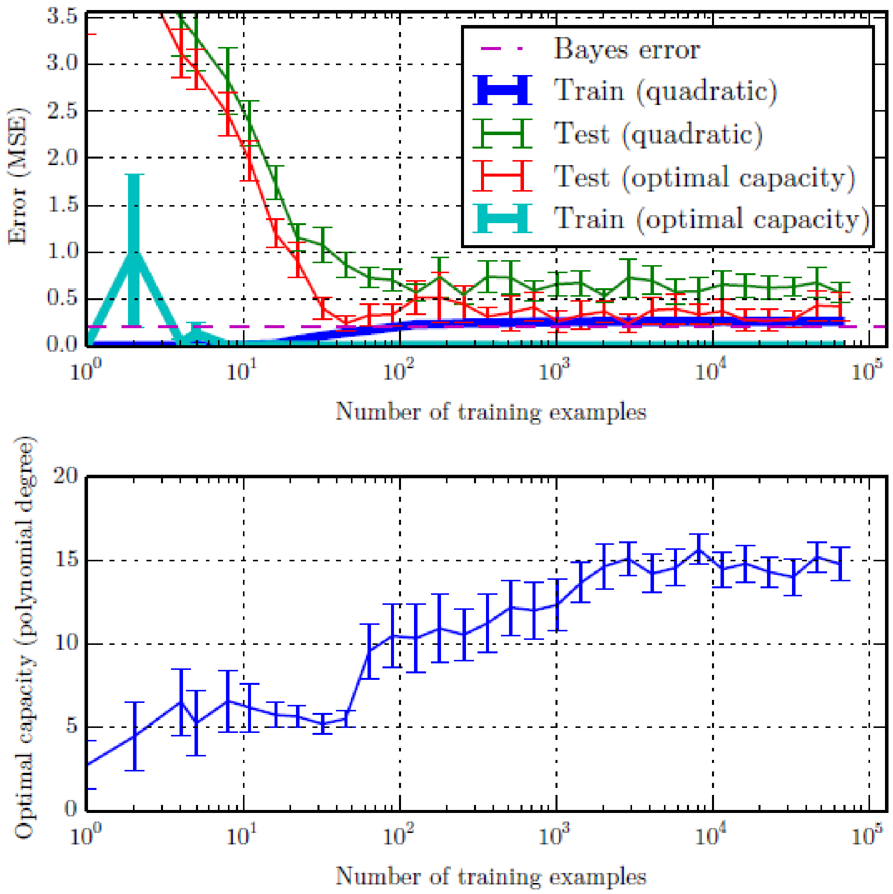


Capacity

- A model's ability to fit a wide variety of functions
- Ways to control the capacity
  - Hypothesis space (input features)
  - The model
    - Representation capacity vs. effective capacity
    - Occam's razor
      - Quantifying model capacity (VC dimension)
    - Nonparametric vs. parametric
  - Size of the training set



Training Data Size



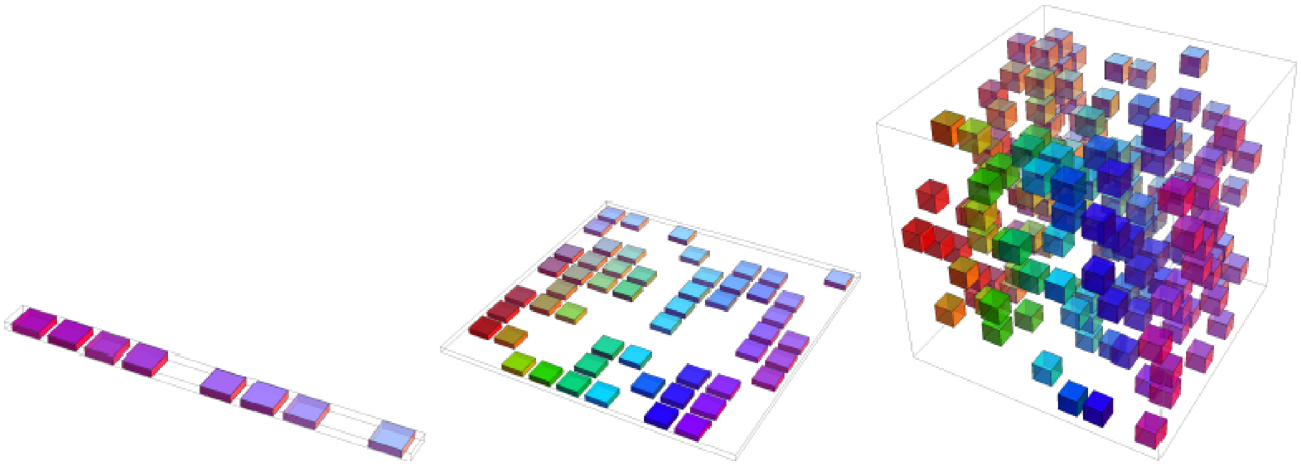
No free Lunch Theorem

- No machine learning algorithm is universally better than any other
  - The most sophisticated algorithm has the same average performance (over all possible tasks) as merely predicting that every point belongs to the same class
  - Goal of real ML research is to understand the mapping of ML algorithms to data generating distributions

Challenges Motivating Deep Learning

The Curse of Dimensionality

- ML learning becomes exceedingly difficult when the number of dimensions in the data is high
  - Statistical challenge



  - Arose the smoothness assumption

Local Constancy and Smoothness Regularization

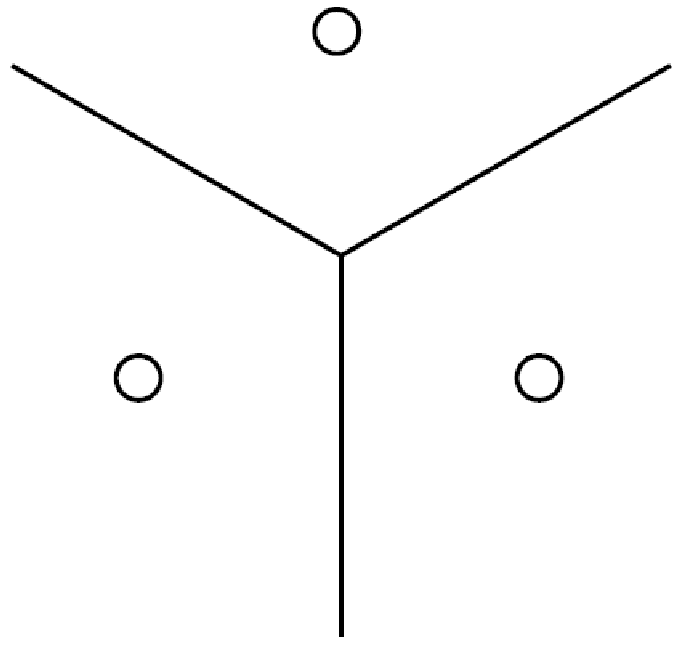
- Local constancy prior: Learnt function should keep stable within a small region

$$f^*(x) \approx f^*(x + \epsilon)$$

- Many simpler algorithms rely exclusively on the local constancy prior to generalize well
    - fail to scale to the statistical challenges in AI-level tasks
        - E.g. KNN, decision tree

Break Input Space Into Regions

Nearest Neighbor

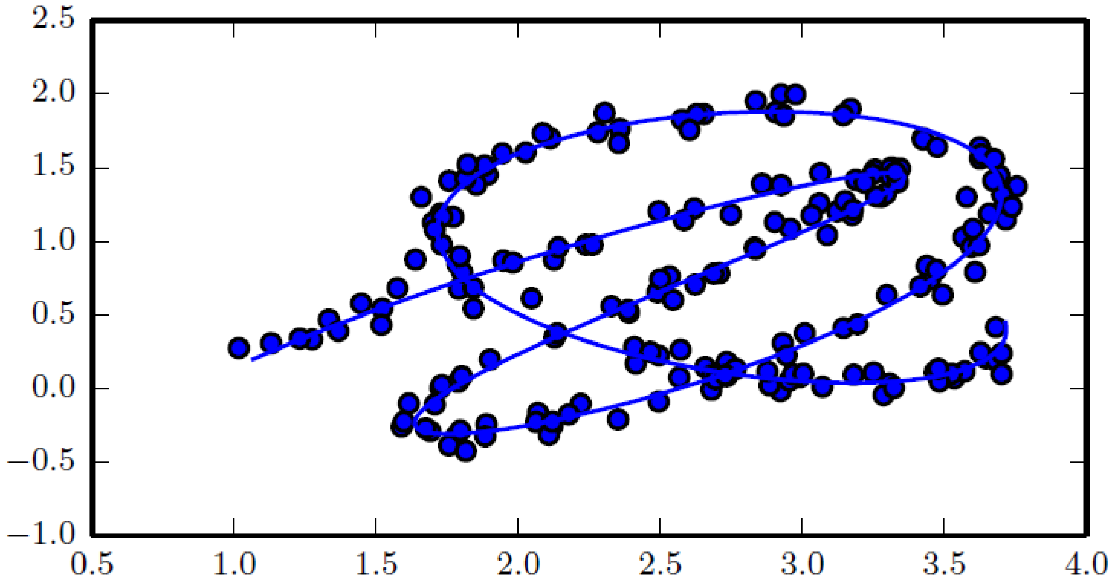


Local Constancy and Smoothness Regularization

- To answer two questions
  - Whether possible to represent a complicated function efficiently?
  - Whether possible to generalize well to new inputs?
- Solutions
  - Introduce dependencies among regions
    - DL methods DO without stronger task specific assumptions: exponential gain

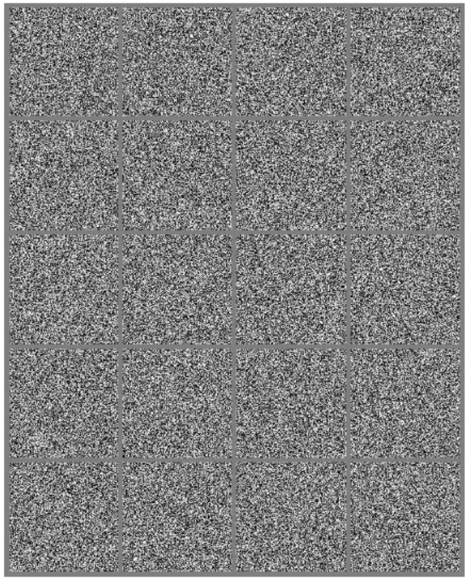
Manifold Learning

- Manifold assumption
  - Most of $\mathbb{R}^n$ consists of invalid inputs
  - Interesting variations happen only when move from one manifold to another
  - The data lies along a low-dimensional manifold



Manifold Learning

- Images, sounds and text strings are highly concentrated, and in favor of manifold hypothesis
  - Represent data in terms of coordinates on the manifold
- Manifold transformations are imaginably possible



Manifold Learning

- Extracting manifolds is challenging but promising
  - E.g. textbook section 20.10.4

Reading Materials

- Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer Publisher, 2006