Introduction to machine learning

Yifeng Tao

School of Computer Science

Carnegie Mellon University

Slides adapted from Tom Mitchell, Eric Xing, Barnabas Poczos

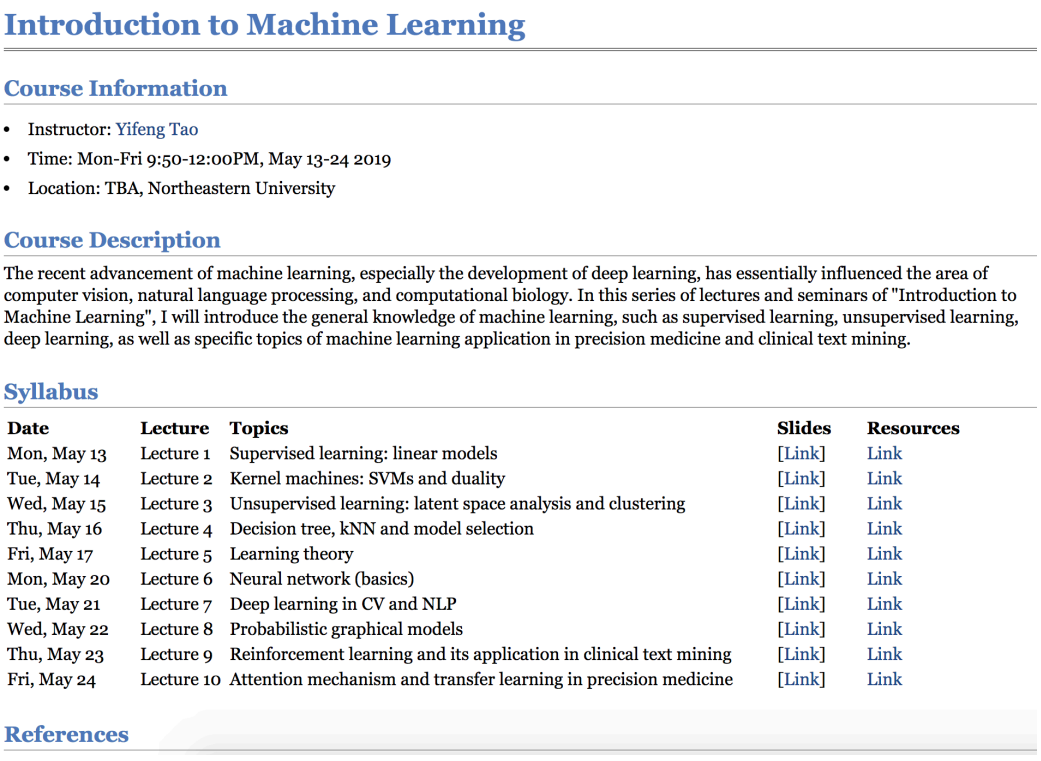
Logistics

- Course website:

http://www.cs.cmu.edu/~yifengt/courses/machine-learning

Slides uploaded after lecture

- Time: Mon-Fri 9:50-11:30am lecture, 11:30-12:00pm discussion
- Contact: yifengt@cs.cmu.edu



Introduction to Machine Learning

Course Information

- Instructor: Yifeng Tao
- Time: Mon-Fri 9:50-12:00PM, May 13-24 2019
- Location: TBA, Northeastern University

Course Description

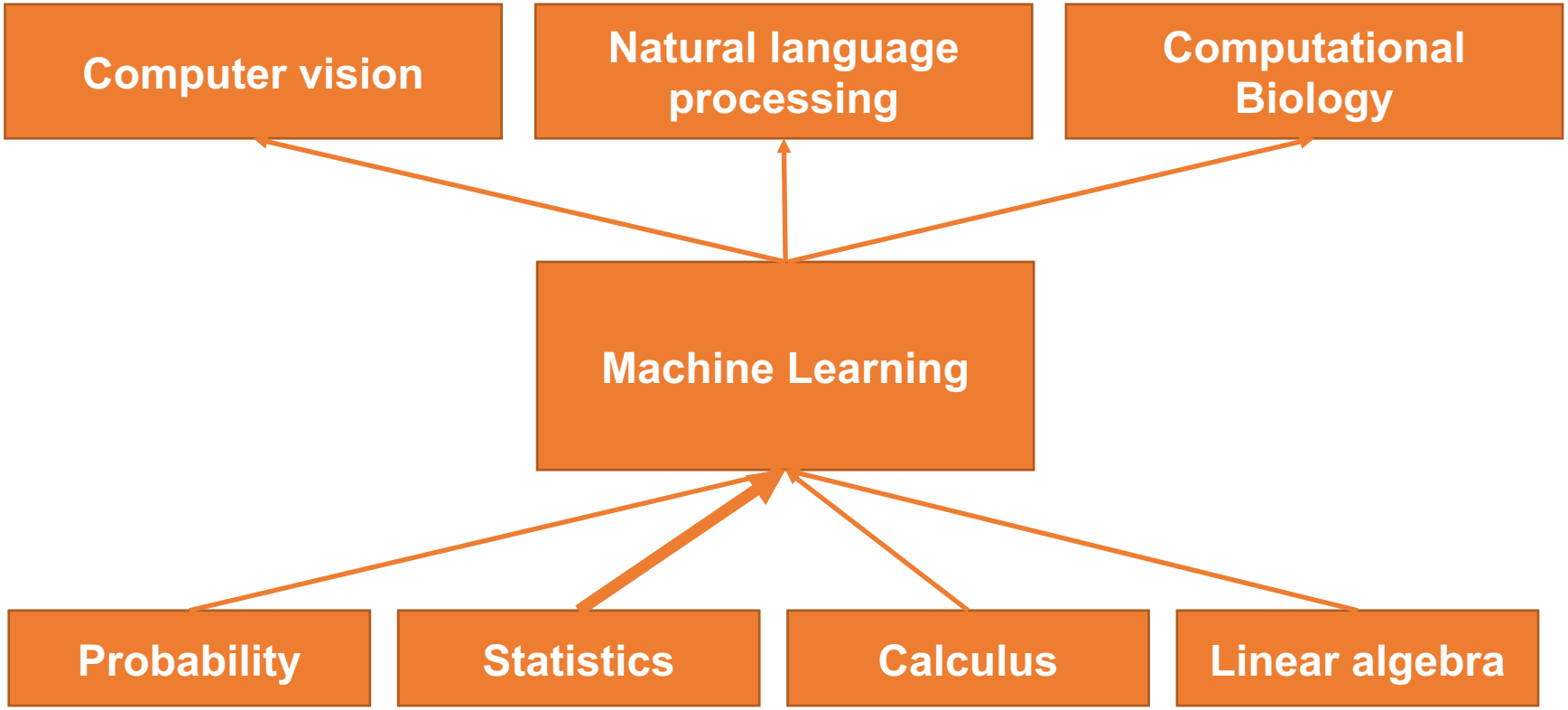
The recent advancement of machine learning, especially the development of deep learning, has essentially influenced the area of computer vision, natural language processing, and computational biology. In this series of lectures and seminars of "Introduction to Machine Learning", I will introduce the general knowledge of machine learning, such as supervised learning, unsupervised learning, deep learning, as well as specific topics of machine learning application in precision medicine and clinical text mining.

Syllabus

Date	Lecture	Topics	Slides	Resources
Mon, May 13	Lecture 1	Supervised learning: linear models	[Link]	Link
Tue, May 14	Lecture 2	Kernel machines: SVMs and duality	[Link]	Link
Wed, May 15	Lecture 3	Unsupervised learning: latent space analysis and clustering	[Link]	Link
Thu, May 16	Lecture 4	Decision tree, kNN and model selection	[Link]	Link
Fri, May 17	Lecture 5	Learning theory	[Link]	Link
Mon, May 20	Lecture 6	Neural network (basics)	[Link]	Link
Tue, May 21	Lecture 7	Deep learning in CV and NLP	[Link]	Link
Wed, May 22	Lecture 8	Probabilistic graphical models	[Link]	Link
Thu, May 23	Lecture 9	Reinforcement learning and its application in clinical text mining	[Link]	Link
Fri, May 24	Lecture 10	Attention mechanism and transfer learning in precision medicine	[Link]	Link

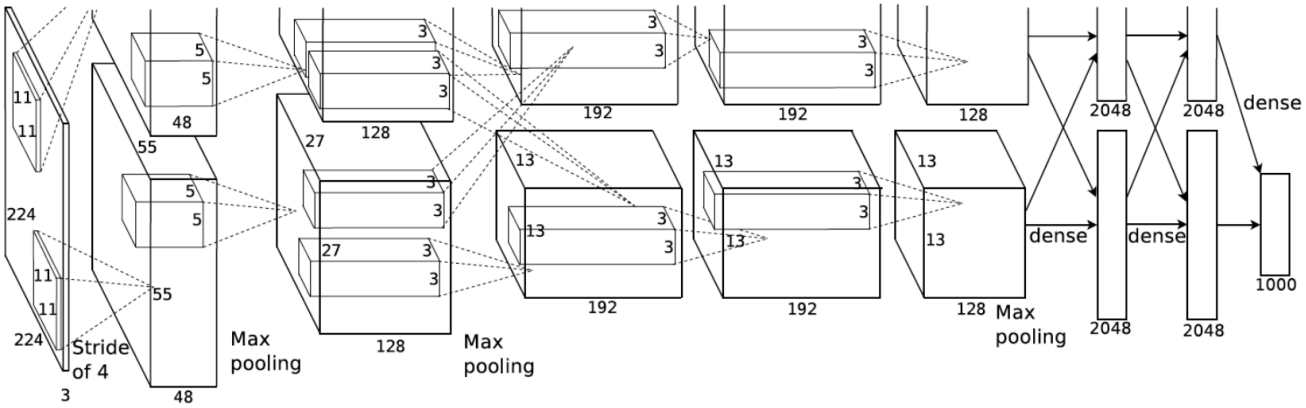
References

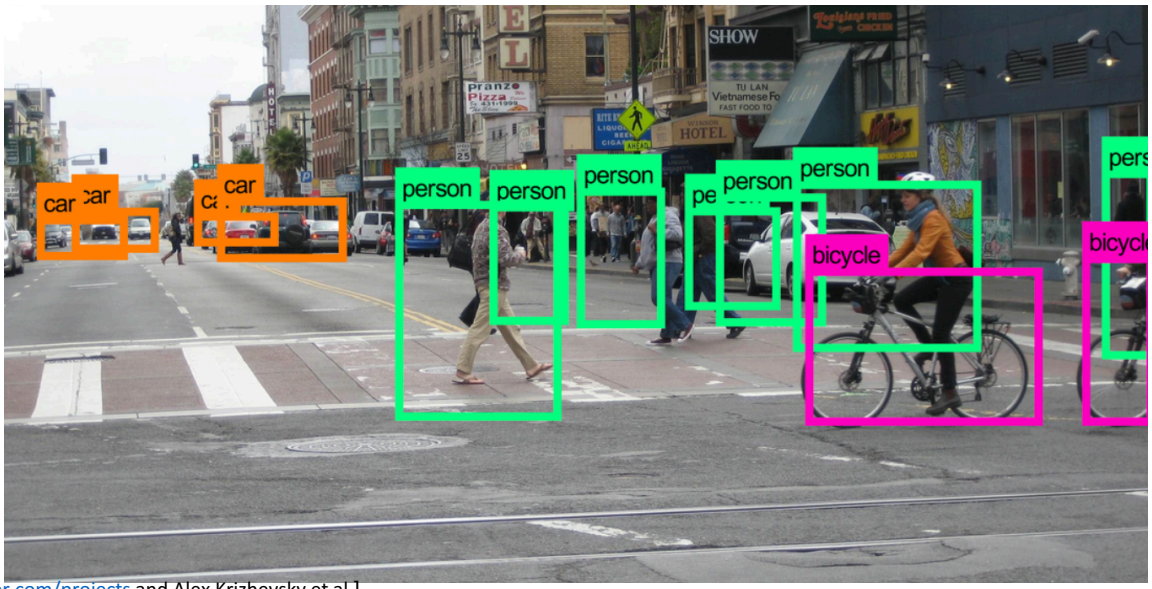
What is machine learning



Computer vision

- Object detection

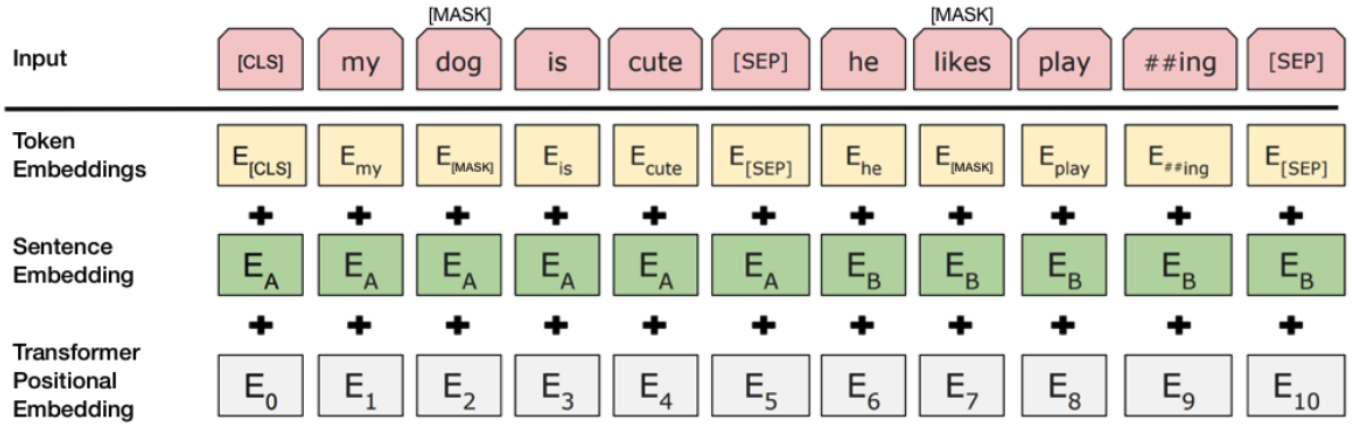




[Figure from https://www.cvdeveloper.com/projects and Alex Krizhevsky et al.]

Natural language processing

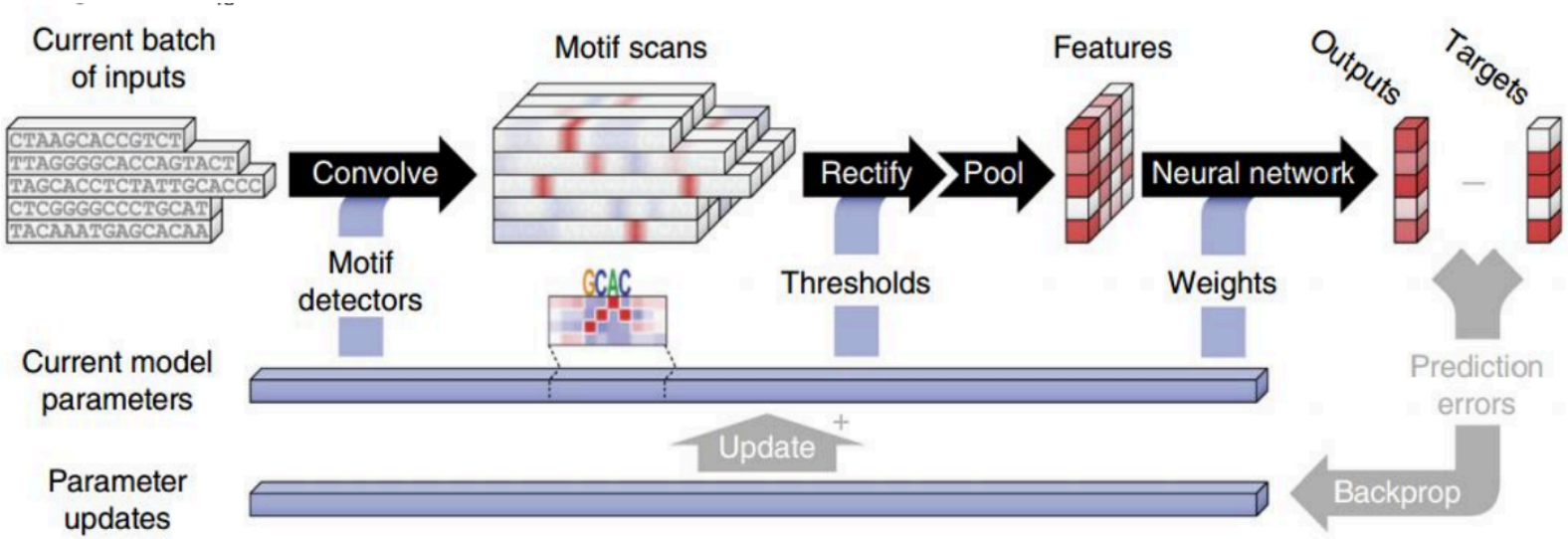
- NER, translation, document classification…



[Figure from Jacob Devlin et al.]

Computational Biology

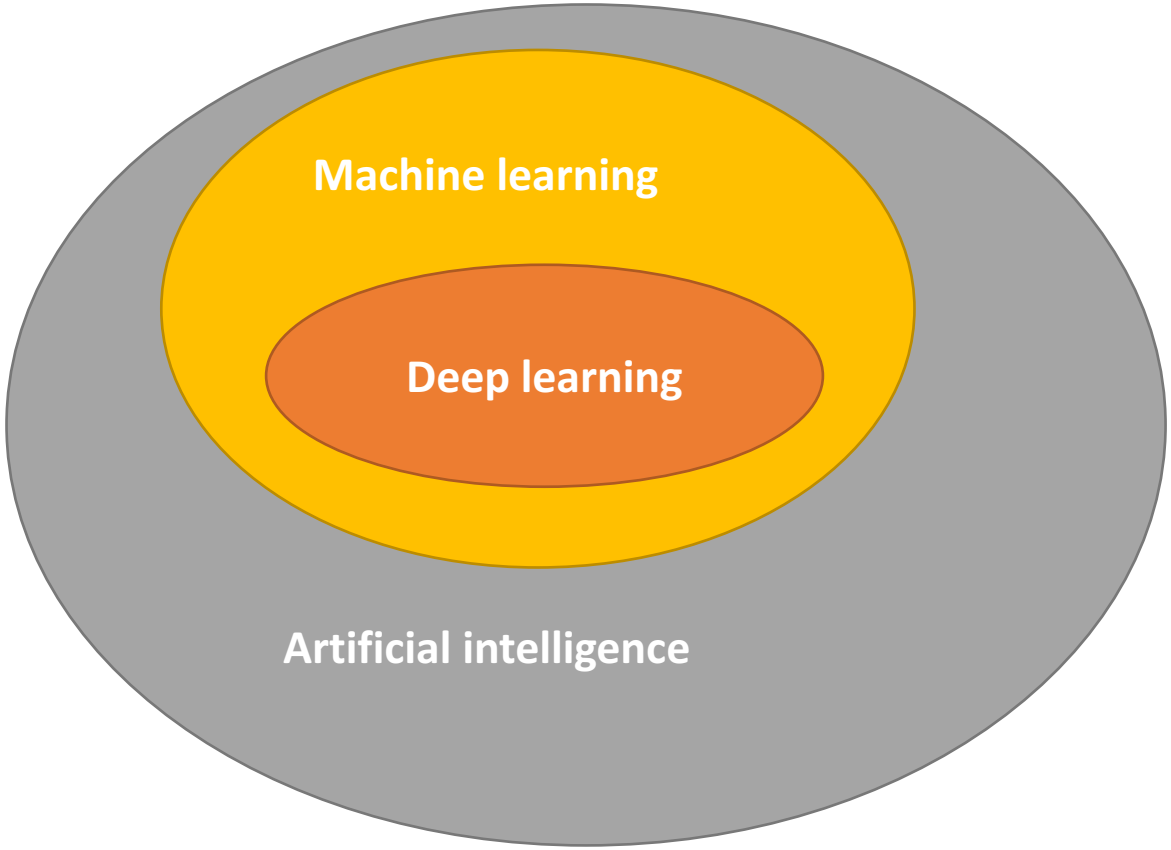
- DNA-protein binding



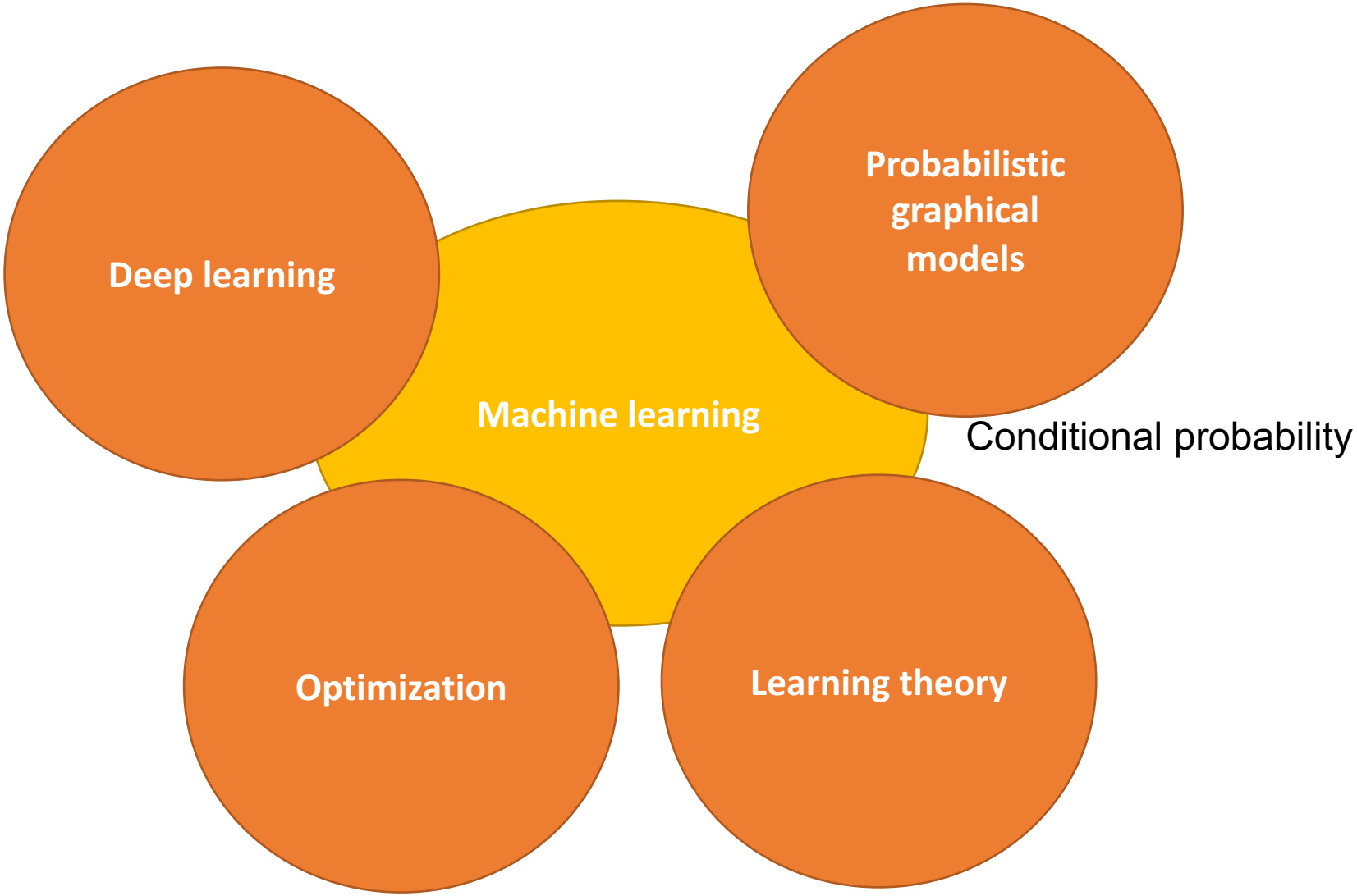
[Figure from Haoyang Zeng et al.]

What is machine learning?

- What are we talking when we talk about AI and ML?



What's more after introduction?



What is machine learning

- Methods that can help generalize information from the observed data so that it can be used to make better decisions in the future.
- Supervised learning: given a set of features and values

$$D = \{(X_i, y_i)\}_{i=1}^{n}$$

  Learn a model $M(\theta)$ that will predict a label to a new feature set.
  - Regression: predict continuous values
  - Classification: predict discrete labels
- Unsupervised learning: Discover patterns in data

$$D = \{X_i\}_{i=1}^{n}$$

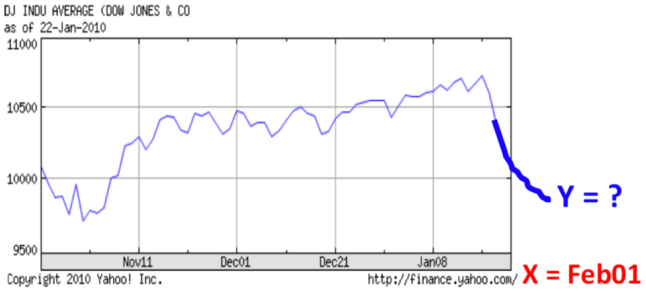
- And more! E.g., transfer learning, semi-supervised learning, reinforcement learning etc.

[Slide from Eric Xing et al.]

Supervised learning

- Goal of supervised learning: Construct a predictor $f : X \to Y$ to minimize a risk (performance measure) $R(f)$

Sports
Science
News



Classification:

$$R(f) = P(f(X) \neq Y)$$

Probability of Error

Regression:

$$R(f) = \mathbb{E}[(f(X) - Y)^2]$$

Mean Squared Error

- Not minimizing empirical errors:

$$\sum_{i=1}^{n} I(f(X_i) \neq Y_i)/n$$

$$\sum_{i=1}^{n} (f(X_i) - Y_i)^2/n$$

- Training and test sets

[Slide from Barnabas Poczos et al.]

Topics

- Supervised learning: linear models
- Kernel machines: SVMs and duality
- Unsupervised learning: latent space analysis and clustering
- Supervised learning: decision tree, kNN and model selection
- Learning theory
- Neural network (basics)
- Deep learning in CV and NLP
- Probabilistic graphical models

- Reinforcement learning and its application in clinical text mining
- Attention mechanism and transfer learning in precision medicine

Supervised learning: linear models

Yifeng Tao

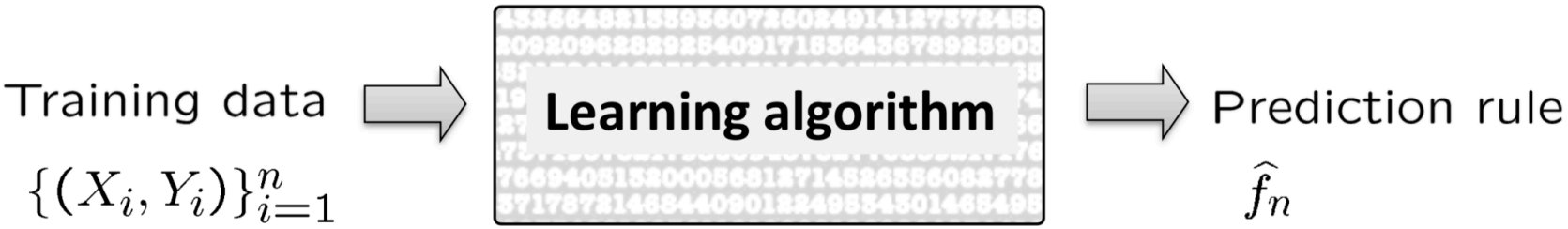
Lecture 1

May 13, 2019

Example of regression

- Predicting reviews of restaurant from factors

	$X_i^{(1)}$	$X_i^{(2)}$	$X_i^{(3)}$	Y_i
i	**Price**	**Distance**	**Cuisine**	**Review**
1	30	21	7	4
2	15	12	8	2
3	27	53	9	5



[Slide from Barnabas Poczos et al.]

Empirical Risk Minimization (ERM)

Optimal predictor: $$f^* = \arg\min_f \mathbb{E}[(f(X) - Y)^2]$$

Empirical Risk Minimizer: $$\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

Class of predictors — Empirical mean

$$\frac{1}{n} \sum_{i=1}^{n} [\text{loss}(Y_i, f(X_i))] \xrightarrow{\text{Law of Large Numbers}} \mathbb{E}_{XY} [\text{loss}(Y, f(X))]$$

- More in the learning theory part…

[Slide from Barnabas Poczos et al.]

Linear Regression

$$\hat{f}_n^L = \arg\min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

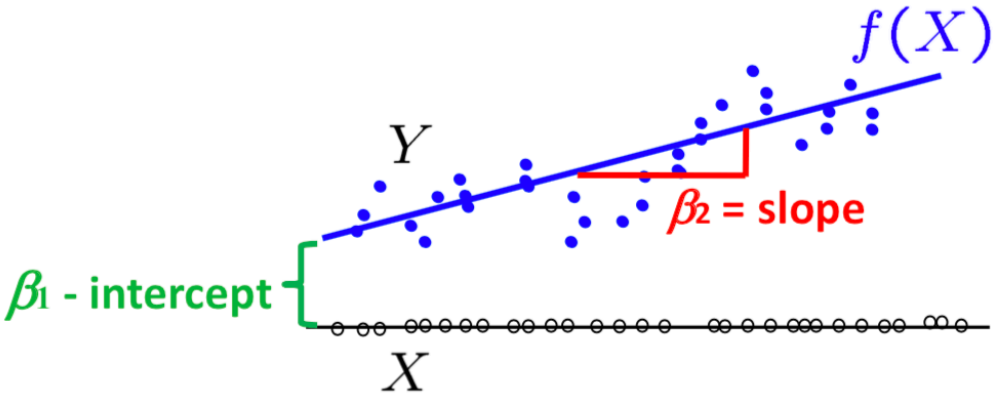
Least Squares Estimator

$\mathcal{F}_L$ - Class of Linear functions

Uni-variate case:

$$f(X) = \beta_1 + \beta_2 X$$

$$\hat{Y}_j = f(X_j) = \beta_1 + \beta_2 X_j$$



[Slide from Barnabas Poczos et al.]

Linear Regression

Multi-variate case: p variables.

$$f(X^{(1)}, \ldots, X^{(p)}) = \beta_1 X^{(1)} + \beta_2 X^{(2)} + \cdots + \beta_p X^{(p)} \in \mathbb{R}$$

$$= X\beta \in \mathbb{R}, \text{ where } X \in \mathbb{R}^{1\times p},\ \beta \in \mathbb{R}^p$$

Training set: n p-dimensional points. Let $X^{(j)} \in \mathbb{R}^n$

Prediction at the training point i:

$$\hat{Y}_i = f(X_i^{(1)}, \ldots, X_i^{(p)}) = \beta_1 X_i^{(1)} + \beta_2 X_i^{(2)} + \cdots + \beta_p X_i^{(p)} \in \mathbb{R}$$

$$\hat{Y} = \mathbf{A}\beta \in \mathbb{R}^n, \text{ where } \mathbf{A} = [X^{(1)} \ldots X^{(p)}] \in \mathbb{R}^{n\times p},$$

***A* = Design matrix** $\beta = [\beta_1 \ldots \beta_p]^T \in \mathbb{R}^p$

[Slide from Barnabas Poczos et al.]

Least Squares Estimator

$$\widehat{f}_n^L = \arg\min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 \qquad X_i \in \mathbb{R}^{1 \times p},\ Y_i \in \mathbb{R}$$

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (X_i\beta - Y_i)^2 \qquad \widehat{f}_n^L(X) = X\widehat{\beta}$$

$$= \arg\min_{\beta} \frac{1}{n} (\mathbf{A}\beta - \mathbf{Y})^T (\mathbf{A}\beta - \mathbf{Y})$$

$$\mathbf{A} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_1^{(1)} & \dots & X_1^{(p)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} & \dots & X_n^{(p)} \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$$

[Slide from Barnabas Poczos et al.]

Least Squares Estimator

$$J(\beta) = \frac{1}{n}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) \in \mathbb{R}$$

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{n}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) = \arg\min_{\beta} J(\beta)$$

$$\left.\frac{\partial J(\beta)}{\partial \beta}\right|_{\widehat{\beta}} = 0$$

$$\Rightarrow \mathbf{A}^T(\mathbf{A}\beta - \mathbf{Y}) = 0$$

[Slide from Barnabas Poczos et al.]

Normal Equations

$$(\mathbf{A}^T\mathbf{A})\hat{\beta} = \mathbf{A}^T\mathbf{Y}$$

p xp p x1 p x1

If $(\mathbf{A}^T\mathbf{A})$ is invertible,

$$\hat{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} \qquad \hat{f}_n^L(X) = X\hat{\beta}, \text{ where } X \in \mathbb{R}^{1\times p}$$

When is $(\mathbf{A}^T\mathbf{A})$ invertible ?

Recall: Full rank matrices are invertible. What is rank of $(\mathbf{A}^T\mathbf{A})$?

Theorem: $rank(A) = rank(A^TA) = rank(AA^T)$

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?

Regularization (later)

[Slide from Barnabas Poczos et al.]

Cases A^TA not invertible

Theorem: $rank(A) = rank(A^T A) = rank(AA^T)$

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?
Regularization (later)

- Gene expression data:
  - n=20,000, p=50-4,000
  - Regularization: Lasso

[Slide from Barnabas Poczos et al.]

Geometric Interpretation

Let $X \in \mathbb{R}^{1\times p}$. The linear prediction in this point X:

$$\hat{f}_n^L(X) = X\hat{\beta} = X(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} \in \mathbb{R}$$

The predictions in every point of the training set:

$$\hat{f}_n^L(\mathbf{A}^T) = \mathbf{A}\hat{\beta} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} \in \mathbb{R}^n \Rightarrow \hat{f}_n^L(\mathbf{A}^T) \in colspace(A)$$

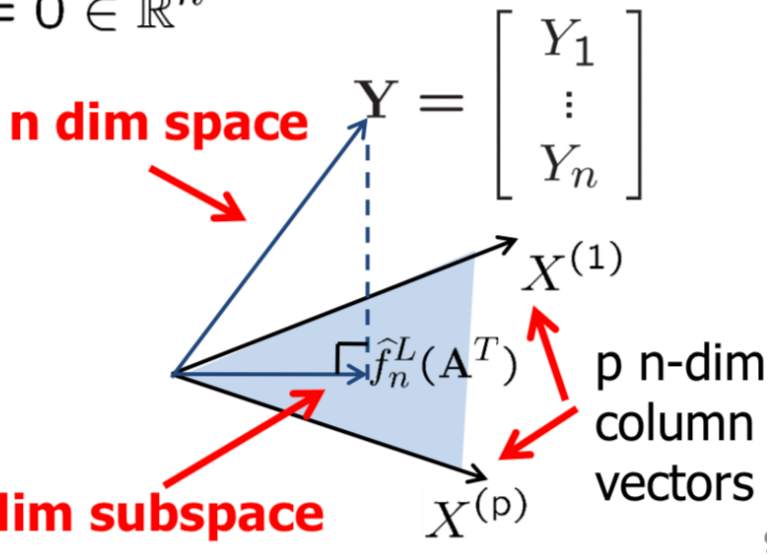
Difference between predicted values and true values on the training set:

$$\hat{f}_n^L(\mathbf{A}^T) - \mathbf{Y} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} - \mathbf{Y}$$

$$\Rightarrow \mathbf{A}^T(\hat{f}_n^L(\mathbf{A}^T) - \mathbf{Y}) = 0 \in \mathbb{R}^n$$

$\hat{f}_n^L(\mathbf{A}^T)$ is the orthogonal projection of $\mathbf{Y}$ onto the linear subspace spanned by the p columns of $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_1^{(1)} & \dots & X_1^{(p)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} & \dots & X_n^{(p)} \end{bmatrix}$$



[Slide from Barnabas Poczos et al.]

Pseudo Inverse (skip)

For $\mathbf{A} \in \mathbb{R}^{m \times n}$, a pseudoinverse of $\mathbf{A}$ is defined as $\mathbf{A}^+ \in \mathbb{R}^{n \times m}$ satisfying:

$$AA^+A = A \qquad (AA^+)^T = AA^+$$
$$A^+AA^+ = A^+ \qquad (A^+A)^T = A^+A$$

A^+ exists for any matrix A, but when A has full rank, A^+ has a simple form.

Left inverse

When A has linearly independent columns $\Rightarrow$ A^TA is invertible $\Rightarrow$ $A^+ = (A^TA)^{-1}A^T$

(Left inverse, since $A^+A = I$.)

Right inverse

When A has linearly independent rows $\Rightarrow$ AA^T is invertible $\Rightarrow$ $A^+ = A^T(AA^T)^{-1}$

(Right inverse, since $AA^+ = I$.)

[Slide from Barnabas Poczos et al.]

Pseudo Inverse (skip)

The pseudo inverse provides a linear least squares solution to a system of linear equations:

$$\mathbf{A}\beta = \mathbf{Y},\ \mathbf{A} \in \mathbb{R}^{n \times p},\ \beta \in \mathbb{R}^{p},\ Y \in \mathbb{R}^{n}.$$

In general, a vector $^{-}$ that solves the system i) may not exist, ii) infinite many might exist, or iii) can be unique.

The pseudoinverse solves the "least-squares" problem as follows:

$$\forall \beta \in \mathbb{R}^{p}, \text{ we have } \|\mathbf{A}\beta - \mathbf{Y}\|_2 \geq \|\mathbf{A}z - \mathbf{Y}\|_2 \text{ where } z = \mathbf{A}^{+}\mathbf{Y}.$$

Equality holds iff $\beta = \mathbf{A}^{+}\mathbf{Y} + (I - \mathbf{A}^{+}\mathbf{A})w$, where w is arbitrary vector.

There are infinite many solutions unless $\mathbf{A}$ has full column rank, in which case $(I - \mathbf{A}^{+}\mathbf{A})$ is a zero matrix

[Slide from Barnabas Poczos et al.]

Polynomial Regression

Let $X \in \mathbb{R}$, $\beta_i \in \mathbb{R}$, $i = 0, 1, \ldots, m$.

Univariate (1-dim) case: $f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_m X^m = \mathbf{X}\beta$

where $\mathbf{X} = [1 \; X \; X^2 \ldots X^m], \beta = [\beta_0 \ldots \beta_m]^T$

Design matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & X_1 & X_1^2 & \cdots & X_1^m \\ \vdots & & & \ddots & \vdots \\ 1 & X_n & X_n^2 & \cdots & X_n^m \end{bmatrix}$$

$$\hat{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} \in \mathbb{R}^{m+1}$$

$$\mathbf{Y} \in \mathbb{R}^n$$

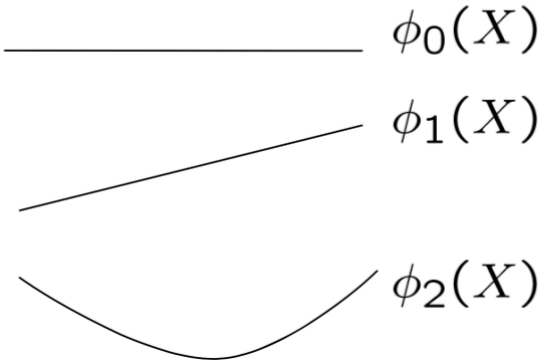
Prediction: $\hat{f}_n(X) = [1 \; X \; X^2 \ldots \; X^m]\hat{\beta} \in \mathbb{R}$

Linear combination of nonlinear features

$$f(X) = \sum_{j=0}^{m} \beta_j X^j = \sum_{j=0}^{m} \beta_j \phi_j(X)$$

Weight of each feature

Nonlinear features



[Slide from Barnabas Poczos et al.]

Maximum Likelihood Estimation (MLE)

- Goal: estimate distribution parameters θ from a dataset of n independent, identically distributed (i.i.d.), fully observed training cases:

$$D = \{(X_i, y_i)\}_{i=1}^n \qquad \boxed{D = \{X_i\}_{i=1}^n}$$

- Maximum Likelihood Estimation (MLE):
  - One of the most common estimators
  - With iid and fully-observability assumptions:

$$P(D;\theta) = P(X_1, ..., X_n;\theta) = \prod_{i=1}^{n} P(X_i;\theta)$$

  - Pick the setting of parameters **most likely to have generated the data** we saw:

$$\hat{\theta} = \arg\max_{\theta} P(D;\theta) = \arg\max_{\theta} \log P(D;\theta)$$

- Maximum conditional likelihood:

$$(X_i, y_i) \sim P(X, y;\beta) = P(X)P(y|X,\beta)$$
$$\hat{\theta} = \arg\max_{\theta} \log P(X_i, y_i;\theta) = \arg\max_{\theta} \log P(y_i|X_i,\theta)$$

[Slide from Eric Xing et al.]

Least Squares and MLE

Let $X \in \mathbb{R}^{1\times p},\ \beta^* \in \mathbb{R}^p.$

Intuition: Signal plus (zero-mean) Noise model

$$Y = f^*(X) + \epsilon = X\beta^* + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

$$\Rightarrow Y \sim \mathcal{N}(X\beta^*, \sigma^2 \mathbf{I})$$

Let $X_i \in \mathbb{R}^{1\times p},\ Y_i \in \mathbb{R},\ i = 1, \ldots, n,\ \beta \in \mathbb{R}^p.$

$$\widehat{\beta}_{\text{MLE}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^n | \beta, \sigma^2)}_{\text{log likelihood}}$$

$$= \arg\min_{\beta} \sum_{i=1}^{n} (X_i\beta - Y_i)^2 = \widehat{\beta}$$

Least Square Estimate is same as Maximum Likelihood Estimate under a Gaussian model !

[Slide from Barnabas Poczos et al.]

Least Squares and MLE

$$p(Y_i|X_i,\beta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(Y_i-\beta^T X_i)^2}{2\sigma^2}\right)$$

- By independence assumption:

$$L(\beta) = \prod_{i=1}^{n} p(Y_i|X_i,\beta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(Y_i-\beta^T X_i)^2}{2\sigma^2}\right)$$

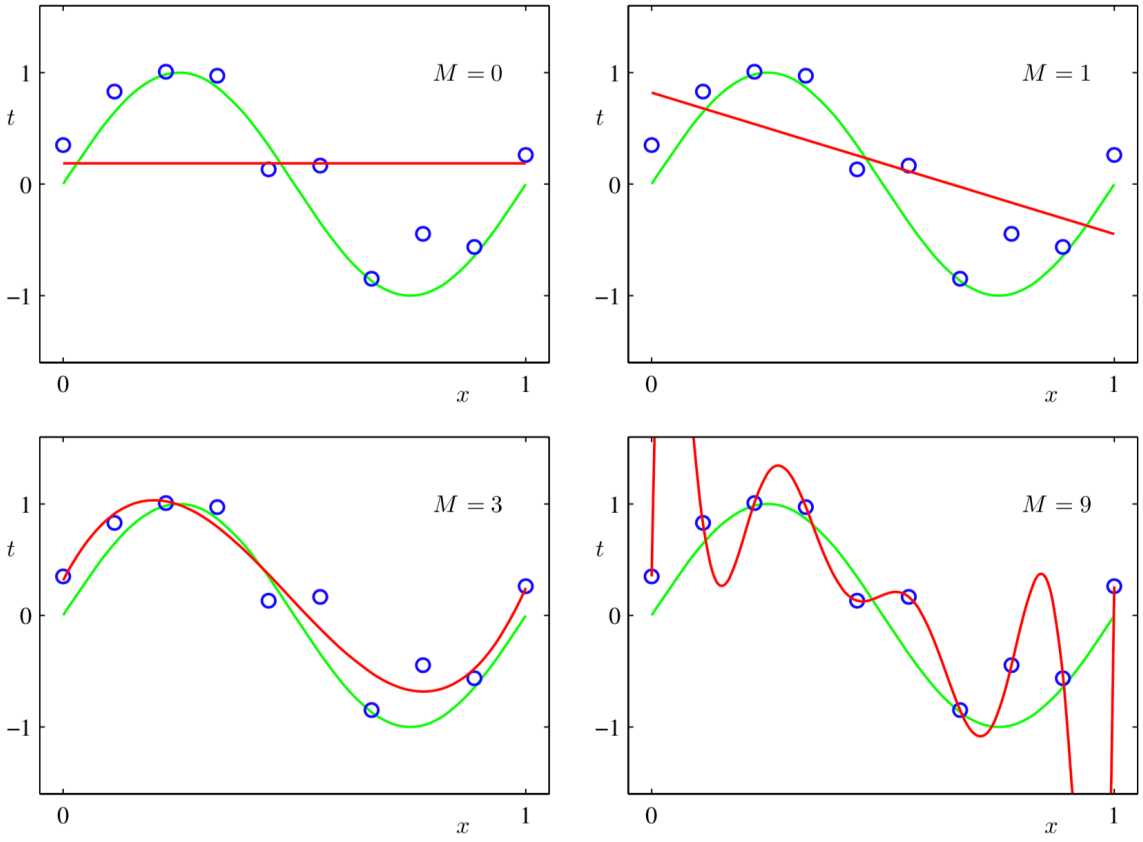
- Therefore,

$$l(\beta) = n\log\frac{1}{\sqrt{2\pi}\sigma} - \frac{\sum_{i=1}^{n}(Y_i-\beta^T X_i)^2}{2\sigma^2}$$

[Slide from Eric Xing et al.]

Regularized Least Squares

- Recap the polynomial regression $\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_M x^M$
- Intuition of overfitting: very large weights
- How to solve/alleviate it?



[Figure from Christopher M. Bishop]

Maximum a Posteriori Estimation (MAP)

- The Bayesian theory:

$$P(\theta; D) = \frac{P(D; \theta)P(\theta)}{P(D)} = \frac{P(D; \theta)P(\theta)}{\int_{\theta'} P(D; \theta')P(\theta')}$$

  - The posterior equals to the likelihood times the prior, up to a constant
- Maximum a posteriori estimator (MAP):

$$\hat{\theta} = \arg\max_{\theta} P(\theta; D) = \arg\max_{\theta} \left(\log P(D; \theta) + \log P(\theta)\right)$$

  - This allows us to capture uncertainty about the model in a principled way

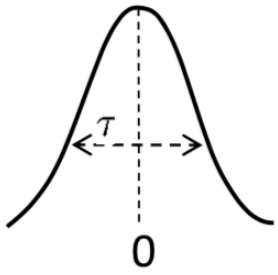
Regularized Least Squares and MAP

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?

$$\widehat{\beta}_{\text{MAP}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^n | \beta, \sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

I) Gaussian Prior

$$\beta \sim \mathcal{N}(0, \tau^2 \mathbf{I}) \qquad p(\beta) \propto e^{-\beta^T\beta/2\tau^2}$$



$$\widehat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda\|\beta\|_2^2$$

$\lambda \downarrow$ constant(σ^2, τ^2)

Ridge Regression

Prior belief that β is Gaussian with zero-mean biases solution to "small" β

[Slide from Barnabas Poczos et al.]

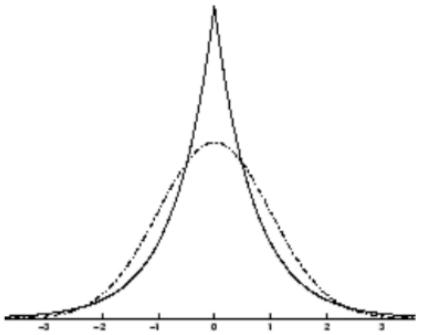
Regularized Least Squares and MAP

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?

$$\widehat{\beta}_{\text{MAP}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^n | \beta, \sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

II) Laplace Prior

$$\beta_i \overset{iid}{\sim} \text{Laplace}(0, t) \qquad p(\beta_i) \propto e^{-|\beta_i|/t}$$



$$\widehat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \|\beta\|_1$$

$\lambda \|\beta\|_1 \downarrow \text{constant}(\sigma^2, t)$

Lasso

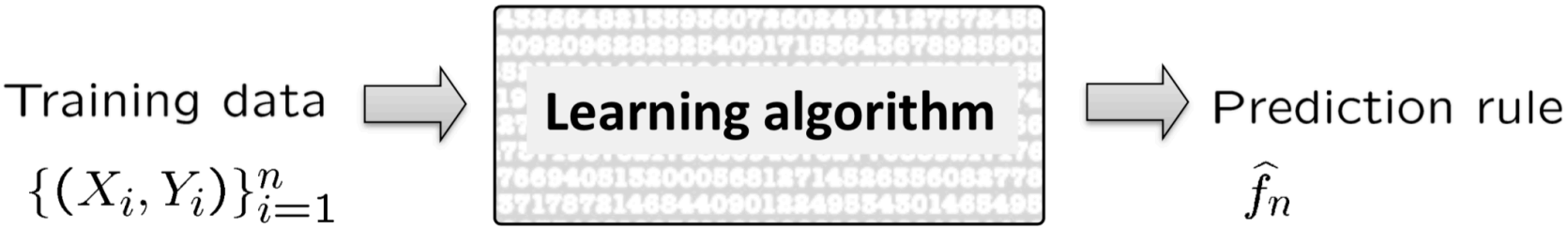
Prior belief that β is Laplace with zero-mean biases solution to "small" β

[Slide from Barnabas Poczos et al.]

Example of classification

- Predicting reviews of restaurant from features

	$X_i^{(1)}$	$X_i^{(2)}$	$X_i^{(3)}$	Y_i
i	**Price**	**Distance**	**Cuisine**	**Review**
1	30	21	7	Good
2	15	12	8	Bad
3	27	53	9	Good



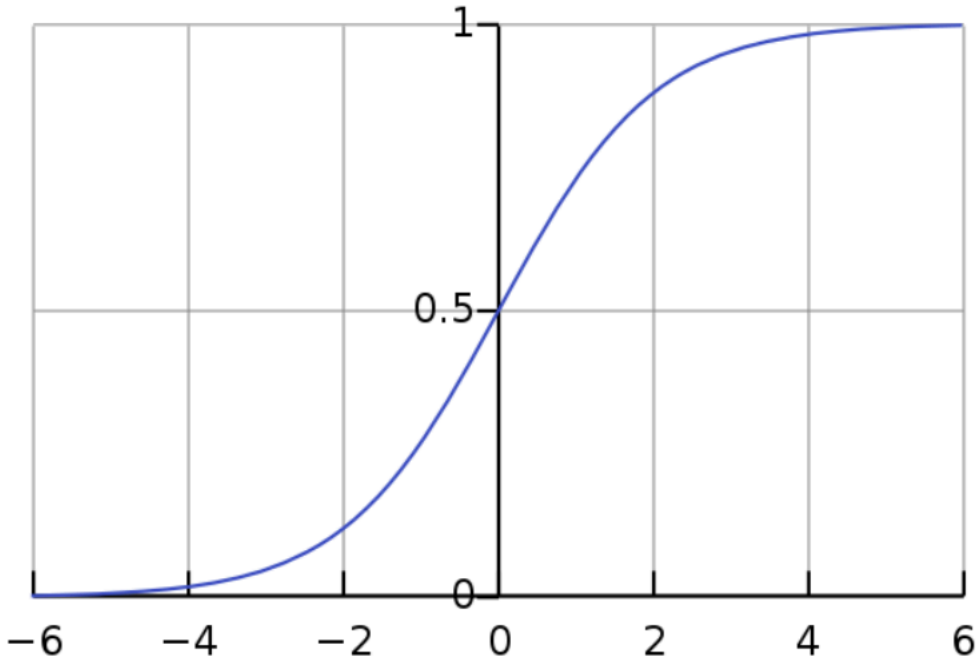
[Slide from Barnabas Poczos et al.]

Logistic regression

- Logistic/Sigmoid function:

$$p = \hat{y} = f(X;\beta) = \text{sigmoid}(\beta^T X) = \frac{1}{1+e^{-\beta^T X}}$$

  - *p*: the probability that y is 1



[Figure from Wikipedia]

Logistic regression: MLE

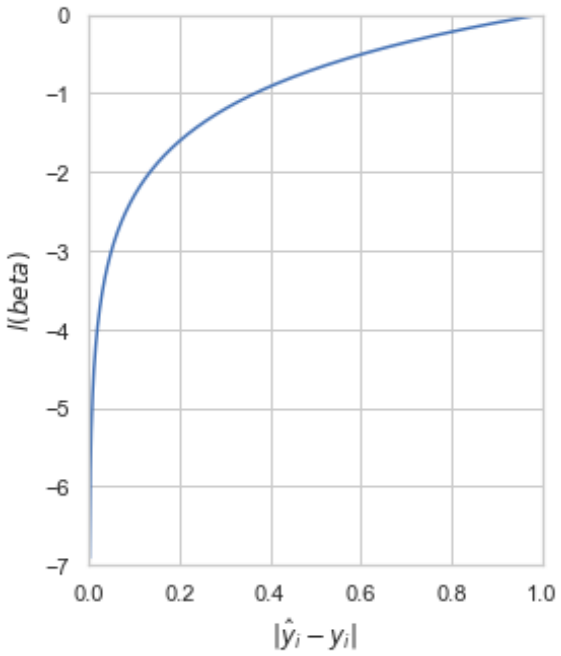
- The likelihood function is:

$$\min_{\beta} \prod_{i=1}^{n} P(y_i; X_i, \beta) = \prod_{i=1}^{n} (\hat{y}_i I(y_i = 1) + (1 - \hat{y}_i) I(y_i = 1)) = \prod_{i=1}^{n} \hat{y}_i^{y_i} (1 - \hat{y}_i)^{1 - y_i}$$

- Therefore, the conditional log-likelihood function:

$$l(\beta) = \ln \prod_{i=1}^{n} P(y_i | X_i, \beta) = \sum_{i=1}^{n} y_i \cdot \text{sigmoid}(\beta^T X_i) + (1 - y_i)(1 - \text{sigmoid}(\beta^T X_i))$$

  - The *-l(β)* is also referred to as "cross-entropy loss"
- Good new: *l(β)* is a concave function of *β*
- Bad news: no closed-form solution to maximize *l(β)*
  - Solution: optimization algorithm (to be discussed)



[Slide from Tom Mitchell et al.]

Logistic regression: MAP

- Gaussian prior of β:

$$l(\beta) = \lambda||\beta||_2^2 + \sum_{i=1}^{n} y_i \cdot \text{sigmoid}(\beta^T X_i) + (1 - y_i)(1 - \text{sigmoid}(\beta^T X_i))$$

- Laplacian prior of β:

$$l(\beta) = \lambda||\beta||_1 + \sum_{i=1}^{n} y_i \cdot \text{sigmoid}(\beta^T X_i) + (1 - y_i)(1 - \text{sigmoid}(\beta^T X_i))$$

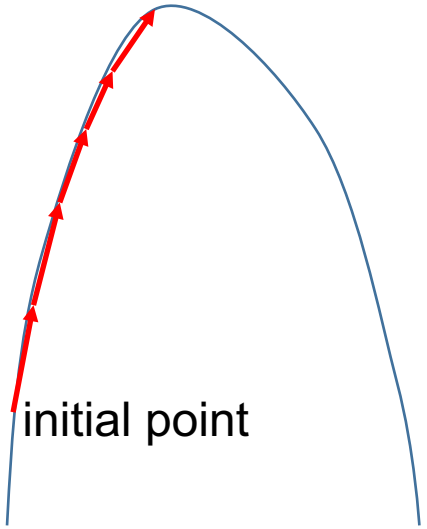
Maximize conditional likelihood: gradient ascent

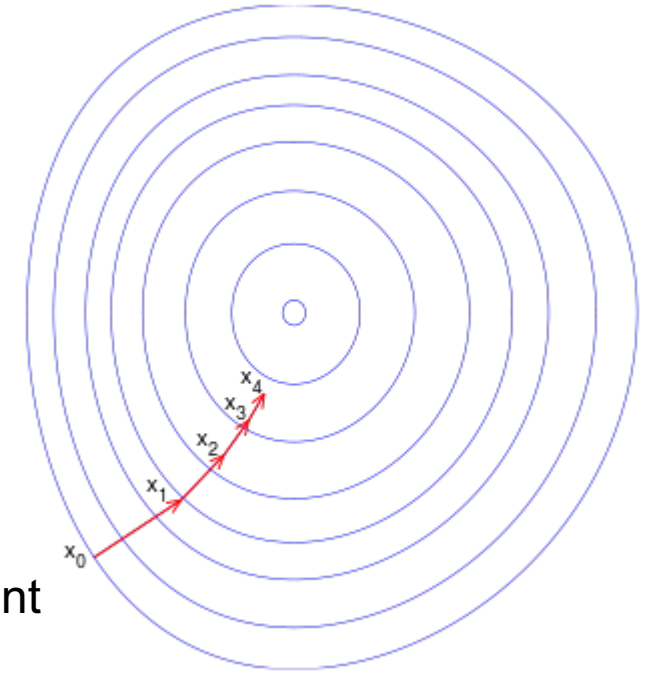
- Gradient ascent algorithm: iterate until change < ε:

$$\beta \leftarrow \beta + \eta \nabla l(\beta)$$

- This applies to linear regression as well, although there exist closed form.

$$\widehat{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y}$$





[Slide from Tom Mitchell et al.]

Bayesian classifier

	$X_i^{(1)}$	$X_i^{(2)}$	$X_i^{(3)}$	Y_i
i	**Price**	**Distance**	**Cuisine**	**Review**
1	30	21	7	Good
2	15	12	8	Bad
3	27	53	9	Good

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

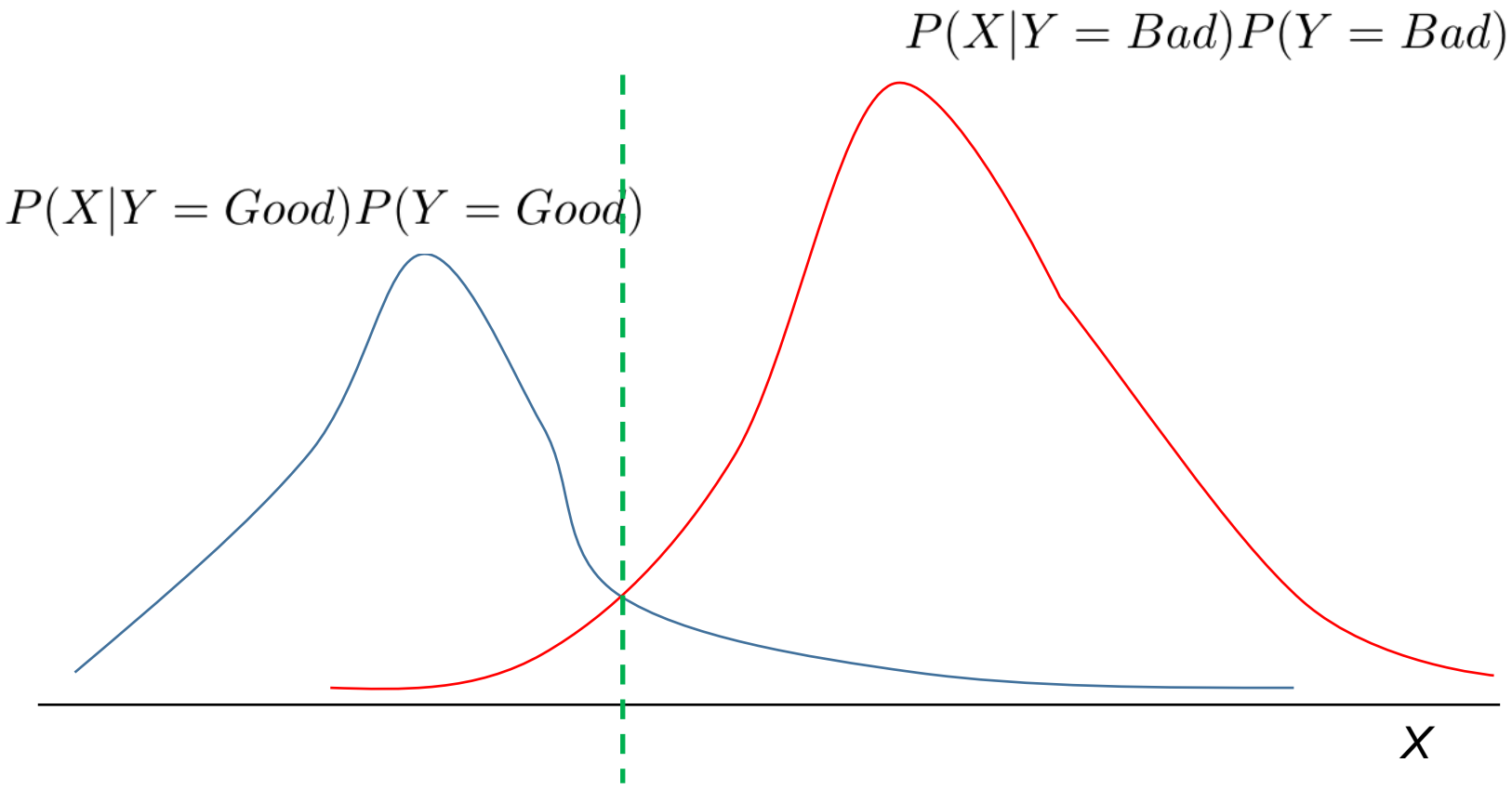
Learning = estimating P(X|Y), P(Y)

Classification = using Bayes rule to calculate $P(Y \mid X^{new})$

- Generative model vs discriminative model

[Slide from Tom Mitchell et al.]

Bayesian classifier



Naïve Bayes

- The Bayesian classifier requires large number of samples to train
- Naïve Bayes assumes:

$$P(X_1 \dots X_n|Y) = \prod_i P(X_i|Y)$$

- The X_i are conditionally independent, given Y.
- Therefore the classification rule for $X^{new}=(X_1, \dots, X_n)$ is

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new}|Y = y_k)$$

[Slide from Tom Mitchell et al.]

Naïve Bayes Algorithm

- Train Naïve Bayes (examples)

  for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \pi_k \prod_i \theta_{ijk}$$

* probabilities must sum to 1, so need estimate only n-1 parameters...

- Very fast to train/estimate!

[Slide from Tom Mitchell et al.]

Bag of words: model the documents

- 8th floor of Gates building, CMU

[Figure from https://twitter.com/smithamilli/status/837153616116985856]

Document classification

- The (independent) probability that the i-th word of a given document occurs in a document from class *C:*

$$p(w_i \mid C)$$

- The probability that a given document *D* contains all of the words given a class *C*:

$$p(D \mid C) = \prod_i p(w_i \mid C)$$

- What is the probability that a given document *D* belongs to a given class *C*?

$$p(C \mid D) = \frac{p(C)\, p(D \mid C)}{p(D)}$$

[Slide from Wikipedia]

Continuous X_i in Naïve Bayes

Eg., image classification: X_i is ith pixel

Gaussian Naïve Bayes (GNB): assume

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

Sometimes assume variance

- is independent of Y (i.e., σ_i),
- or independent of X_i (i.e., σ_k)
- or both (i.e., σ)

[Slide from Tom Mitchell et al.]

Estimating parameters of GNB

- Y discrete, X_i continuous

Maximum likelihood estimates:

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

jth training example

ith feature

kth class

δ(x)=1 if x true, else 0

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

[Slide from Tom Mitchell et al.]

Inference of Gaussian Naïve Bayes

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, $< X_1 \ldots X_n >$
  - Y is boolean
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik},\sigma_i)$
  - model P(Y) as Bernoulli (π)

Assume σ_{ik} indep of Y

- What does that imply about the form of P(Y|X)?

$$P(Y = 1|X =< X_1, \ldots X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

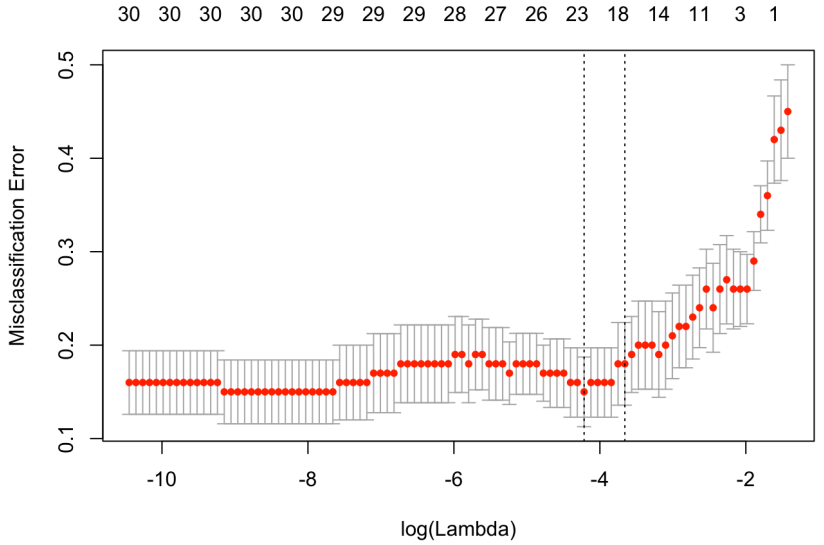
[Slide from Tom Mitchell et al.]

Linear models in application

- R: glmnet package
  - Comprehensive regression formats
    - Linear / Logistic / Cox regression…

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda \left[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1 \right]$$

  - Flexible penalty form
    - Ridge, Lasso, elastic net regression
  - Optimization algorithms with a lot of heuristics
    - E.g., coordinate descent, warm start…
  - Easy to analyze results in a few lines
- Python: scikit-learn package



Take home message

- Machine learning is to learn a model based on observed data that can be generalized
- Two paradigms in supervised learning
  - Regression
  - Classification
- Objective function and probabilistic explanation (MLE)
  - For linear regression, minimize MSE can be equivalent to MLE
  - For logistic regression, minimize cross-entropy is equivalent to MLE
- Regularization and probabilistic explanation (MAP)
  - Ridge is equivalent to Gaussian prior
  - Lasso is equivalent to Laplacian prior
- Generative model vs discriminative model for classification
  - The most import application of Naïve Bayes: document classification
  - The inference of Naïve Bayes can be reduced to the same form of logistic regression

References

- Eric Xing, Tom Mitchell. 10701 Introduction to Machine Learning: http://www.cs.cmu.edu/~epxing/Class/10701-06f/
- Barnabás Póczos. 10715 Advanced Introduction to Machine Learning: https://sites.google.com/site/10715advancedmlintro2017f/lectures
- Eric Xing, Ziv Bar-Joseph. 10701 Introduction to Machine Learning: http://www.cs.cmu.edu/~epxing/Class/10701/
- Christopher M. Bishop. Pattern recognition and Machine Learning
- Wikipedia