UVA CS 4501 - 001 / 6501 – 007 Introduction to Machine Learning and Data Mining

Lecture 16: Generative vs. Discriminative / K-nearest-neighbor Classifier / LOOCV

Yanjun Qi / Jane, , PhD

University of Virginia
Department of
Computer Science

Yanjun Qi / UVA CS 4501-01-6501-07

Where are we ? ➔ Five major sections of this course

- ❑ Regression (supervised)
- ❑ Classification (supervised)
- ❑ Unsupervised models
- ❑ Learning theory
- ❑ Graphical models

Where are we ? ➔ Three major sections for classification

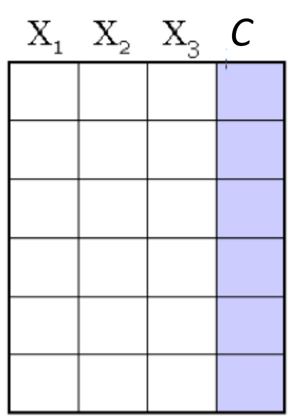
- We can divide the large variety of classification approaches into roughly three major types

1. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., logistic regression, support vector machine, decisionTree

2. Generative:
   - build a generative statistical model
   - e.g., naïve bayes classifier, Bayesian networks

3. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

A Dataset for classification



$$f : \boxed{X} \longrightarrow \boxed{C}$$

Output as Discrete Class Label $C_1, C_2, \ldots, C_L$

Generative: $\operatorname*{argmax}_C P(C \mid X) = \operatorname*{argmax}_C P(X, C) = \operatorname*{argmax}_C P(X \mid C) P(C)$

Discriminative: $P(C \mid \mathbf{X}) \quad C = c_1, \cdots, c_L$

- **Data**/*points/instances/examples/samples/records*: [rows]
- **Features**/*attributes/dimensions/independent variables/covariates/predictors/regressors*: [columns, except the last]
- **Target**/*outcome/response/label/dependent variable*: special column to be predicted [last column]

Generative

Yanjun Qi / UVA CS 4501-01-6501-07

Multinomial Naïve Bayes as Stochastic Language Models

the	boy	likes	the	dog
0.2	0.01	0.0001	0.2	0.0005

Multiply all five terms

Model C1	
0.2	the
0.01	boy
0.0001	said
0.0001	likes
0.0001	black
0.0005	dog
0.01	garden

Model C2	
0.2	the
0.0001	boy
0.03	said
0.02	likes
0.1	black
0.01	dog
0.0001	garden

the	boy	likes	black	dog
0.2	0.01	0.0001	0.0001	0.0005
0.2	0.0001	0.02	0.1	0.01

P(s|C2) P(C2) > P(s|C1) P(C1)

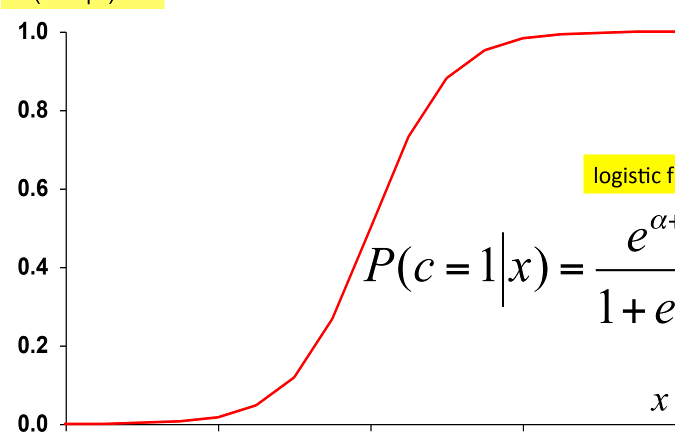
10/22/14 5

Discriminative

Yanjun Qi / UVA CS 4501-01-6501-07

Logistic regression models for binary target variable coded 0/1.

e.g. Probability of disease



logistic function

$$P(c=1|x)=\frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

Logit function

$$\ln\left[\frac{P(c=1|x)}{P(c=0|x)}\right]=\ln\left[\frac{P(c=1|x)}{1-P(c=1|x)}\right]=\alpha+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_p x_p$$

10/22/14 6

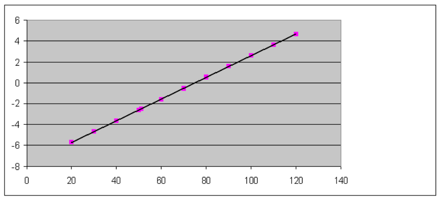
Binary Logistic Regression

In summary that the logistic regression tells us two things at once.

- Transformed, the "log odds" (logit) are linear.

ln[p/(1-p)]

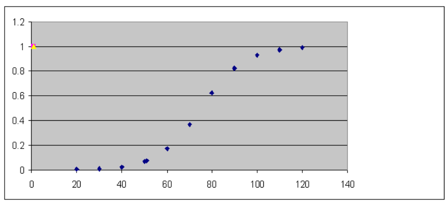
Odds= p/(1-p)



x

- Logistic Distribution

P (Y=1|x)



x

This means we use Bernoulli distribution to model the target variable with its Bernoulli parameter p=p(y=1|x) predefined.

p 1-p

Today : Relevant classifiers / KNN / LOOCV

- ✓ Logistic regression (cont.)
- ✓ Naïve Bayes Gaussian Classifier
- ✓ K-nearest neighbor
- ✓ LOOCV

Multinomial Logistic Regression Model

The method directly models the posterior probabilities as the output of regression

$$\Pr(G=k \mid X=x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}, \quad k = 1, \ldots, K-1$$

$$\Pr(G=K \mid X=x) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}$$

x is p-dimensional input vector

β_k is a p-dimensional vector for each k

Total number of parameters is $(K-1)(p+1)$

Note that the class boundaries are linear

MLE for Logistic Regression Training

Let's fit the logistic regression model for K=2, i.e., number of classes is 2

Training set: (x_i, y_i), i=1,…,N

For Bernoulli distribution

$$p(y \mid x)^y (1-p)^{1-y}$$

Log-likelihood:

$$l(\beta) = \sum_{i=1}^{N} \{\log \Pr(Y = y_i \mid X = x_i)\}$$

$$= \sum_{i=1}^{N} y_i \log(\Pr(Y=1 \mid X = x_i)) + (1-y_i)\log(\Pr(Y=0 \mid X=x_i))$$

$$= \sum_{i=1}^{N} (y_i \log \frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)}) + (1-y_i)\log \frac{1}{1+\exp(\beta^T x_i)})$$

$$= \sum_{i=1}^{N} (y_i \beta^T x_i - \log(1+\exp(\beta^T x_i)))$$

x_i are $(p+1)$-dimensional input vector with leading entry 1
β is a $(p+1)$-dimensional vector
y_i = 1 if C_i =1; y_i = 0 if C_i =0

We want to maximize the log-likelihood in order to estimate β

Newton-Raphson for LR (optional)

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} (y_i - \frac{\exp(\beta^T x)}{1+\exp(\beta^T x)}) x_i = 0$$

(p+1) Non-linear equations to solve for (p+1) unknowns

Solve by Newton-Raphson method:

$$\beta^{new} \leftarrow \beta^{old} - [(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T})]^{-1} \frac{\partial l(\beta)}{\partial \beta},$$

$$\text{where, } (\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T}) = -\sum_{i=1}^{N} x_i x_i^T \underbrace{(\frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)})}_{p(x_i;\beta)} \underbrace{(\frac{1}{1+\exp(\beta^T x_i)})}_{1 - p(x_i;\beta)}$$

Newton-Raphson for LR (optional)

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} (y_i - \frac{\exp(\beta^T x)}{1+\exp(\beta^T x)}) x_i = X^T(y-p)$$

$$(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T}) = -X^T W X$$

So, NR rule becomes: $\beta^{new} \leftarrow \beta^{old} + (X^T W X)^{-1} X^T (y-p),$

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}_{N-by-(p+1)}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}_{N-by-1}, \quad p = \begin{bmatrix} \exp(\beta^T x_1)/(1+\exp(\beta^T x_1)) \\ \exp(\beta^T x_2)/(1+\exp(\beta^T x_2)) \\ \vdots \\ \exp(\beta^T x_N)/(1+\exp(\beta^T x_N)) \end{bmatrix}_{N-by-1},$$

$X : N \times (p+1)$ matrix of x_i

$y : N \times 1$ matrix of y_i

$p : N \times 1$ matrix of $p(x_i;\beta^{old})$

$W : N \times N$ diagonal matrix of $p(x_i;\beta^{old})(1-p(x_i;\beta^{old}))$

$$(\frac{\exp(\beta^T x_i)}{(1+\exp(\beta^T x_i))})(1 - \frac{1}{(1+\exp(\beta^T x_i))})$$

Newton-Raphson for LR…

- Newton-Raphson
  - $\beta^{new} = \beta^{old} + (X^T W X)^{-1} X^T (y - p)$

    $= (X^T W X)^{-1} X^T W (X\beta^{old} + W^{-1}(y - p))$

    $= (X^T W X)^{-1} X^T W z$

    Re expressing Newton step as weighted least square step
  - Adjusted response

    $z = X\beta^{old} + W^{-1}(y - p)$
  - Iteratively reweighted least squares (IRLS)

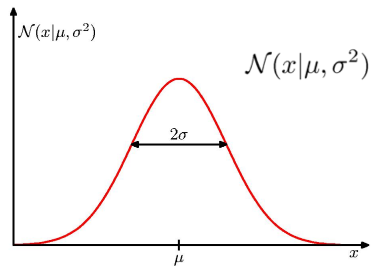
    $$\beta^{new} \leftarrow \arg\min_{\beta} (z - X\beta^T)^T W (z - X\beta^T)$$

    $$\leftarrow \arg\min_{\beta} (y - p)^T W^{-1} (y - p)$$

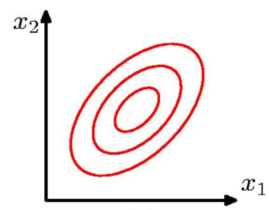
Today : Relevant classifiers / KNN / LOOCV

- ✓ Logistic regression (cont.)
- ✓ Gaussian Naïve Bayes Classifier
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA
  - Discriminative vs. Generative
- ✓ K-nearest neighbor
- ✓ LOOCV

The Gaussian Distribution



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



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

Covariance Matrix

Mean

Courtesy: http://research.microsoft.com/~cmbishop/PRML/index.htm

Multivariate Gaussian Distribution

- A multivariate Gaussian model: $\mathbf{x} \sim N(\mu,\Sigma)$ where

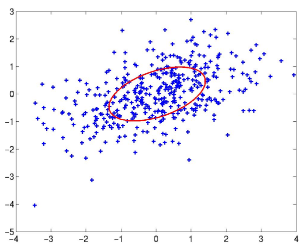
Here μ is the mean vector and Σ is the covariance matrix, if p=2

$\mu = \{\mu_1, \mu_2\}$ Σ =

var(x_1)	cov(x_1,x_2)
cov(x_1,x_2)	var(x_2)

- The covariance matrix captures linear dependencies among the variables

MLE Estimation for Multivariate Gaussian

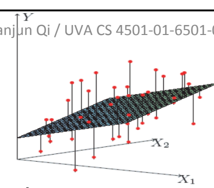


- We can fit statistical models by maximizing the probability / likelihood of generating the observed samples:
  $L(x_1, \ldots, x_n \mid \Theta) = p(x_1 \mid \Theta) \ldots p(x_n \mid \Theta)$
  (the samples are assumed to be independent)

- In the Gaussian case, we simply set the mean and the variance to the sample mean and the sample variance:

$$\overline{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \overline{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{\mu})^2$$

Probabilistic Interpretation of Linear Regression



- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

where ε is an error term of unmodeled effects or random noise

- Now assume that ε follows a Gaussian $N(0,\sigma)$, then we have:

$$p(y_i \mid x_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right)$$

- By independence (among samples) assumption:

$$L(\theta) = \prod_{i=1}^{n} p(y_i \mid x_i;\theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right)$$

Probabilistic Interpretation of Linear Regression (cont.)

- Hence the log-likelihood is:

$$l(\theta) = n \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y_i - \theta^T \mathbf{x}_i)^2$$

- Do you recognize the last term?

Yes it is: $$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$$

- Thus under independence assumption, residual means square is equivalent to MLE of ϑ !

Today : Relevant classifiers / KNN / LOOCV

- ✓ Logistic regression (cont.)
- ✓ Gaussian Naïve Bayes Classifier
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA
  - Discriminative vs. Generative
- ✓ K-nearest neighbor
- ✓ LOOCV

Gaussian Naïve Bayes Classifier

$$\underset{C}{\operatorname{argmax}} P(C|X) = \underset{C}{\operatorname{argmax}} P(X,C) = \underset{C}{\operatorname{argmax}} P(X|C)P(C)$$

$$\hat{P}(X_j | C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (avearage) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

Naïve Bayes Classifier

$$P(X|C) = P(X_1, X_2, \cdots, X_p | C)$$
$$= P(X_1 | X_2, \cdots, X_p, C) P(X_2, \cdots, X_p | C)$$
$$= P(X_1 | C) P(X_2, \cdots, X_p | C)$$
$$= P(X_1 | C) P(X_2 | C) \cdots P(X_p | C)$$

Gaussian Naïve Bayes Classifier

- Continuous-valued Input Attributes
  - Conditional probability modeled with the normal distribution

$$\hat{P}(X_j | C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (avearage) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_p)$, $C = c_1, \cdots, c_L$
    Output: $p \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$
  - Test Phase: for $\mathbf{X}' = (X'_1, \cdots, X'_p)$
    - Calculate conditional probabilities with all the normal distributions
    - Apply the MAP rule to make a decision

Naïve Gaussian means ?

Not Naïve

$$P(X_1, X_2, \cdots, X_p \mid C) =$$

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

Naïve

$$P(X_1, X_2, \cdots, X_p \mid C = c_j) = P(X_1 \mid C)P(X_2 \mid C) \cdots P(X_p \mid C)$$

$$= \prod_i \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

Diagonal Matrix

$$\Sigma\_j = \Lambda\_j$$

Each class' covariance matrix is diagonal

Today : Relevant classifiers / KNN / LOOCV

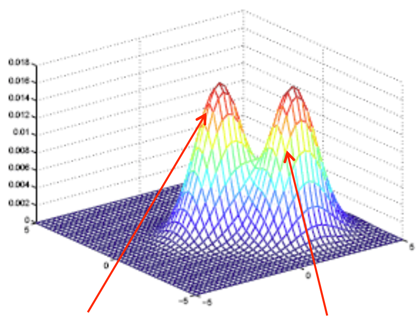
- ✓ Logistic regression (cont.)
- ✓ Gaussian Naïve Bayes Classifier
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA, RDA
  - Discriminative vs. Generative
- ✓ K-nearest neighbor ,
- ✓ LOOCV

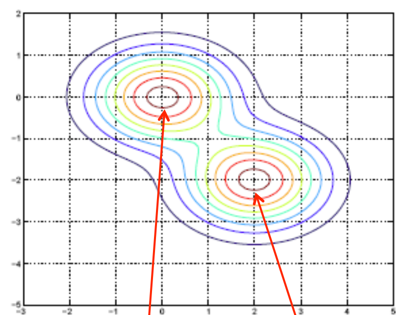
If covariance matrix not Identity but same e.g. ➔ LDA (Linear Discriminant Analysis)

Linear Discriminant Analysis : $\Sigma_k = \Sigma, \forall k$

Each class' covariance matrix is the same

The Gaussian Distribution are shifted versions of each other





Optimal Classification

$$\underset{k}{\operatorname{argmax}} P(C\_k \mid X) = \underset{k}{\operatorname{argmax}} P(X, C) = \underset{k}{\operatorname{argmax}} P(X \mid C) P(C)$$

$$= \arg\max_k \Big[\cancel{-\log((2\pi)^{p/2}|\Sigma|^{1/2})} - \frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k) + \log(\pi_k) \Big]$$

$$= \arg\max_k \boxed{\Big[-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k) + \log(\pi_k) \Big]}$$

Linear Discriminant Function for LDA

- Note

$$-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k) = x^T \Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T \Sigma^{-1}\mu_k - \frac{1}{2}x^T \Sigma^{-1} x$$

Define **Linear Discriminant Function**

$$\delta(x) = -\frac{1}{2}(x-\mu_k)^T \sum^{-1}(x-\mu_k) + log\pi_k$$

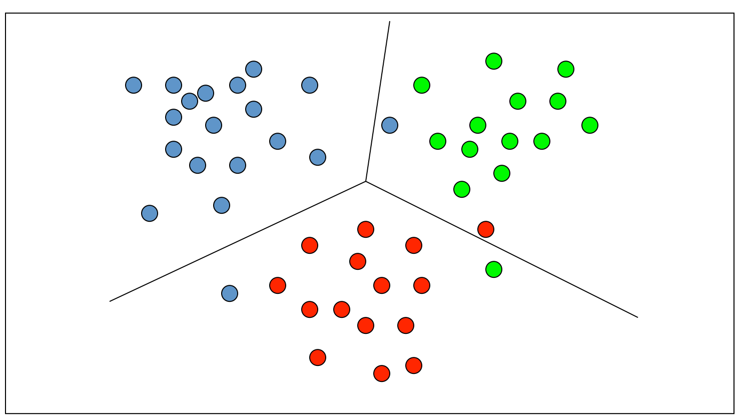
➔ The Decision Boundary Between class k and l, $\{x : \delta_k(x) = \delta_l(x)\}$, is linear

$$\log\frac{P(C\_k \mid X)}{P(C\_l \mid X)} = \log\frac{P(X \mid C\_k)}{P(X \mid C\_l)} + \log\frac{P(C\_k)}{P(C\_l)}$$

$$= \log\frac{\pi_k}{\pi_\ell} - \frac{1}{2}(\mu_k + \mu_\ell)^T \Sigma^{-1}(\mu_k - \mu_\ell) + x^T \Sigma^{-1}(\mu_k - \mu_\ell), \quad (4.9)$$

Boundary points X : when P(c_k|X) == P(c_l|X), the left linear equation ==0, a linear line

Visualization (three classes)



If covariance matrix not Identity not same e.g. ➔ QDA (Quadratic Discriminant Analysis)

- Estimate the covariance matrix Σ_k separately for each class k, $k = 1, 2, ..., K$.
- *Quadratic discriminant function*:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k \ .$$

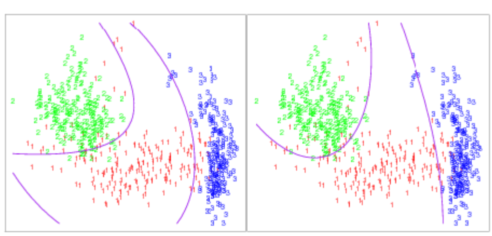
- Classification rule:

$$\hat{G}(x) = \arg\max_k \delta_k(x) \ .$$

- Decision boundaries are quadratic equations in x.
- QDA fits the data better than LDA, but has more parameters to estimate.

LDA on Expanded Basis

- Expand input space to include X_1X_2, X_1^2, and X_2^2.
- Input is five dimensional: $X = (X_1, X_2, X_1X_2, X_1^2, X_2^2)$.



LDA with quadratic basis Versus QDA

Figure 4.6: *Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space $x_1, x_2, x_{12}, x_1^2, x_2^2$). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.*

Regularized Discriminant Analysis

- A compromise between LDA and QDA.
- Shrink the separate covariances of QDA toward a common covariance as in LDA.
- Regularized covariance matrices:

$$\hat{\Sigma}_k(\alpha) = \alpha \hat{\Sigma}_k + (1 - \alpha)\hat{\Sigma} .$$

- The quadratic discriminant function $\delta_k(x)$ is defined using the shrunken covariance matrices $\hat{\Sigma}_k(\alpha)$.
- The parameter α controls the complexity of the model.

Today : Relevant classifiers / KNN / LOOCV

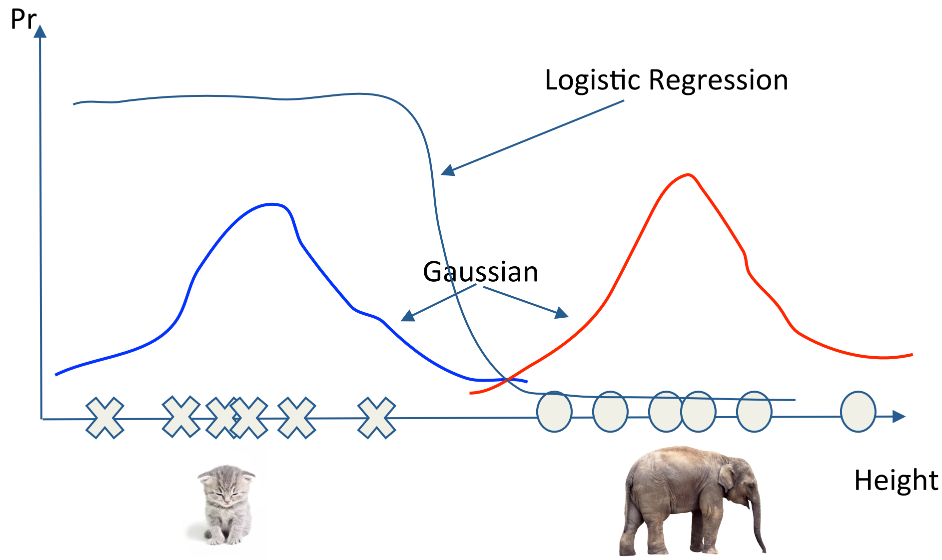
- ✓ Logistic regression (cont.)
- ✓ Gaussian Naïve Bayes Classifier
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA
  - Discriminative vs. Generative
- ✓ K-nearest neighbor
- ✓ LOOCV

LDA vs. Logistic Regression

- **LDA (Generative model)**
  - Assumes Gaussian class-conditional densities and a common covariance
  - Model parameters are estimated by maximizing the full log likelihood, parameters for each class are estimated independently of other classes, $K_p + \frac{p(p+1)}{2} + (K-1)$ parameters
  - Makes use of marginal density information $\Pr(x)$
  - Easier to train, low variance, more efficient if model is correct
  - Higher asymptotic error, but converges faster

- **Logistic Regression (Discriminative model)**
  - Assumes class-conditional densities are members of the (same) exponential family distribution
  - Model parameters are estimated by maximizing the conditional log likelihood, simultaneous consideration of all other classes, $(K-1)(p+1)$parameters
  - Ignores marginal density information $\Pr(x)$
  - Harder to train, robust to uncertainty about the data generation process
  - Lower asymptotic error, but converges more slowly

Discriminative vs. Generative



Discriminative vs. Generative

- Definitions
  - h_{gen} and h_{dis}: generative and discriminative classifiers
  - $h_{gen, inf}$ and $h_{dis, inf}$: same classifiers but trained on the entire population (asymptotic classifiers)
  - n → infinity, h_{gen} → $h_{gen, inf}$ and h_{dis} → $h_{dis, inf}$

Ng, Jordan,. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." *Advances in neural information processing systems* 14 (2002): 841.

Discriminative vs. Generative

Proposition 1:

$$\epsilon\left(h_{dis,\inf}\right) \leq \epsilon\left(h_{gen,\inf}\right)$$

Proposition 2:

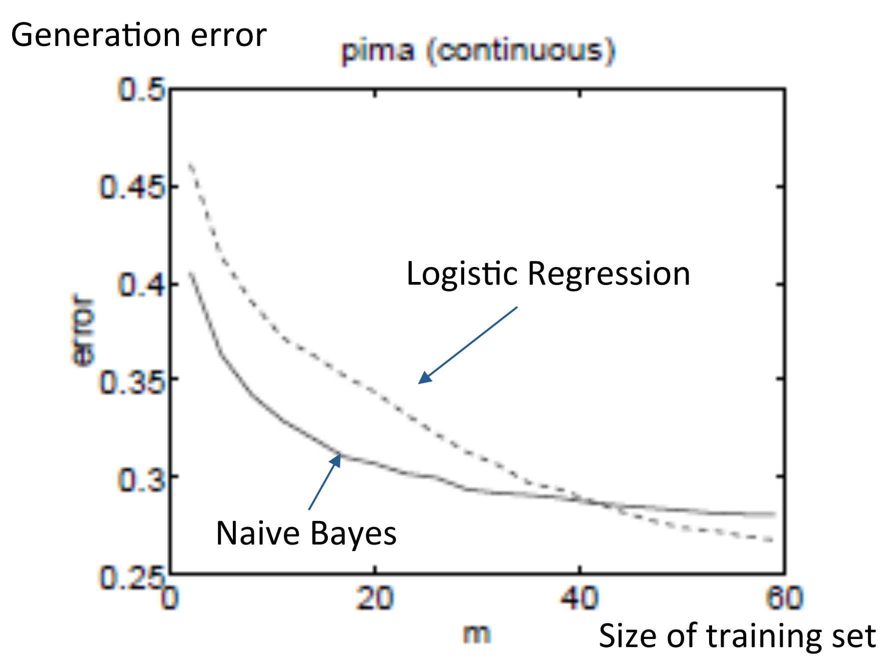
$$\epsilon\left(h_{dis}\right) \leq \epsilon\left(h_{dis,\inf}\right) + O\left(\sqrt{(\frac{p}{n} * \log(\frac{n}{p}))}\right)$$

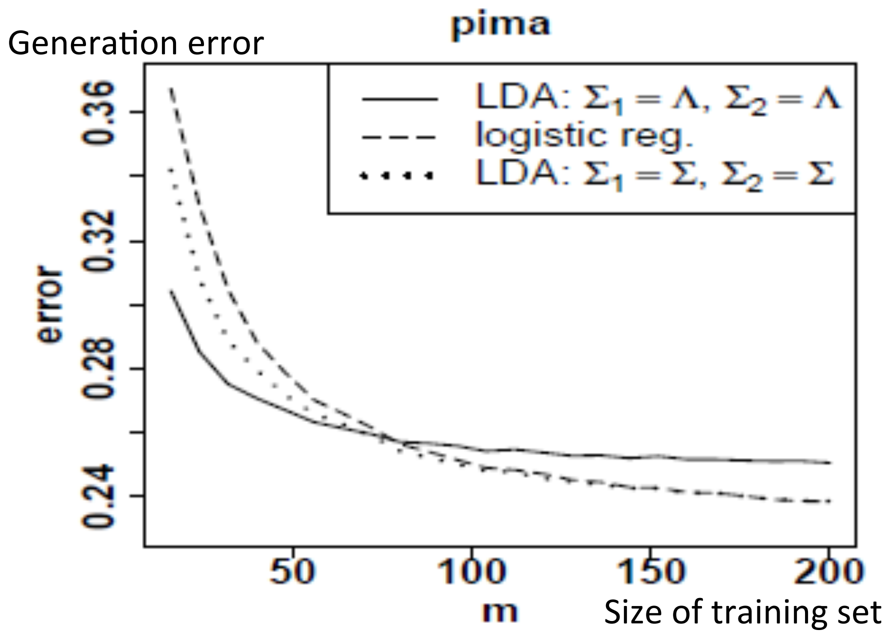
- p : number of dimensions
- n : number of observations
- ϵ : generalization error

Logistic Regression vs. NBC

Discriminative classifier (Logistic Regression)
- Smaller asymptotic error
- Slow convergence ~ size of training set O(p)

Generative classifier (Naive Bayes)
- Larger asymptotic error
- Can handle missing data (EM)
- Fast convergence ~ size of training set O(lg(p))





Xue, Jing-Hao, and D. Michael Titterington. "Comment on "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes"."*Neural processing letters* 28.3 (2008): 169-187.

Logistic Regression vs. NBC

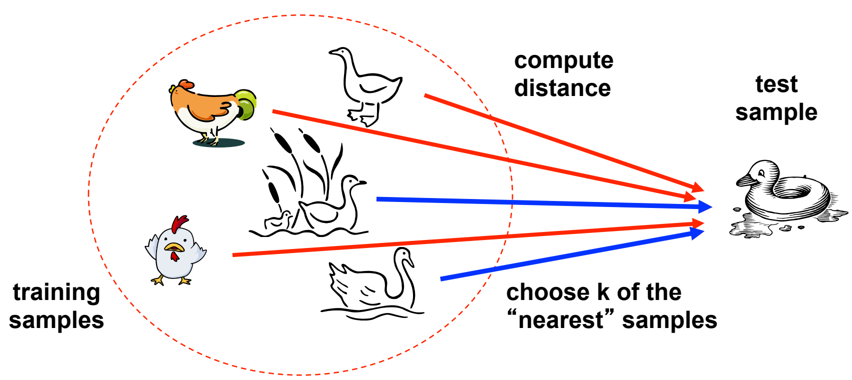
- Empirically, generative classifiers approach their asymptotic error faster than discriminative ones
  - Good for small training set
  - Handle missing data well (EM)
- Empirically, discriminative classifiers have lower asymptotic error than generative ones
  - Good for larger training set

Today : Generative vs. Discriminative / KNN / LOOCV

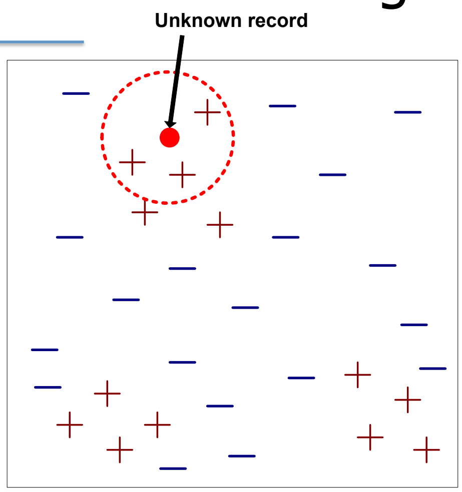
- ✓ Logistic regression (cont.)
- ✓ Gaussian Naïve Bayes Classifier
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA
  - Discriminative vs. Generative
- ✓ K-nearest neighbor ,
- ✓ LOOCV

Nearest neighbor classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



Nearest neighbor classifiers

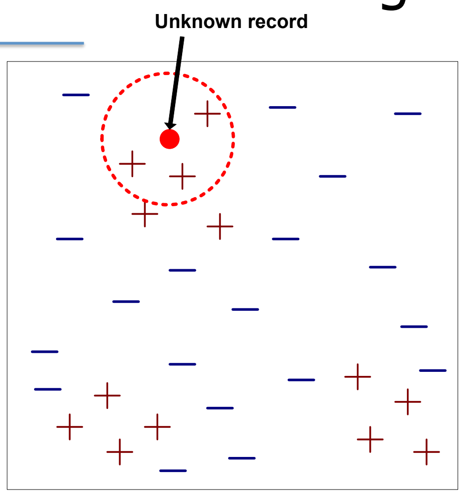


Requires three inputs:

1. The set of stored training samples
2. Distance metric to compute distance between samples
3. The value of *k*, i.e., the number of nearest neighbors to retrieve

Yanjun Qi / UVA CS 4501-01-6501-07

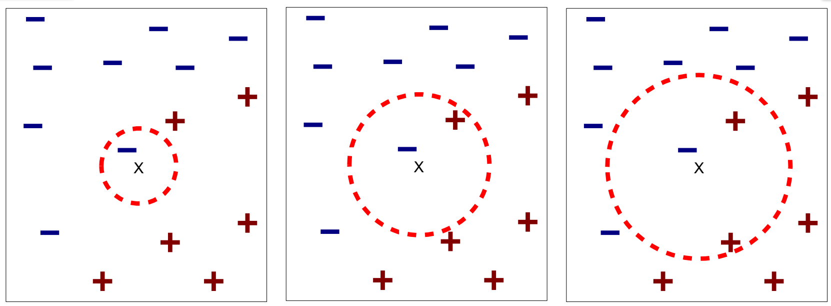
Nearest neighbor classifiers



To classify unknown sample:

1. Compute distance to other training records
2. Identify *k* nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of nearest neighbor

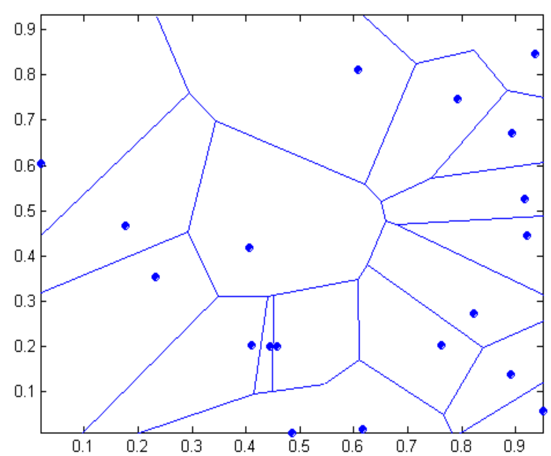


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

k-nearest neighbors of a sample x are datapoints that have the *k* smallest distances to x

1-nearest neighbor

Voronoi diagram



Nearest neighbor classification

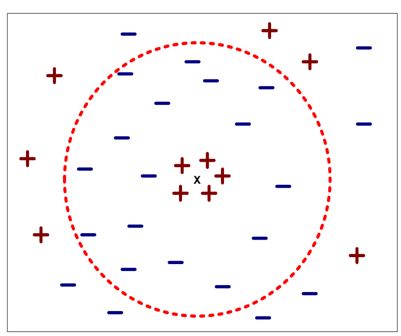
- Compute distance between two points:
  - For instance, Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Options for determining the class from nearest neighbor list
  - Take majority vote of class labels among the *k*-nearest neighbors
  - Weight the votes according to distance
    - example: weight factor w = 1 / d^2

Nearest neighbor classification

- Choosing the value of *k*:
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include points from other classes



Nearest neighbor classification

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5 m to 1.8 m
    - weight of a person may vary from 90 lb to 300 lb
    - income of a person may vary from $10K to $1M

Yanjun Qi / UVA CS 4501-01-6501-07

Nearest neighbor classification…

- Problem with Euclidean measure:
  - High dimensional data
    - curse of dimensionality
  - Can produce counter-intuitive results

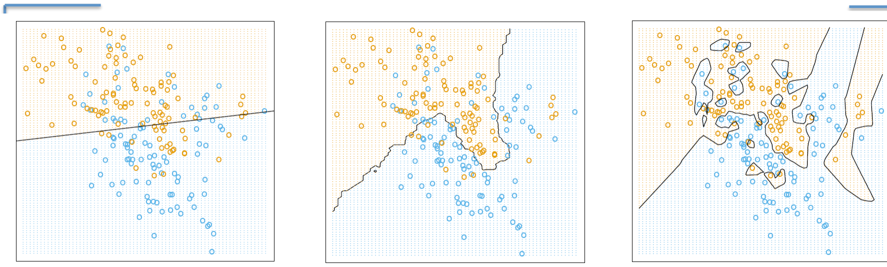
1 1 1 1 1 1 1 1 1 1 1 0	vs	1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1		0 0 0 0 0 0 0 0 0 0 0 1
d = 1.4142		**d = 1.4142**

◆ one solution: normalize the vectors to unit length

Nearest neighbor classification

- *k*-Nearest neighbor classifier is a lazy learner
  - Does not build model explicitly.
  - Unlike eager learners such as decision tree induction and rule-based systems.
  - Classifying unknown samples is relatively expensive.
- *k*-Nearest neighbor classifier is a local model, vs. global model of linear classifiers.

Decision boundaries in global vs. local models



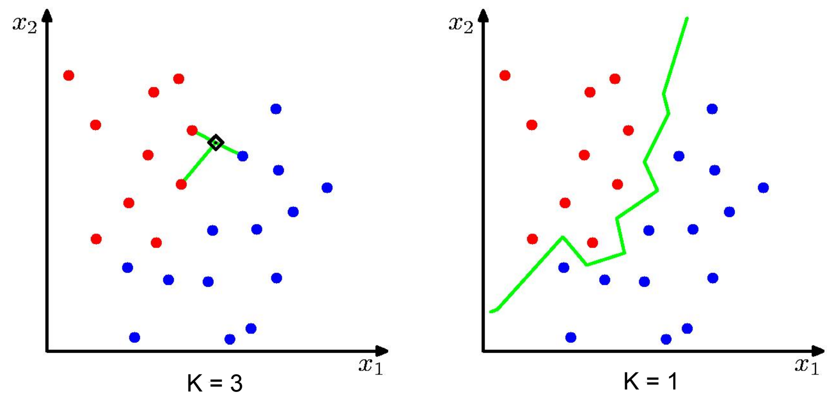
linear regression | 15-nearest neighbor | 1-nearest neighbor

- global
- stable
- can be inaccurate

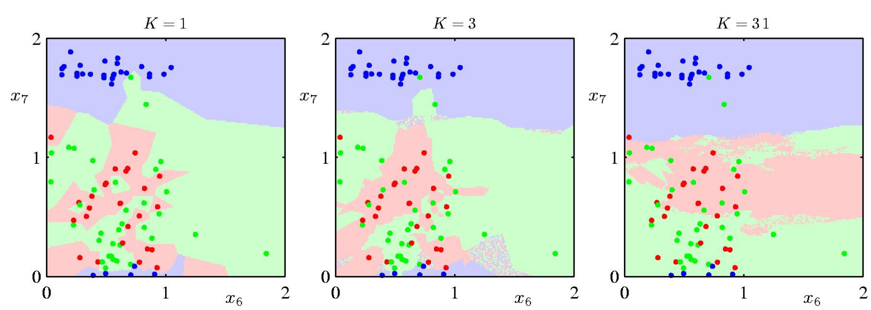
- local
- accurate
- unstable

What ultimately matters: ***GENERALIZATION***

K-Nearest-Neighbours for Classification (2)



K-Nearest-Neighbours for Classification (3)



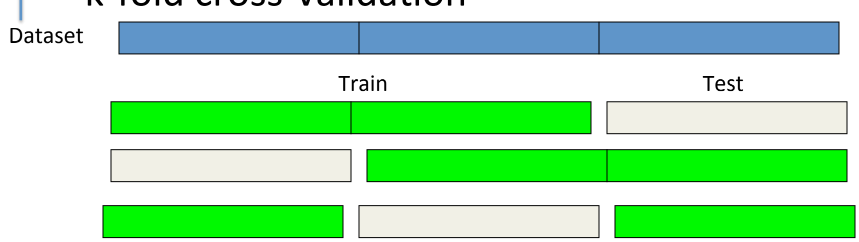
- K acts as a smother
- For $N \to \infty$, the error rate of the 1-nearest-neighbour classifier is never more than twice the optimal error (obtained from the true conditional class distributions).

Today : Generative vs. Discriminative / KNN / LOOCV

- ✓ Logistic regression (cont.)
- ✓ Gaussian Naïve Bayes Classifier
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA
  - Discriminative vs. Generative
- ✓ K-nearest neighbor ,
- ✓ LOOCV

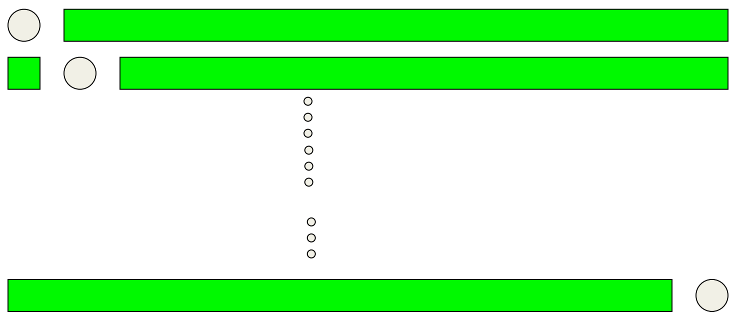
cross-validation (e.g. K=3)

- k-fold cross-validation



Common Splitting Strategies

- Leave-one-out (n-fold cross validation)



Leave-one-out cross validation

- **Leave-one-out cross validation (LOOCV)** is K-fold cross validation taken to its logical extreme, with K equal to n, the number of data points in the set.
- That means that n separate times, the function optimization is trained on all the data except for one point and a prediction is made for that point.
- As before the average error is computed and used to evaluate the model.

CV-based Model Selection
We're trying to decide which algorithm to use.

- We train each machine and make a table...

i	f_i	TRAINERR	10-FOLD-CV-ERR	Choice
1	f_1			
2	f_2			
3	f_3			√
4	f_4			
5	f_5			
6	f_6			

Which kind of cross-validation ?

	Downside	Upside
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive. Has some weird behavior	Doesn't waste data
10-fold	Wastes 10% of the data. 10 times more expensive than test set	Only wastes 10%. Only 10 times more expensive instead of R times.
3-fold	Wastier than 10-fold. Expensivier than test set	Slightly better than test-set
R-fold	Identical to Leave-one-out	

Today Recap: Generative vs. Discriminative / KNN / LOOCV

- ✓ Logistic regression (cont.)
- ✓ Gaussian Naïve Bayes Classifier
  - Gaussian distribution
  - Gaussian NBC
  - LDA, QDA
  - Discriminative vs. Generative
- ✓ K-nearest neighbor ,
- ✓ LOOCV

References

- ❑ Prof. Tan, Steinbach, Kumar's "Introduction to Data Mining" slide
- ❑ Prof. Andrew Moore's slides
- ❑ Prof. Eric Xing's slides
- ❑ Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.