Since 2004
UET
ĐẠI HỌC CÔNG NGHỆ, ĐHQGHN
VNU-University of Engineering and Technology

Since 1906
VNU
ĐẠI HỌC QUỐC GIA HÀ NỘI
Vietnam National University, Hanoi

INT3405 - Machine Learning

Lecture 4: Classification (P1)

Duc-Trong Le & Viet-Cuong Ta

Hanoi, 02/2023

Recap: Key Issues in Machine Learning

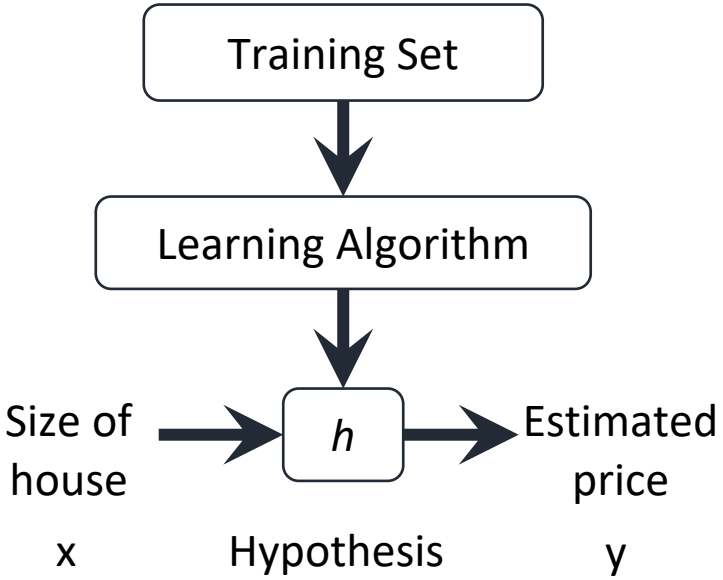
- What are good hypothesis spaces?
  - Which spaces have been useful in practical applications and why?
- What algorithms can work with these spaces?
  - Are there general design principles for machine learning algorithms?

We choose To Optimize

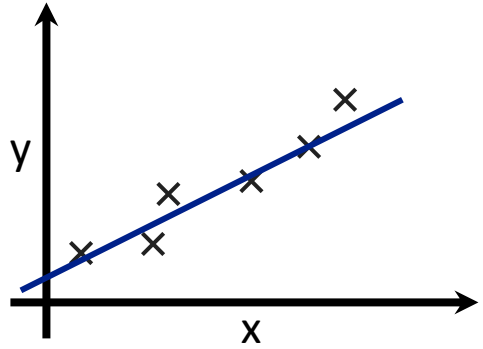
- How can we find the best hypothesis in an efficient way?
  - How to find the optimal solution efficiently ("optimization" question)
- How can we optimize accuracy on future data?
  - Known as the "overfitting" problem (i.e., "generalization" theory)
- How can we have confidence in the results?
  - How much training data is required to find accurate hypothesis? ("statistical" question)

- Are some learning problems computationally intractable? ("computational" question)
- How can we formulate application problems as machine learning problems? ("engineering" question)

Recap: Model Representation



How do we represent *h* ?

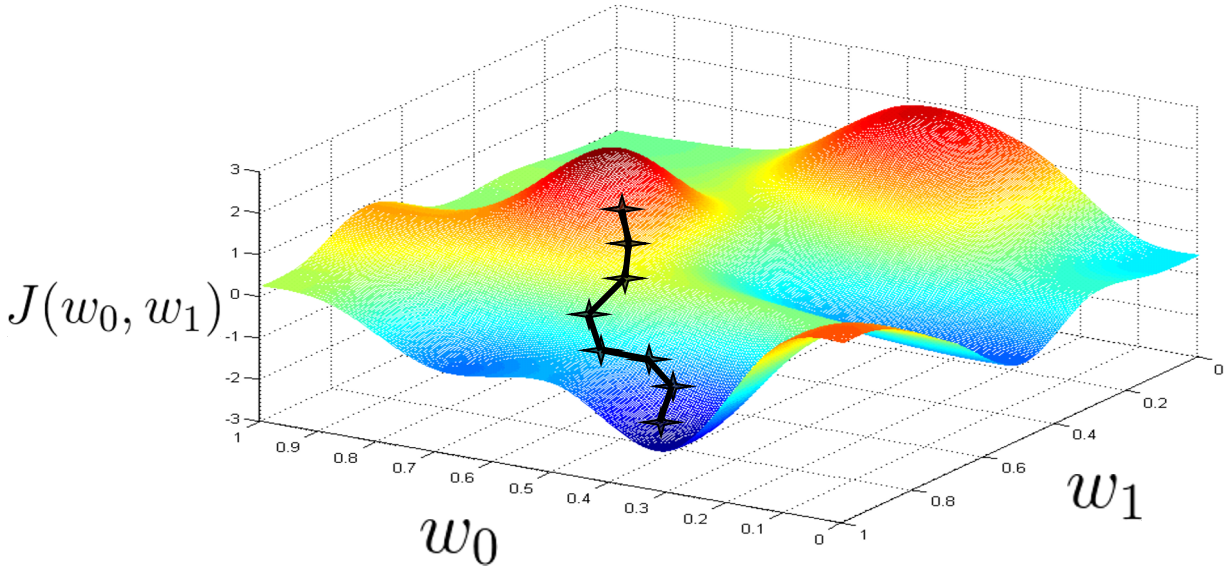


$$h(x) = w_0 + w_1 x$$

Linear regression with one variable.
"**Univariate Linear Regression**"

How to choose parameters w_0, w_1 ?

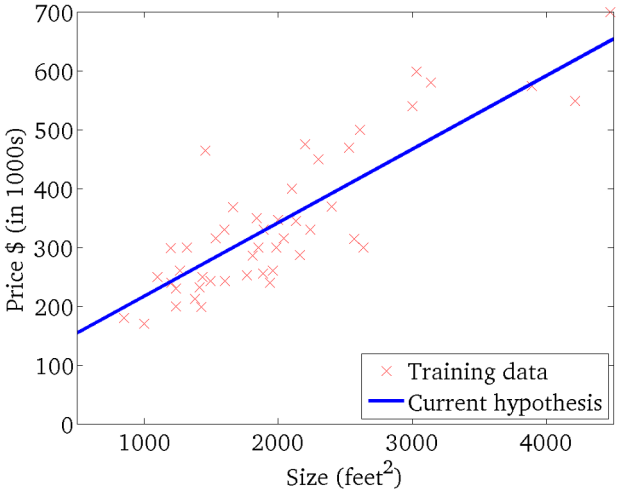
Recap: Gradient Descent for Optimization



Recap: Gradient Descent Example

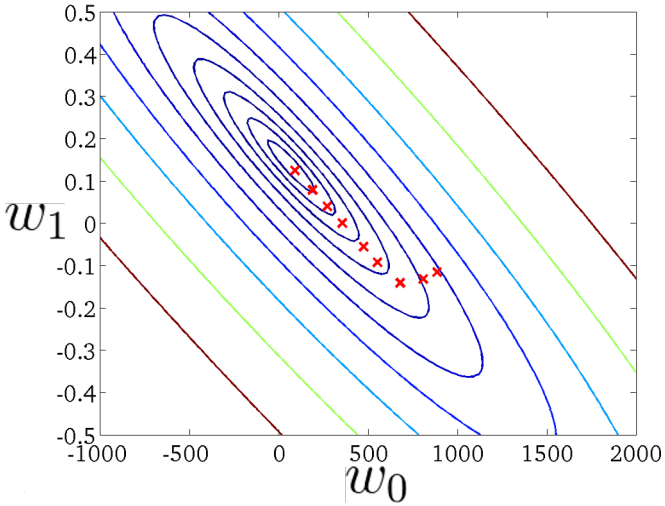
$h(x)$

(for fixed w_0, w_1 , this is a function of x)



$J(w_0, w_1)$

(function of the parameters w_0, w_1)



How fast to converge to the Global Optimal?

Normal Equation (3)

- Matrix-vector formulation

$$J(\mathbf{w}) = \frac{1}{2m}(X\mathbf{w} - \mathbf{y})^T(X\mathbf{w} - \mathbf{y})$$

$$\nabla J(\mathbf{w}) = \nabla_{\mathbf{w}}\left(\frac{1}{2m}(X\mathbf{w} - \mathbf{y})^T(X\mathbf{w} - \mathbf{y})\right)$$

$$= X^T X\mathbf{w} - X^T\mathbf{y} = \mathbf{0}$$

$$X^T X\mathbf{w} = X^T\mathbf{y}$$

- Analytical solution

$$\mathbf{w} = \left((X^T X)^{-1}X^T\right)\mathbf{y} = X^\dagger\mathbf{y}$$

Take $O(mn^2+n^3)$

$$X^\dagger = (X^T X)^{-1}X^T$$

Outline

- Bayesian Learning
  - Bayes Theorem
  - MAP learning vs. MLE learning
- Probabilistic Generative Models
  - Naïve Bayes Classifier
- Discriminative Models
  - Logistic Regression
  - Decision Tree
  - K-Nearest Neighbors

Bayes Theorem

- Bayes Theorem

Posterior $\propto$ Likelihood Prior

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

Thomas Bayes (1702–1761)

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

Maximum A Posterior Learning (MAP)

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

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

Prior encodes the knowledge/preference

MAP Learning

- For each hypothesis h in H, calculate the posterior prob.

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

- Output the hypothesis h with the highest posterior prob.

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

- Comments:
  - Computational intensive
  - Give a standard for judging the performance of learning algorithms
  - Choosing P(h) reflects our prior knowledge about the learning task

Maximum-Likelihood Estimation (MLE)

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

- Maximum Likelihood Estimation (MLE) learning
  - Assume each hypothesis is equally probably a prior

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

  - Maximizing the likelihood of the training data

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

Relationship between MLE Learning and Least-Squared Error Learning (1)

- Consider $\mathcal{D} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
- Assume $y_i = f(x_i) + \epsilon_i \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$

  $y_i \sim \mathcal{N}(f(x_i), \sigma^2)$
- We want learn $h(x) = w^T x$ for *f(x)*
- Linear Regression minimizes the objective (cost function) of the Mean Squared Error

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (h(x_i) - y_i)^2$$

Relationship between MLE Learning and Least-Squared Error Learning (2)

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

$$y_i = f(x_i) + \epsilon_i$$

$$= \arg\max_{h \in \mathcal{H}} \prod_{i=1}^{m} P(y_i|h; \mathbf{x}_i)$$

$$= \arg\max_{h \in \mathcal{H}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - h(x_i))^2}$$

$$h_{ML} = \arg\max_{h \in \mathcal{H}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(y_i - h(x_i))^2$$

$$h_{ML} = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{m} (y_i - h(x_i))^2$$

Probabilistic Generative Models (1)

- Classify instance x into one of *K* classes

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

Density function for class C_k

Class prior

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

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

Probabilistic Generative Models (2)

- Classification decision

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

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

- The key is to decide the parameters

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

Probabilistic Generative Models (3)

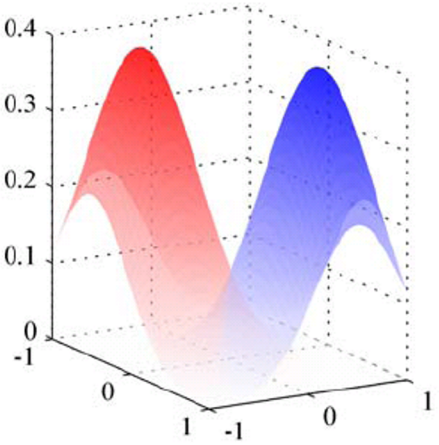
- Given training data $(\mathbf{x}_i, y_i), i = 1, \ldots, n$
- We have closed-form solutions:

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

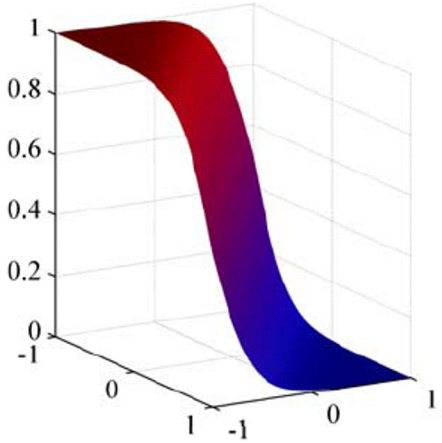
$$\Sigma_k = \frac{\sum_{i=1}^{n} \delta(y_i, \mathcal{C}_k)(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top}{\sum_{i=1}^{n} \delta(y_i, \mathcal{C}_k)}$$

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

Probabilistic Generative Models (4)



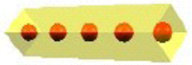
class-conditional densities



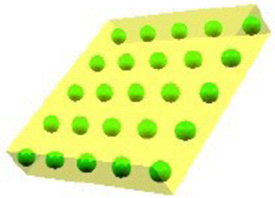
posterior probability

Curse of Dimensionality

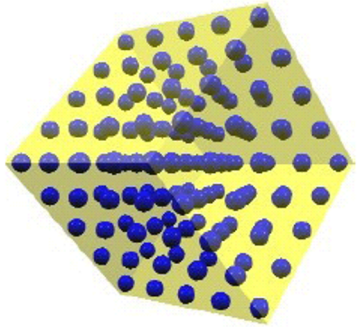
- One challenge of learning with high-dimensional data is **insufficient data samples**
- Suppose 5 samples/objects is considered enough in 1-D
  - 1D : 5 points
  - 2D : 25 points
  - 3D : 125 points
  - 10D : 9 765 625 points



5 points



25 points



125 points

Naïve Bayes Classifier (1)

- 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

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

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

Naïve Bayes Classifier (2)

- **Text categorization**

  $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ word histogram of a document

- Bag of words assumption:
  - Assume position doesn't matter
- Conditional independence:

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

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

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

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

Parameter Estimation

- Learning by Maximum Likelihood Estimates
  - 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

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

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

- Smoothing
  - Avoid zero prob.

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

Naïve Bayes Classifier (3)

- Bad approximation
- Good classification accuracy

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

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

Naïve Bayes Classifier (4)

We are interested $p(\mathcal{C}_k|\mathbf{x})$, not $p(\mathbf{x}|\mathcal{C}_k)$

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

Naïve Bayes Classifier:

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

Example: “Play Tennis” (1)

- Based on the examples in the table, classify the following datum ***x***:

  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” (2)

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

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

$$P(PlayTennis = yes) = 9/14 = 0.64$$

$$P(PlayTennis = no) = 5/14 = 0.36$$

$$P(Wind = strong \mid PlayTennis = yes) = 3/9 = 0.33$$

$$P(Wind = strong \mid PlayTennis = no) = 3/5 = 0.60$$

$$etc.$$

$$P(yes)P(sunny \mid yes)P(cool \mid yes)P(high \mid yes)P(strong \mid yes) = 0.0053$$

$$P(no)P(sunny \mid no)P(cool \mid no)P(high \mid no)P(strong \mid no) = \mathbf{0.0206}$$

$$\Rightarrow answer : PlayTennis(x) = no$$

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

Decision Boundary of Naïve Bayes (1)

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

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

$$\ln \frac{p(\mathcal{C}_1|\mathbf{x})}{p(\mathcal{C}_2|\mathbf{x})} = \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} + \sum_{j=1}^{d} x_j \ln \frac{p(w_j|\mathcal{C}_1)}{p(w_j|\mathcal{C}_2)}$$

weight for word w_j

Linear decision boundary

Decision Boundary of Naïve Bayes (2)

- Consider two class classification
- Gaussian density function
- Shared covariance matrix

$$p(\mathbf{x}|\mathcal{C}_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$\Sigma_1 = \Sigma_2 = \Sigma$$

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

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

Outline

- Bayesian Learning
  - Bayes Theorem
  - MAP learning vs. MLE learning
- Probabilistic Generative Models
  - Naïve Bayes Classifier
- Discriminative Models
  - Logistic Regression
  - Decision Tree
  - K-Nearest Neighbors

Discriminative Models: Logistic Regression

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

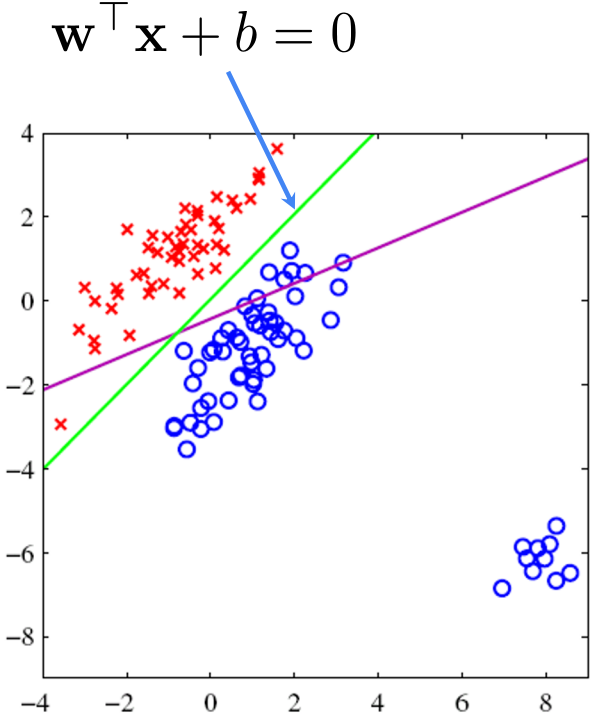
$$\ln \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})} = \mathbf{w}^\top \mathbf{x} + b \rightarrow \mathbf{w}^\top \mathbf{x}$$

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

Logistic Regression

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

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



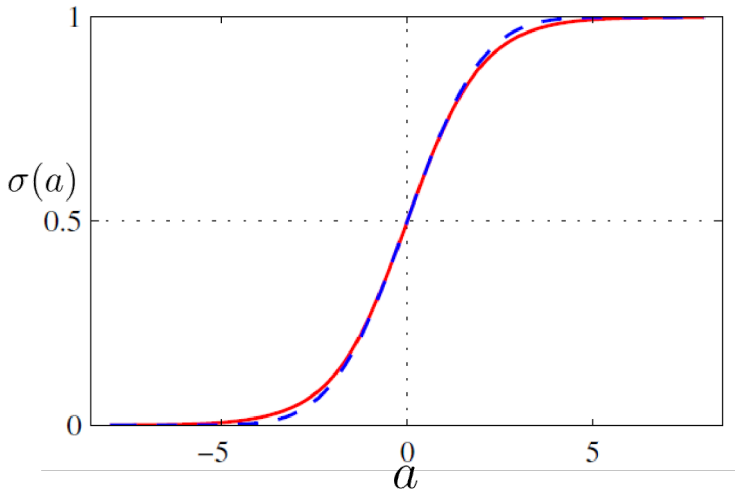
Logistic Sigmoid Function

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

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

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

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



Logistic Regression

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

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

- Learn parameter **w** by Maximum Likelihood Estimation (MLE)

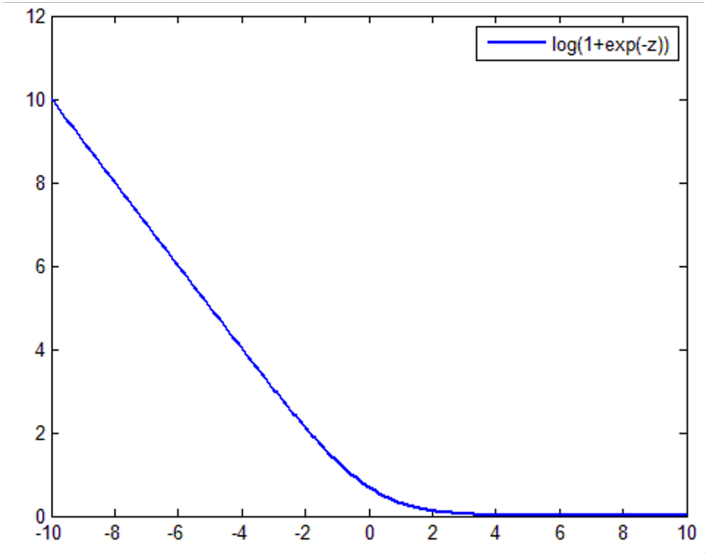
$$\mathbf{w}^* = \max_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \sum_{i=1}^{N} \ln p(y_i|\mathbf{x}_i)$$

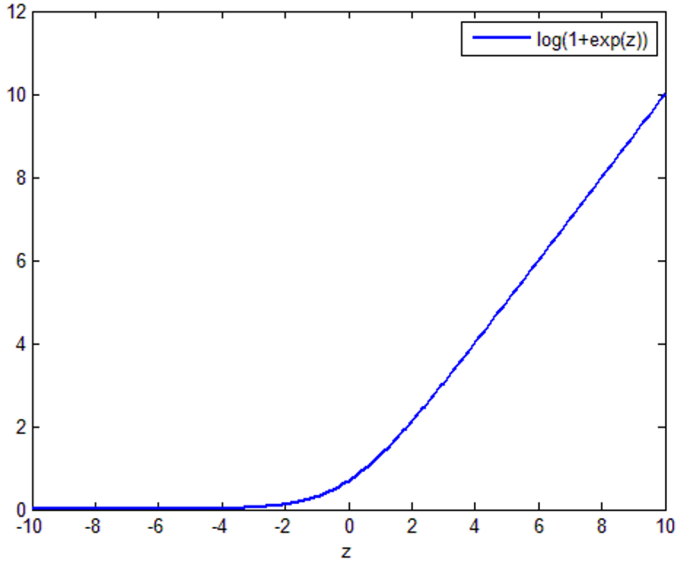
$$\mathbf{w}^* = \min_{\mathbf{w}} \sum_{i=1}^{N} \ln\left(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)\right)$$

Convex Objective Functions

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

If y = 1 Convex Loss Functions: If y = -1





Logistic Regression

- Convex objective function, global optimal

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

- No closed-form solution
- Gradient Descent

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

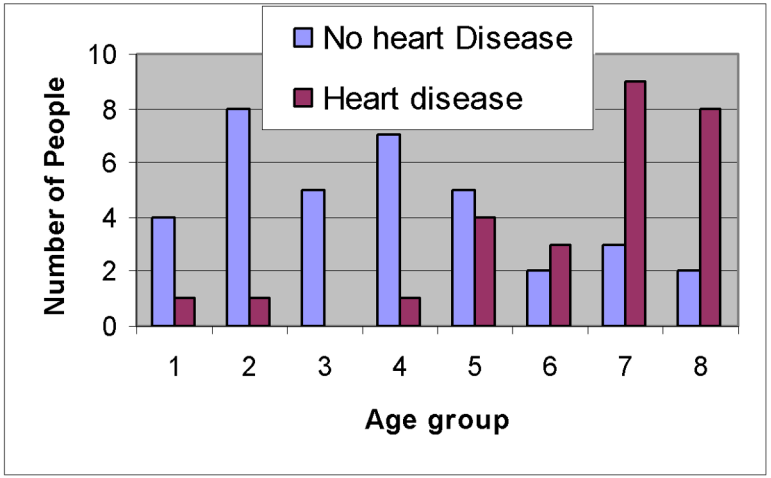
$$-\sum_{i=1}^{N} y_i \mathbf{x}_i (1 - \sigma(y_i \mathbf{w}^\top \mathbf{x}))$$

Classification error

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

$$\eta_t \propto 1/\sqrt{t}$$

Example: Heart Disease (1)



1: 25-29

2: 30-34

3: 35-39

4: 40-44

5: 45-49

6: 50-54

7: 55-59

8: 60-64

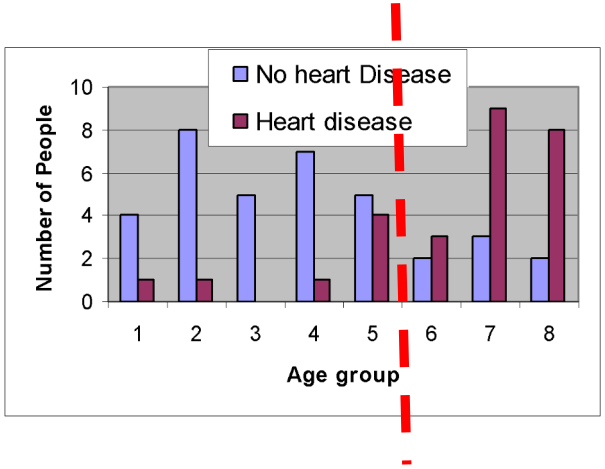
- Input feature *x*: age group id
- Output *y*: if having heart disease
  - y=1: having heart disease
  - y=-1: no heart disease

Example: Heart Disease (2)

$$p(y|x) = \frac{1}{1 + \exp(-y[xw_1 + w_0])}$$

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{8} n_i^+ \ln p(y = 1|i) + n_i^- \ln p(y = -1|i)$$

$$w_1 = 0.58, w_0 = -3.34$$



Example: Text Categorization (1)

- Learn to classify text into two categories
- Input **d**:
  - a document, represented by a word histogram
- Output
  - y=±1:

    +1 for political document

    -1 for non-political document

Example: Text Categorization (2)

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

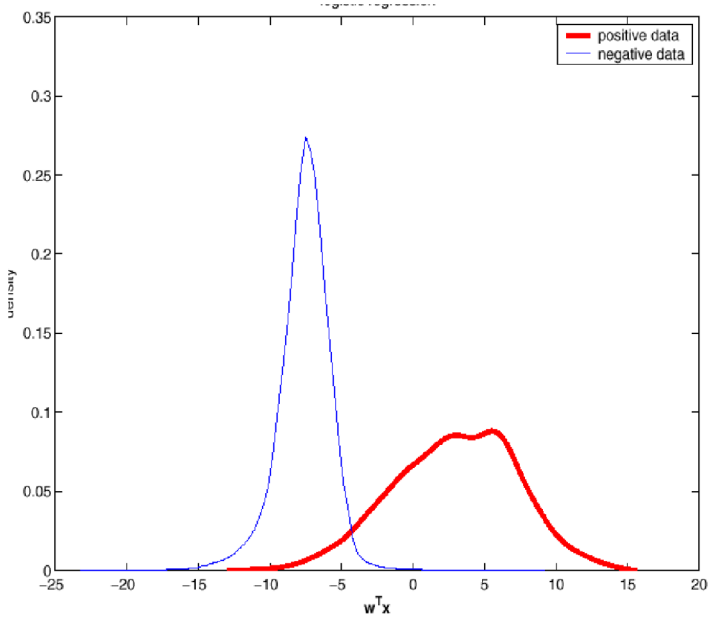
$$\mathbf{d}_i = (d_{i,1}, \ldots, d_{i,m}) \qquad y_i \in \{-1, +1\}$$

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

w_j indicates the importance of word j

Example: Text Categorization (3)

- Dataset: Reuter-21578
- Classification accuracy
  - Naïve Bayes: 77%
  - Logistic regression: 88%

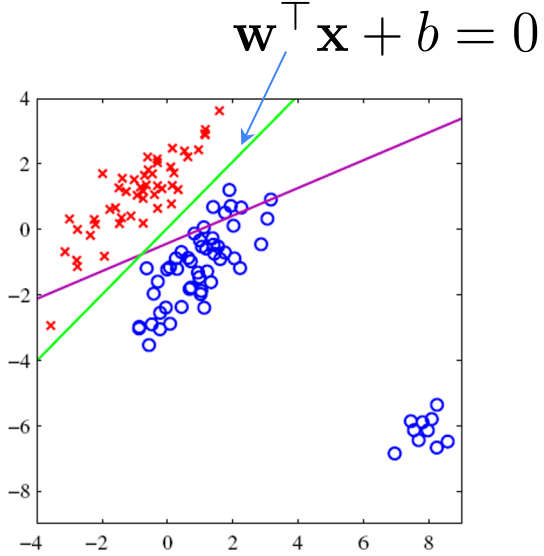


Multi-class Logistic Regression

- How to extend logistic regression model to multi-class classification ?

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

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



Conditional Exponential Model (1)

- Consider K classes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K$
- Define

$$p(\mathcal{C}_k|\mathbf{x}) \propto \exp(\mathbf{w}_k^\top \mathbf{x})$$

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(\mathbf{w}_k^\top \mathbf{x})$$

- where Z is normalization factor:

Normalization factor (partition function)

$$Z(\mathbf{x}) = \sum_{k=1}^{K} \exp(\mathbf{w}_k^\top \mathbf{x})$$

- Need to learn

$$\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K$$

Conditional Exponential Model (2)

- Learn weights **w**'s by maximum likelihood estimation

$$\mathcal{L}(W) = \sum_{i=1}^{N} \ln p(y_i|\mathbf{x}_i) = \sum_{i=1}^{N} \ln \frac{\exp(\mathbf{x}_i^\top \mathbf{w}_{y_i})}{\sum_{k=1}^{K} \exp(\mathbf{x}_i^\top \mathbf{w}_k)}$$

$$W^* = \arg\max_W \mathcal{L}(W)$$

- Modified Conditional Exponential Model

$$p(\mathcal{C}_j|\mathbf{x}) = \begin{cases} \dfrac{1}{1+\sum_{k=2}^{K} \exp(\mathbf{w}_k^\top \mathbf{x})} & j = 1 \\ \dfrac{\exp(\mathbf{w}_j^\top \mathbf{x})}{1+\sum_{k=2}^{K} \exp(\mathbf{w}_k^\top \mathbf{x})} & j > 1 \end{cases}$$

Logistic Regression versus Naïve Bayes

- Both are linear decision boundaries
  - Naïve Bayes: $$w_j = \ln \frac{p(j|y=+1)}{p(j|y=-1)}$$
  - Logistic regression: learn weights by MLE

- Both can be viewed as modeling p(x|y)
  - Naïve Bayes: independence assumption
  - Logistic regression: assume an exponential family distribution for p(x|y) (a broad assumption)

Discriminative versus 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)

Summary

- Bayesian Learning
  - Bayes Theorem
  - MAP learning vs. MLE learning
- Probabilistic Generative Models
  - Naïve Bayes Classifier
- Discriminative Models
  - Logistic Regression
  - Decision Tree
  - K-Nearest Neighbors

Thank you

Email me
cuongtv@vnu.edu.vn