Bayesian Learning

Instructor: Steven C.H. Hoi

School of Information Systems

Singapore Management University

Email: chhoi@smu.edu.sg

Outline

- Bayesian Learning
  - Maximum-Likelihood Estimation (MLE)
  - Bayes Theorem
  - Maximum A Posterior (MAP)
- Generative Models
  - Naïve Bayes Classifier
- Discriminative Models
  - Logistic Regression

Density Estimation

- **Density Estimation** task
  - To construct an estimate of an unobservable underlying probability density function, based on some observed data

- **Data**
  - Data sample x drawn i.i.d. (independent identically distributed) from set X according to some distribution d,

$$x_1, \ldots, x_m \in X.$$

- **Problem**
  - To find a distribution p out of a set P that best estimates the true distribution d

Maximum-Likelihood Estimation (MLE)

- **Likelihood**: probability of observing sample under distribution d , which, given the independence assumption is $$\Pr[x_1, \ldots, x_m] = \prod_{i=1}^{m} p(x_i)$$
- **MLE Principle**: select a distribution maximizing the sample probability

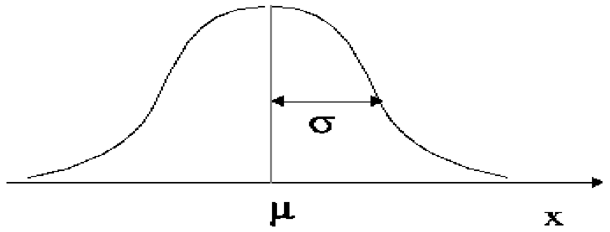
$$p_\star = \underset{p \in \mathcal{P}}{\operatorname{argmax}} \prod_{i=1}^{m} p(x_i),$$ Likelihood

$$p_\star = \underset{p \in \mathcal{P}}{\operatorname{argmax}} \sum_{i=1}^{m} \log p(x_i).$$ Log-likelihood

Example: Gaussian Distribution

- **Problem**: find most likely Gaussian distribution, given sequence of real-valued observations:

$$3.18, 2.35, .95, 1.175, \ldots$$



- **Normal distribution:**

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

- **(log)-Likelihood**: $l(p) = -\frac{1}{2} m \log(2\pi\sigma^2) - \sum_{i=1}^{m} \frac{(x_i - \mu)^2}{2\sigma^2}.$
- **Solution**:

Maximum-Likelihood Estimation (MLE)

- Given training data D, MLE is to find the best hypothesis h that maximizes the likelihood of the training data

$$h_{\mathrm{ML}} = \arg\max_{h \in \mathcal{H}} P(\mathcal{D}|h)$$

- What if you have some ideas about your hypothesis/parameters?

Bayes Theorem


Thomas Bayes (1702–1761)

- Bayes Theorem/Rule

Posterior $\propto$ Likelihood Prior

$$\boxed{P(h|D)} = \frac{\boxed{P(D|h)}\boxed{P(h)}}{P(D)}$$

  - P(h) = prior probability of hypothesis h (Prior)
  - P(D) = prior probability of training data D (Evidence)
  - P(h|D) = conditional probability of h given D (Posterior)
  - P(D|h) = conditional probability of D given h (Likelihood)

Example: Disease Diagnosis

- Given:
  - A doctor knows that **meningitis** causes **stiff neck** 50% of the time
  - Prior probability of any patient having **meningitis** is 1/50,000
  - Prior probability of any patient having **stiff neck** is 1/20
- If a patient has the **Stiff neck** symptom, what is the probability he/she has the **Meningitis** disease?

M – Meningitis S – Stiff neck

$$P(M \mid S) =$$

Maximum A Posterior (MAP)

- Maximum a Posterior (MAP)
  - Find the most probable hypothesis given the training data by maximizing the posterior prob.

$$h_{\mathrm{MAP}} = \arg\max_{h \in \mathcal{H}} P(h|\mathcal{D})$$

$$= \arg\max_{h \in \mathcal{H}} \frac{P(\mathcal{D}|h)P(h)}{P(\mathcal{D})}$$

Prior encodes the knowledge /preference

$$h_{\mathrm{MAP}} = \arg\max_{h \in \mathcal{H}} P(\mathcal{D}|h)\boxed{P(h)}$$

Maximum A Posterior (MAP)

- For each hypothesis h in H, calculate the posterior probability

$$P(h|\mathcal{D}) \propto P(\mathcal{D}|h)P(h)$$

- Output the hypothesis h with the highest posterior probability:

$$h_{MAP} = \arg\max_{h \in \mathcal{H}} P(h|\mathcal{D})$$

- Comment:
  - Choosing P(h) reflects our prior knowledge about the learning task

MAP vs MLE

- **MLE:** Finding a hypothesis h that maximizes the likelihood of the training data

$$h_{\mathrm{ML}} = \arg\max_{h \in \mathcal{H}} P(\mathcal{D}|h)$$

- **MAP:** Finding a hypothesis h that maximizes the posterior probability given the training data

$$h_{MAP} = \arg\max_{h \in \mathcal{H}} P(h|\mathcal{D})$$

- For a uniform prior, MLE coincides with MAP

$$P(h|\mathcal{D}) \propto P(\mathcal{D}|h)P(h)$$

$$P(h_i) = P(h_j) \quad \forall h_i, h_j \in \mathcal{H}$$

Generative Models: Naïve Bayes

Probabilistic Generative Models

- Given training data sampled from K classes:

$$(\mathbf{x}_i, y_i), i = 1, \ldots, n$$

- Classify instance x into one of *K* classes

$$p(\mathcal{C}_k|\mathbf{x}) \propto \boxed{p(\mathbf{x}|\mathcal{C}_k)}\boxed{p(\mathcal{C}_k)}$$

Density function for class C_k

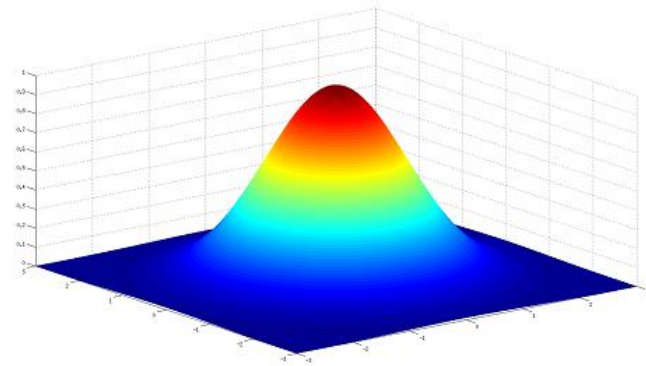
Class prior

Probabilistic Generative Models

- Classify instance x into one of *K* classes

$$p(\mathcal{C}_k|\mathbf{x}) \propto \boxed{p(\mathbf{x}|\mathcal{C}_k)}\,p(\mathcal{C}_k)$$

Density function for class C_k



$$\begin{aligned} p(\mathbf{x}|\mathcal{C}_k) &= \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \\ &= \frac{1}{(2\pi)^{d/2}|\Sigma_k|} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right) \end{aligned}$$

$$\mathbf{x} \in \mathbb{R}^d,\ \mu_k \in \mathbb{R}^d,\ \Sigma_k \in S_{++}^{d\times d}$$

Classification by MAP

- Making a classification decision by MAP

$$k^* = \arg\max_{1 \leq k \leq K} p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

$$\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

- The key is to estimate the parameters

$$\mu_k, \Sigma_k, p(\mathcal{C}_k)$$

Parameter Estimation

- Given training data $(\mathbf{x}_i, y_i), i = 1, \ldots, n$
- Closed-form solutions by MLE:

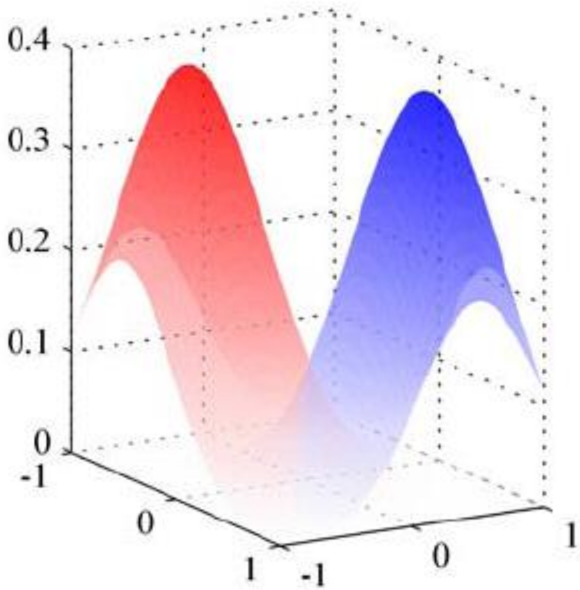
$$\mu_k = \frac{\sum_{i=1}^n \delta(y_i, \mathcal{C}_k)\mathbf{x}_i}{\sum_{i=1}^n \delta(y_i, \mathcal{C}_k)} \qquad \delta(y_i, \mathcal{C}_k) = \begin{cases} 1 & \text{if } \ y_i = \mathcal{C}_k \\ 0 & \text{otherwise.} \end{cases}$$

$$\Sigma_k = \frac{\sum_{i=1}^n \delta(y_i, \mathcal{C}_k)(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top}{\sum_{i=1}^n \delta(y_i, \mathcal{C}_k)}$$

$$p(y = \mathcal{C}_k) = \frac{1}{n}\sum_{i=1}^n \delta(y_i, \mathcal{C}_k)$$

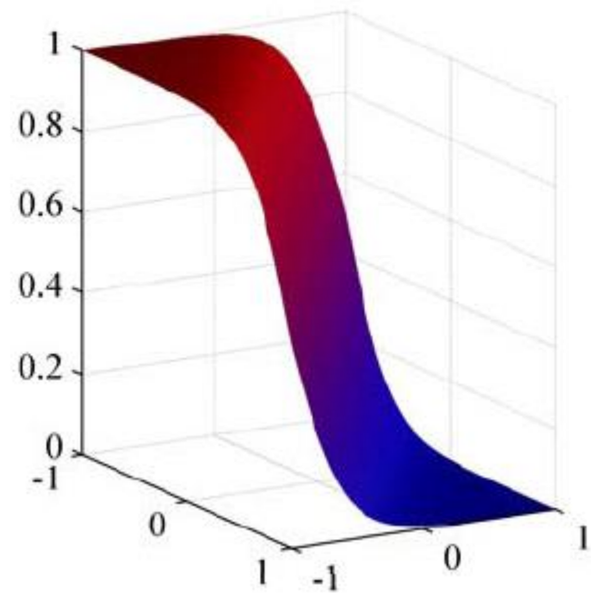
Probabilistic Generative Models

- Two-class Gaussian generative models



class-conditional densities

$$p(\mathbf{x}|\mathcal{C}_k)$$

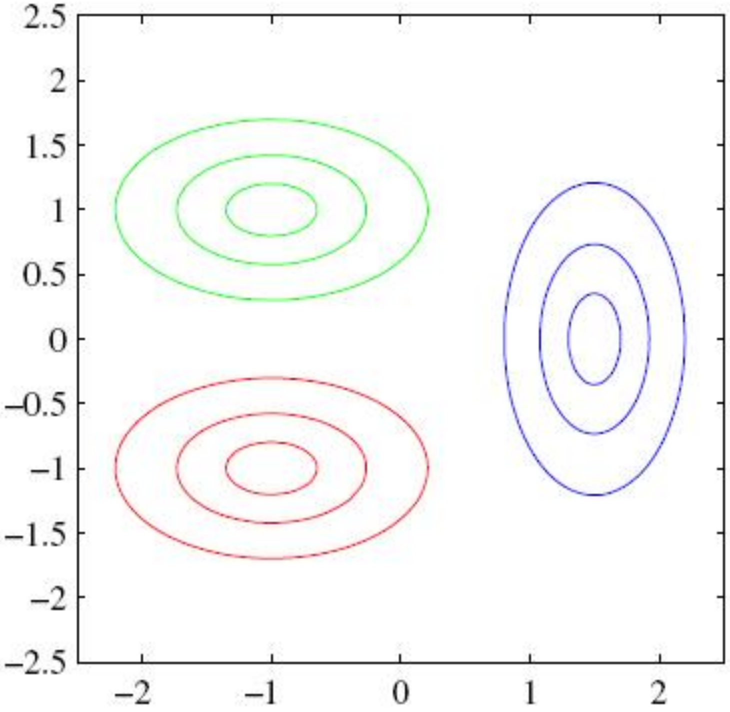


posterior probability

$$p(\mathcal{C}_k|\mathbf{x})$$

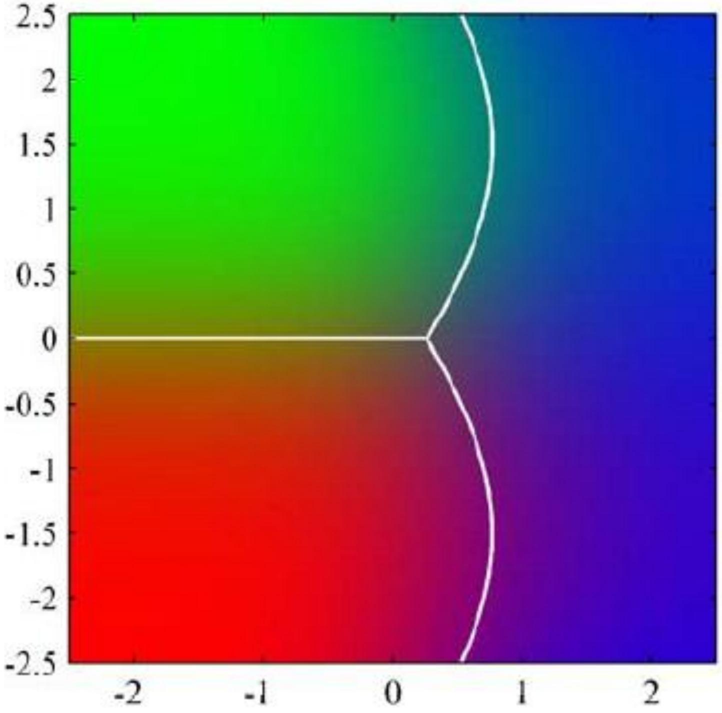
Probabilistic Generative Models

- Three-class Gaussian generative models



class-conditional densities

$$p(\mathbf{x}|\mathcal{C}_k)$$

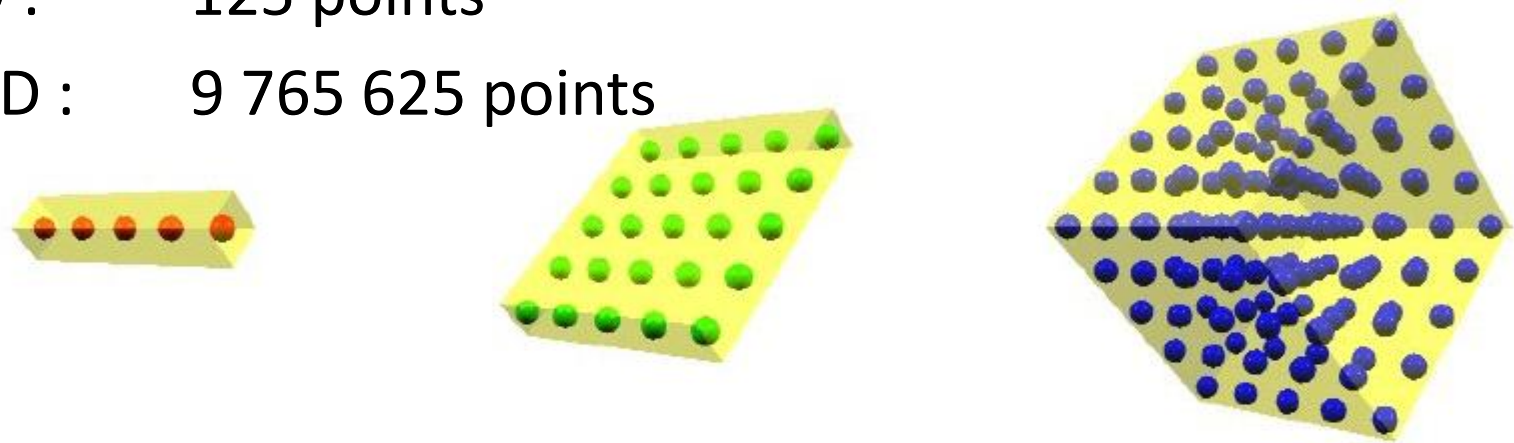


posterior probabilities

$$p(\mathcal{C}_k|\mathbf{x})$$

Curse of Dimensionality

- One challenge of learning with high-dimensional data is **insufficient data samples**
- Suppose 5 samples/objects is considered enough in 1-D
  - 1D : 5 points
  - 2D : 25 points
  - 3D : 125 points
  - 10D : 9 765 625 points



5 points 25 points 125 points

Probabilistic Generative Models

- Singularity of covariance matrix

$$\Sigma_k = \frac{\sum_{i=1}^{n} \delta(y_i, \mathcal{C}_k)(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top}{\sum_{i=1}^{n} \delta(y_i, \mathcal{C}_k)}$$

- Overfitting problem
  - Sample size too small for high-dimensional data
- Solutions
  - Diagonalize the covariance matrix
  - Smoothing/regularization

Naïve Bayes Classifier

- Hard to estimate $p(\mathbf{x}|\mathcal{C}_k)$ for high dimensional data **x**
- Conditional Independence assumption
  - All attributes are conditionally independent
- Naïve Bayes approximation

distribution of 1 D

$$p(\mathbf{x}|\mathcal{C}_k) \approx \prod_{j=1}^{d} p(x_j|\mathcal{C}_k)$$

- Gaussian distribution for Gaussian Naïve Bayes

$$p(\mathbf{x}|\mathcal{C}_k) = \mathcal{N}(\mathbf{x}|\mu, \Sigma) \approx \underbrace{\prod_{j=1}^{d} p(x_j|\mathcal{C}_k) = \prod_{j=1}^{d} \mathcal{N}(x_j|\mu_j, \sigma_j^2)}_{\text{Diagonalize the covariance matrix}}$$

Naïve Bayes Classifier

- For classification task, we are interested in $p(\mathcal{C}_k|\mathbf{x})$ not $p(\mathbf{x}|\mathcal{C}_k)$

$$P(\mathcal{C}_k|\mathbf{x}) = \frac{P(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k)}{P(\mathbf{x})} \quad \propto p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

- Naïve Bayes (NB) Classifier:

$$\mathcal{C}_{NB} = \arg\max_{\mathcal{C}_k} P(\mathcal{C}_k) \prod_j P(x_j|\mathcal{C}_k)$$

Parameter Estimate for Discrete-Valued Inputs

- Previously we assume Gaussian distribution for continuous-valued inputs

$$p(\mathbf{x}|\mathcal{C}_k) = \mathcal{N}(\mathbf{x}|\mu, \Sigma) \approx \prod_{j=1}^{d} p(x_j|\mathcal{C}_k) = \prod_{j=1}^{d} \mathcal{N}(x_j|\mu_j, \sigma_j^2)$$

- Parameter estimate for discrete-valued inputs

$$P(x_j = v|\mathcal{C}_k) = \frac{\sum_{i=1}^{n} \delta(x_{ij}, v)\delta(y_i, \mathcal{C}_k)}{\sum_{i=1}^{n} \delta(y_i, \mathcal{C}_k)}$$

$$\delta(y_i, \mathcal{C}_k) = \begin{cases} 1 & \text{if } \ y_i = \mathcal{C}_k \\ 0 & \text{otherwise.} \end{cases} \qquad \delta(x_{ij}, v) = \begin{cases} 1 & \text{if } x_{ij} = v \\ 0 & \text{otherwise} \end{cases}$$

Example: “Play Tennis or Not”

- Based on the examples in the table, classify the following test sample:

  x=(Outl=Sunny, Temp=Cool, Hum=High, Wind=strong)

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

Example: “Play Tennis or Not”

$$h_{NB} = \arg\max_{h\in[yes,no]} P(h)P(\mathbf{x}\mid h) = \arg\max_{h\in[yes,no]} P(h)\prod_t P(a_t\mid h)$$

$$= \arg\max_{h\in[yes,no]} P(h)P(Outlook = sunny\mid h)P(Temp = cool\mid h)P(Humidity = high\mid h)P(Wind = strong\mid h)$$

P(h=Yes|x=(sunny, cool, high, strong))

$\propto$

P(yes)P(sunny|y)P(cool|y)P(high|y)P(strong|y)

P(yes) =

P(sunny|yes) =

P(cool|yes) =

P(high|yes) =

P(strong|yes) =

P(h=No|x=(sunny, cool, high, strong))

$\propto$

P(no)P(sunny|n)P(cool|n)P(high|n)P(strong|n)

P(no) =

P(sunny|no) =

P(cool|no) =

P(high|no) =

P(strong|no) =

The Independence Assumption

- Makes computation possible
- Yields optimal classifiers when satisfied
- Fairly good empirical results
- But is seldom satisfied in practice, as attributes (variables) are often correlated
- Attempts to overcome this limitation:
  - Bayesian networks, that combine Bayesian reasoning with causal relationships between attributes

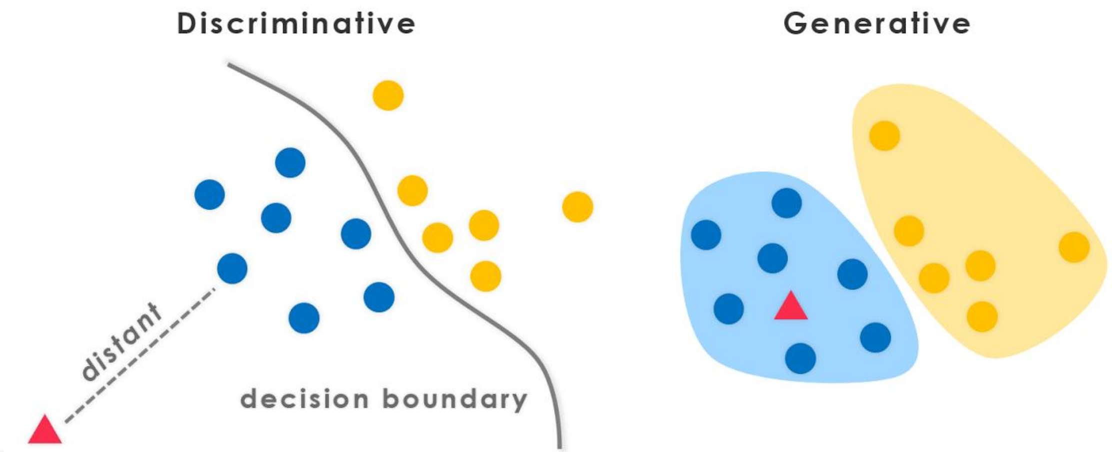
Discriminative Models: Logistic Regression

Discriminative Models

- Generative models
  - First need to estimate $p(\mathbf{x}|\mathcal{C}_k)$ and $p(\mathcal{C}_k)$
  - Then apply Bayes Theorem to predict

$$p(\mathcal{C}_k|\mathbf{x}) \propto p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

- Discriminative models
  - Why not directly model $p(\mathcal{C}_k|\mathbf{x})$

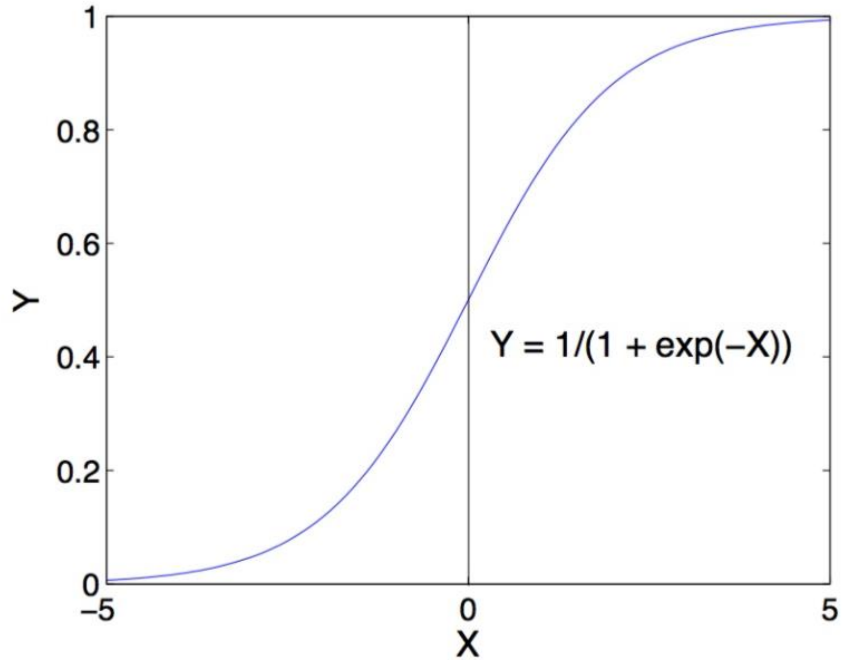


Logistic Regression

- How to model the distribution $p(\mathcal{C}_k|\mathbf{x})$
- Logistic Regression assumes a parametric form for the distribution:

$$
\begin{aligned}
p(y|\mathbf{x}) &= \frac{1}{\exp(-y\mathbf{w}^\top\mathbf{x})+1} \\
&= \sigma(y\mathbf{w}^\top\mathbf{x})
\end{aligned}
$$

logistic / sigmoid function



Logistic / Sigmoid Function

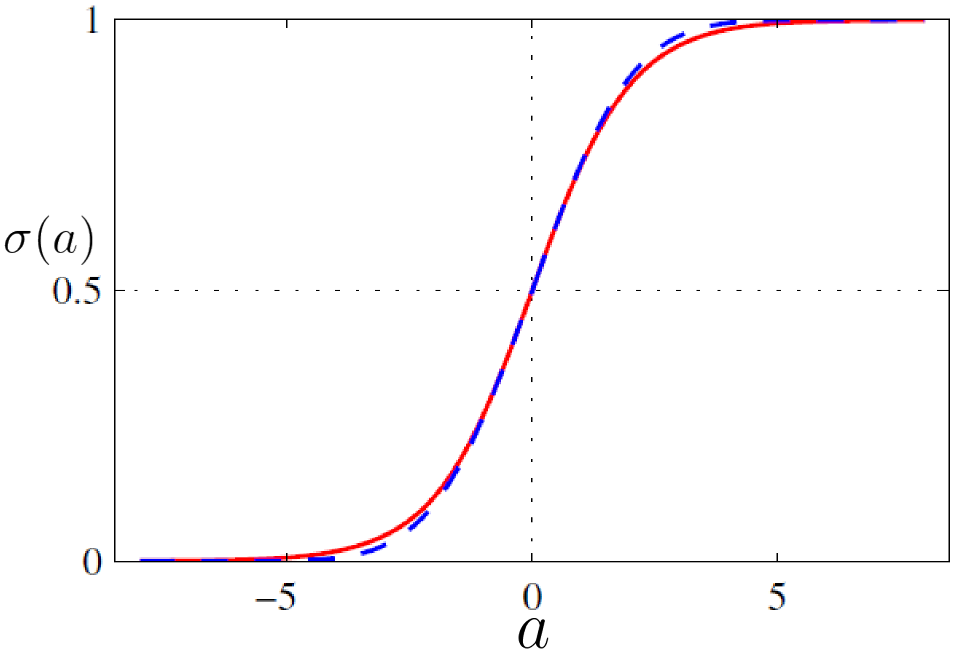
- The *logistic / sigmoid* function $\sigma(a)$

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$

- Property

$$\sigma(-a) = 1 - \sigma(a)$$

$$\frac{d\sigma}{da} = \sigma(1-\sigma)$$



Decision boundary of Logistic Regression

- Consider two-class classification

$$p(y=1|\mathbf{x}) > p(y=-1|\mathbf{x}) \Leftrightarrow \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})} > 1$$

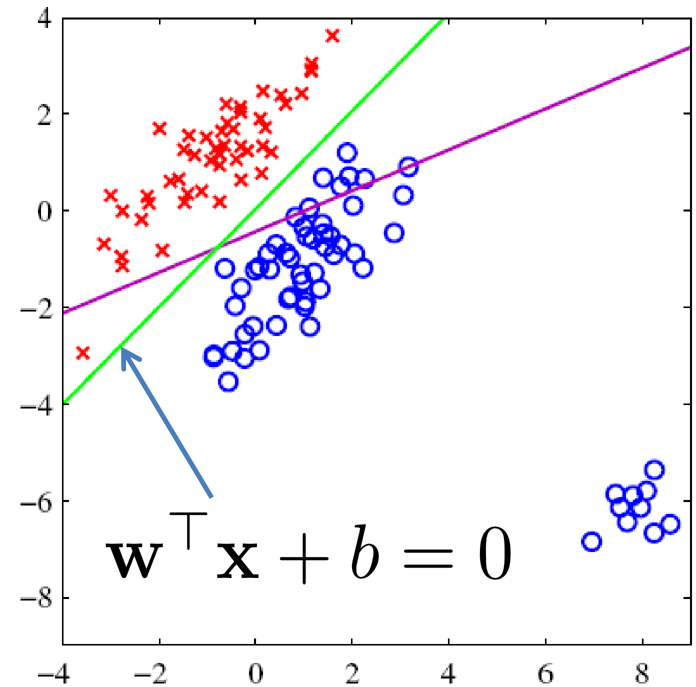
- For logistic function

$$\ln \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})} = \mathbf{w}^\top \mathbf{x} + b \rightarrow \mathbf{w}^\top \mathbf{x}$$

- Decision boundary is linear

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

$$y = \begin{cases} +1 & \text{if } \mathbf{w}^\top \mathbf{x} + b > 0 \\ -1 & \text{otherwise} \end{cases}$$



Logistic Regression: Optimization

- How to learn the optimal parameters **w**:

$$y = \begin{cases} +1 & \text{if } \mathbf{w}^\top \mathbf{x} + b > 0 \\ -1 & \text{otherwise} \end{cases} \quad \begin{aligned} p(y|\mathbf{x}) &= \frac{1}{\exp(-y\mathbf{w}^\top \mathbf{x}) + 1} \\ &= \sigma(y\mathbf{w}^\top \mathbf{x}) \end{aligned}$$

- Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

- Likelihood or the Log-Likelihood:

$$\mathcal{L}(\mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i; \mathbf{w}) \Leftrightarrow \ln \mathcal{L}(\mathbf{w}; \mathcal{D}) = \sum_{i=1}^{N} \ln p(y_i|\mathbf{x}_i; \mathbf{w})$$

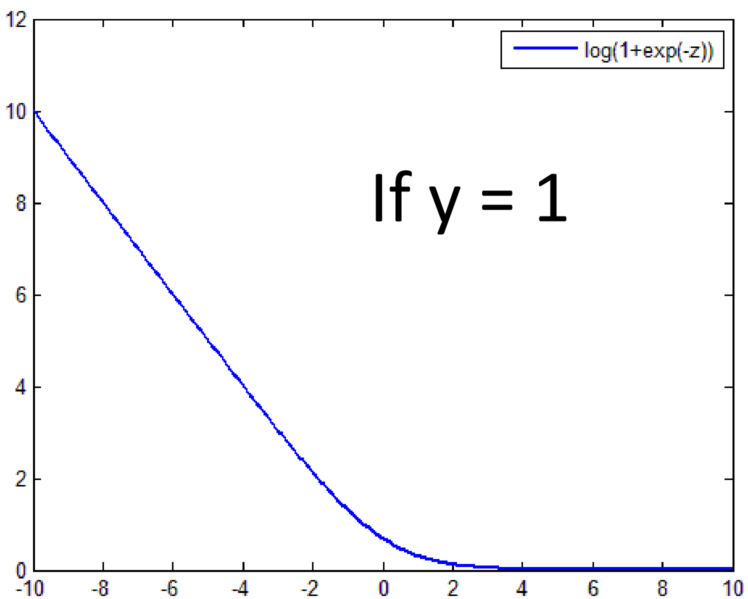
Optimization

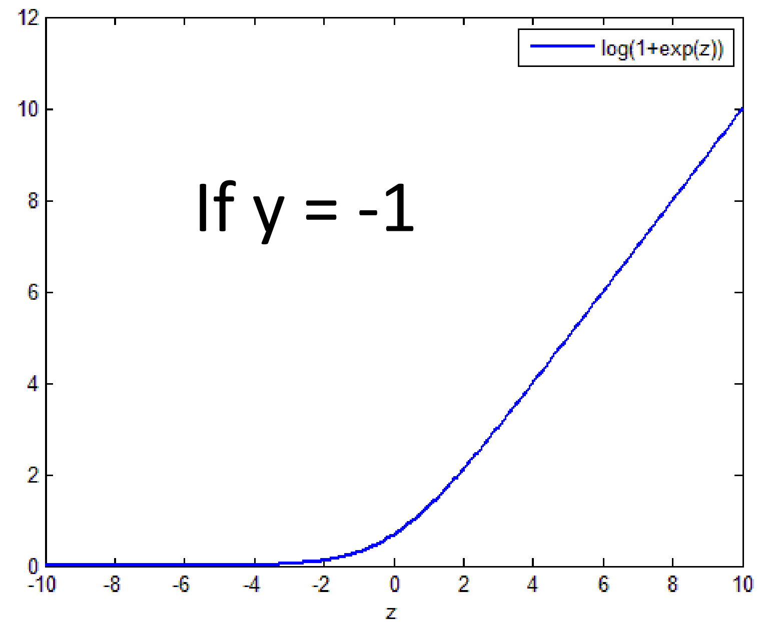
- Maximum Likelihood Estimation:

$$\mathbf{w}^* = \max_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \sum_{i=1}^{N} \ln p(y_i|\mathbf{x}_i)$$

$$\mathbf{w}^* = \min_{\mathbf{w}} \sum_{i=1}^{N} \ln\left(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)\right)$$

- The objective function is convex!





Optimization: Gradient Descent

- Convex objective function: global optima

$$\mathbf{w}^* = \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \sum_{i=1}^{N} \ln\left(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)\right)$$

- No closed-form solution!
- (Batch) Gradient Descent:

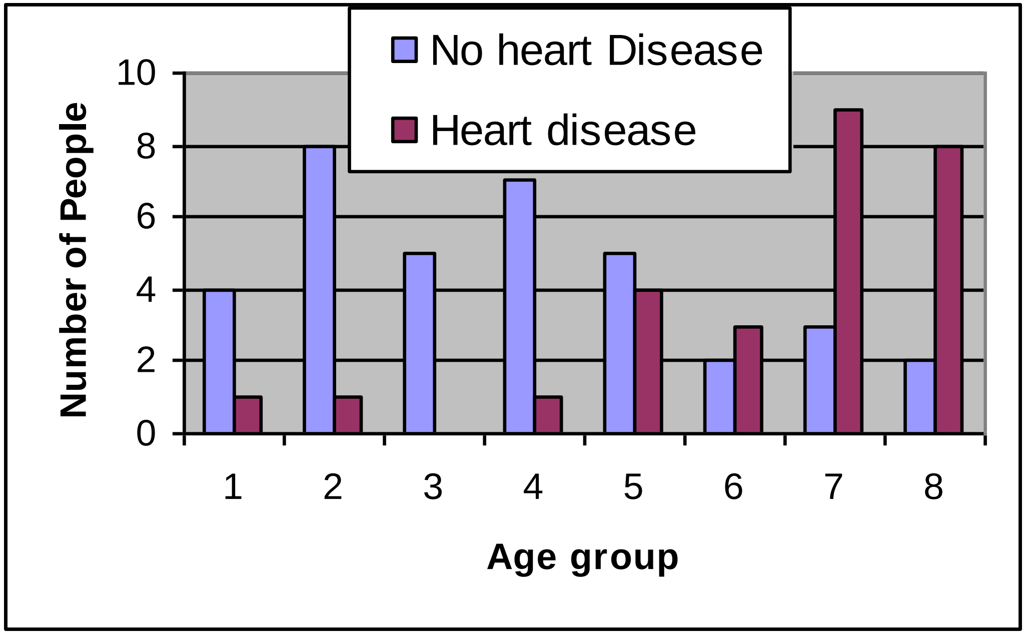
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \mathcal{L}(\mathbf{w}) \qquad \eta_t \propto 1/\sqrt{t}$$

$$\nabla \mathcal{L}(\mathbf{w}) = \sum_{i=1}^{N} \frac{-y_i \mathbf{x}_i \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)}{1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)} = -\sum_{i=1}^{N} y_i \mathbf{x}_i (1 - p(y_i|\mathbf{x}_i))$$

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

Classification error

Example: Heart Disease



1: 25-29

2: 30-34

3: 35-39

4: 40-44

5: 45-49

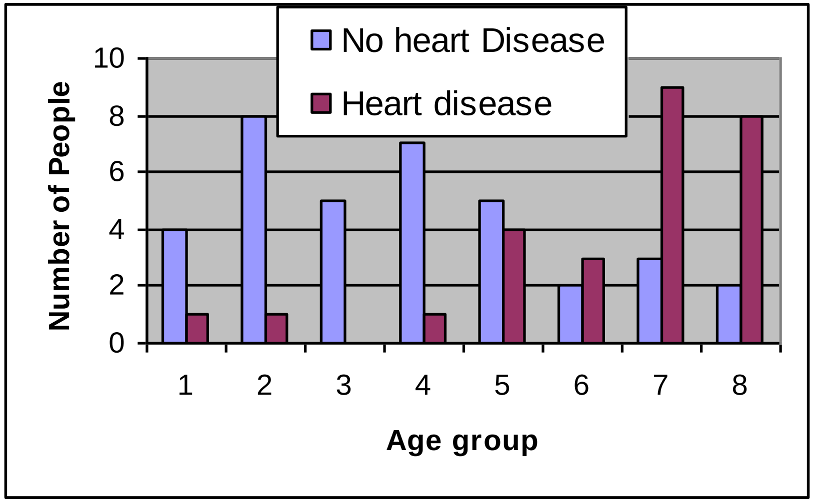
6: 50-54

7: 55-59

8: 60-64

- Input feature *x*: age group id
- Output *y*: if having heart disease
  - y=+1: having heart disease
  - y=-1: no heart disease

Example: Heart Disease



- Logistic Regression

$$p(y \mid x) = \frac{1}{1 + \exp\left[-y(xw + c)\right]}$$

$$\theta = \{w, c\}$$

- Learning w and c: MLE approach

$$l(D_{train}) = \sum_{i=1}^{8} \left\{ n_i(+) \log p(+ \mid i) + n_i(-) \log p(- \mid i) \right\}$$

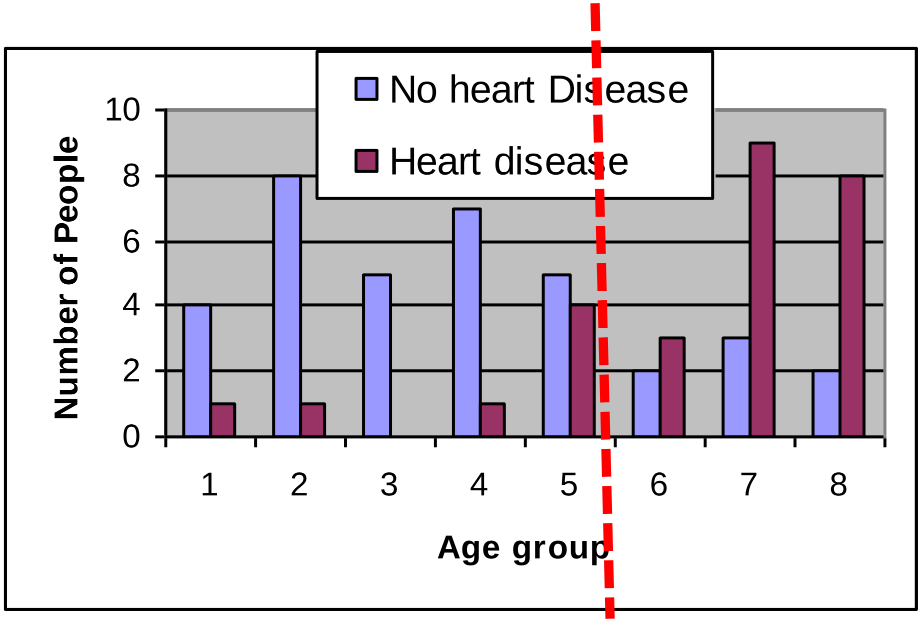
$$= \sum_{i=1}^{8} \left\{ n_i(+) \log \frac{1}{1 + \exp\left[-iw - c\right]} + n_i(-) \log \frac{1}{1 + \exp\left[iw + c\right]} \right\}$$

- Numerical optimization: w = 0.58, c = -3.34

Example: Heart Disease

$$p(+\mid x;\theta)=\frac{1}{1+\exp\left[-xw-c\right]};\, p(-\mid x;\theta)=\frac{1}{1+\exp\left[xw+c\right]}$$

- w = 0.58
  - An old person is more likely to have heart disease
- c = -3.34
  - xw+c < 0 → p(+|x) < p(-|x)
  - xw+c > 0 → p(+|x) > p(-|x)
  - xw+c = 0 → decision boundary
    - x* = 5.78 → 53 year old



Discriminative vs. Generative

Discriminative Models

- Model P(y|x) directly

Pros

- Usually better performance (with small training data)
- Robust to noise data

Cons

- Slow convergence (e.g., LR by gradient descent)
- Expensive computation

Generative Models

- Model P(x|y) directly

Pros

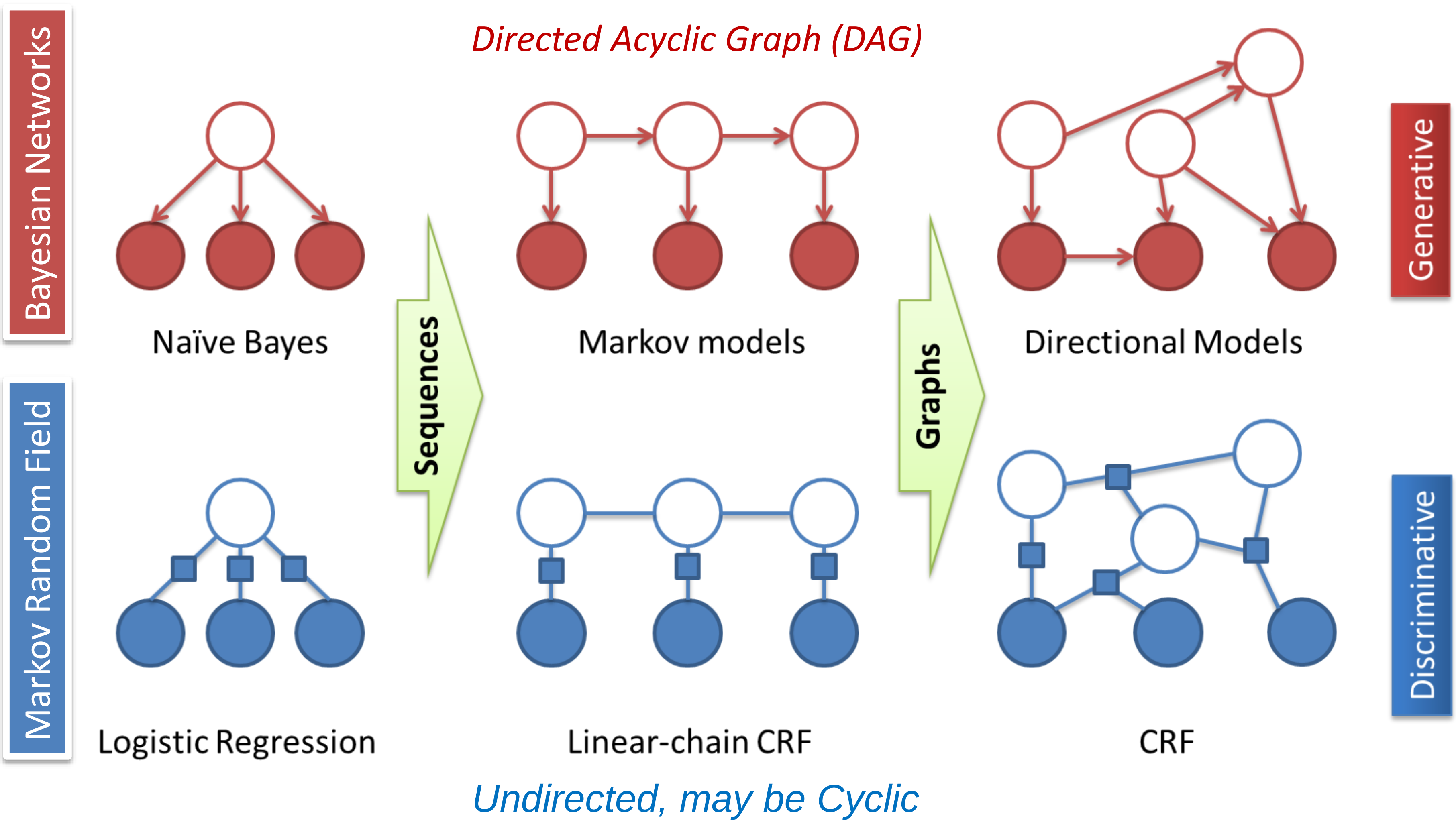
- Usually fast convergence
- Cheap computation (easier to learn, e.g. NB)

Cons

- Sensitive to noise data
- Usually performs worse (with small training data)

One more thing

Probabilistic Graphical Model (PGM)



Adapted from C. Sutton, A. McCallum, "An Introduction to Conditional Random Fields", ArXiv, November 2010

Summary

- Bayesian Learning
  - Bayes Theorem
  - MAP vs. MLE
- Generative Models
  - Naïve Bayes Classifier
- Discriminative Models
  - Logistic Regression

Appendix

- Naïve Bayes for Text Classification
- Logistic Regression for Text Classification
- Naïve Bayes vs Logistic Regression

Naïve Bayes for Text Classification

- Text document represented by the Bag of Words (word histogram of a document)

$$\mathbf{x} = (x_1, x_2, \ldots, x_d)$$

- Multinomial Naive Bayes Classifier
  - Conditional independence: word in one position in the document tells us nothing about words in other positions

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{j=1}^{d} p(x_j|\mathcal{C}_k) \propto \prod_{j=1}^{d} [p(w_j|\mathcal{C}_k)]^{x_j}$$

Occuring times of word w_j in document $\mathbf{x}$

Probability of observing word w_j from documents in class $\mathcal{C}_k$

How to compute $p(w_j|\mathcal{C}_k)$?

Parameter Estimation

- Learning by Maximum Likelihood Estimate
  - Simply count the frequencies in the data

$$P(w_j|\mathcal{C}_k) = \frac{count(w_j, \mathcal{C}_k)}{\sum_{w \in \mathcal{V}} count(w_j, \mathcal{C}_k)}$$

  - Create a mega-document for topic k by concatenating all the docs in this topic
  - Compute frequency of w in the mega-document

Problem with Maximum Likelihood

- What if there is a new word (e.g., any novel words created in internet) in a test document which never appears in the training data

$$\forall \mathcal{C}_k, \qquad P(\text{“}newword\text{”}|\mathcal{C}_k) = 0$$

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{j=1}^{d} p(x_j|\mathcal{C}_k) \propto \prod_{j=1}^{d} [p(w_j|\mathcal{C}_k)]^{x_j} = 0$$

Smoothing to Avoid Overfitting

- Smoothing to avoid Zero Probability

$$P(w_j|\mathcal{C}_k) = \frac{count(w_j, \mathcal{C}_k) + 1}{\sum_{w \in \mathcal{V}}(count(w_j, \mathcal{C}_k) + 1)}$$
$$= \frac{count(w_j, \mathcal{C}_k) + 1}{|\mathcal{V}| + \sum_{w \in \mathcal{V}} count(w, \mathcal{C}_k)}$$

Example

- Apply NB classifier to predict the test document:

	docID	words in documents	c= China?
Training set	1	Chinese Beijing Chinese	Yes
	2	Chinese Chinese Shanghai	Yes
	3	Chinese Macau	Yes
	4	Tokyo Japan Chinese	No
Test set	5	Chinese Chinese Chinese Tokyo Japan	?

Ans:

$P(Y) = 3/4$

$P(Chinese|Y)$

$P(Japan|Y)=P(Tokyo|Y)$

$P(N) = 1/4$

$P(Chinese|N)$

$P(Japan|N)=P(Tokyo|N)$

P(Y|d5)

$\propto P(Y)P(Chinese|Y)^3 P(Tokyo|Y)P(Japan|Y)$

P(N|d5)

$\propto P(N)P(Chinese|N)^3 P(Tokyo|N)P(Japan|N)$

Naïve Bayes Classifier

- Bad approximation

$$p(\mathbf{x}|\mathcal{C}_k) \approx \prod_{j=1}^{d} p(x_j|\mathcal{C}_k)$$

- Good classification accuracy
- NB is not naïve!

Text categorization for 20 Newsgroups

Given 1000 training documents from each group Learn to classify new documents according to which newsgroup it came from

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

Naive Bayes: 89% classification accuracy

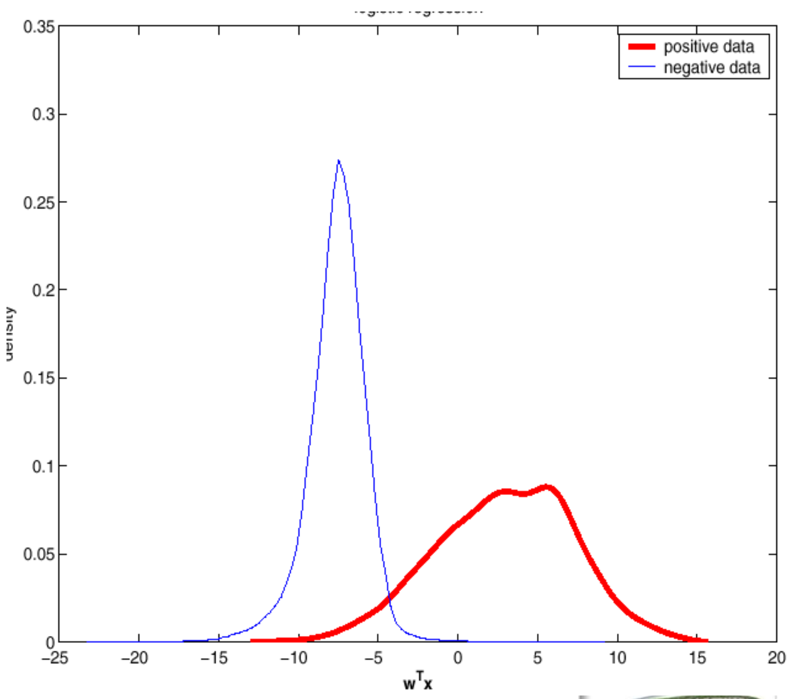
Example 2: Text Categorization

- Training data $\mathcal{D} = \{(\mathbf{d}_1, y_1), \ldots, (\mathbf{d}_N, y_N)\}$

$$\mathbf{d}_i = (d_{i,1}, \ldots, d_{i,m}) \qquad y_i \in \{-1, +1\}$$

$$p(y|\mathbf{d}) = \frac{1}{1 + \exp(-y[\mathbf{w}^\top \mathbf{d} + w_0])}$$

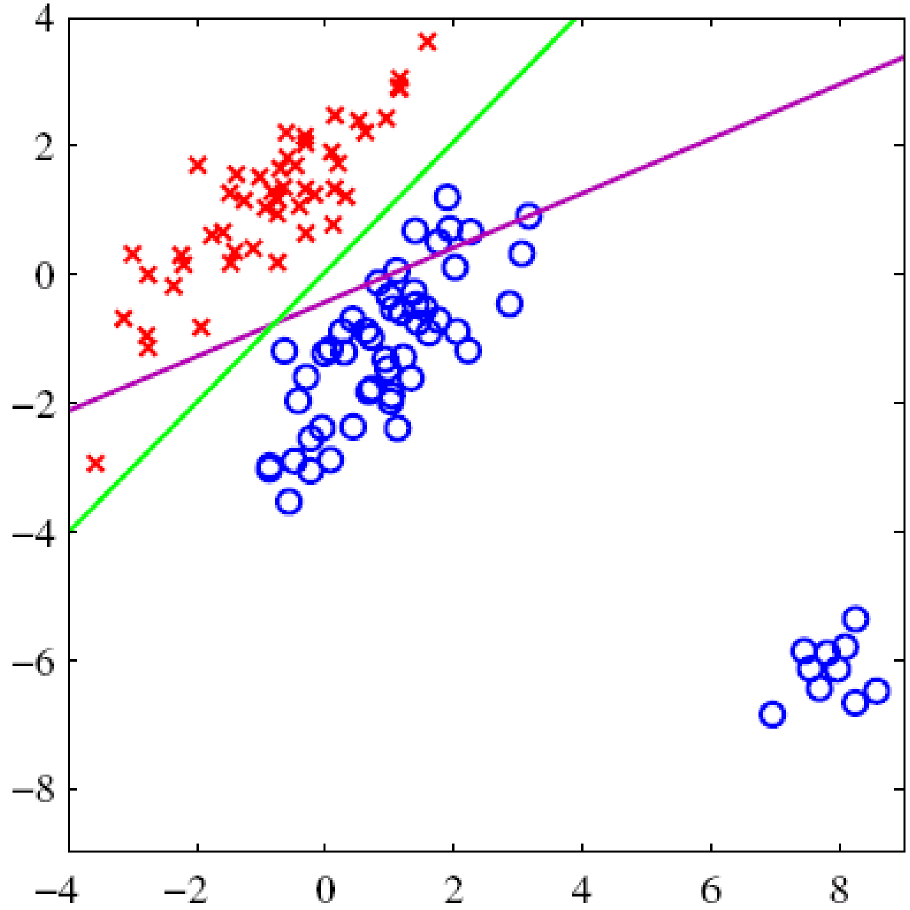
w_j indicates the importance of word j



- Dataset: Reuter-21578
  - Political vs non-political
- Classfication accuracy
  - Naïve Bayes: 77%
  - Logistic regression: 88%

Naïve Bayes vs Logistic Regression

- Both learn linear decision boundary



Decision Boundary of Naïve Beyes

- Consider text categorization of two classes
- The ratio determines the decision

$$\frac{P(\mathcal{C}_1|\mathbf{x})}{P(\mathcal{C}_2|\mathbf{x})} = \frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)} \times \frac{P(\mathbf{x}|\mathcal{C}_1)}{P(\mathbf{x}|\mathcal{C}_2)}$$

$$\ln \frac{p(\mathcal{C}_1|\mathbf{x})}{p(\mathcal{C}_2|\mathbf{x})} = \underbrace{\ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} + \sum_{j=1}^{d} x_j \overset{\text{weight for word } w_j}{\boxed{\ln \frac{p(w_j|\mathcal{C}_1)}{p(w_j|\mathcal{C}_2)}}}}_{\text{Linear decision boundary}}$$

Decision Boundary of Naïve Beyes

- Consider two class classification
- Gaussian density function $p(\mathbf{x}|\mathcal{C}_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$
- Shared covariance matrix $\Sigma_1 = \Sigma_2 = \Sigma$

$$\ln \frac{p(\mathcal{C}_1|\mathbf{x})}{p(\mathcal{C}_2|\mathbf{x})} \propto \underbrace{\ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} - \mathbf{x}^\top \Sigma^{-1}(\mu_1 - \mu_2)}_{\text{Linear decision boundary}}$$

Decision Boundary

- Generative models essentially create linear decision boundaries
- Why not directly model the linear decision boundary

$$\ln \frac{p(\mathcal{C}_1|\mathbf{x})}{p(\mathcal{C}_2|\mathbf{x})} = b + \mathbf{x}^\top \mathbf{w}$$

$\mathbf{w} = (w_1, \ldots, w_d)$ needs to be learned

Logistic Regression

- Generative models often lead to linear decision boundary
- Linear discriminatory model
  - Directly model the linear decision boundary

$$\ln \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})} = \mathbf{w}^\top \mathbf{x} + b \rightarrow \mathbf{w}^\top \mathbf{x}$$

  - **w** is the parameter to be decided

Logistic Regression

$$\ln \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})} = \mathbf{w}^\top \mathbf{x}$$

$$\begin{aligned} p(y|\mathbf{x}) &= \frac{1}{\exp(-y\mathbf{w}^\top \mathbf{x}) + 1} \\ &= \sigma(y\mathbf{w}^\top \mathbf{x}) \end{aligned}$$

logistic / sigmoid function

$$\mathbf{w}^\top \mathbf{x} + b = 0$$