Machine Learning

10-701/15-781, Fall 2008

Theory of Classification
and
Nonparametric Classifier

Eric Xing
Lecture 2, September 10, 2008

Reading: Chap. 2.5 CB and handouts

Miscellaneous

- TA introduction!
- Recitation time/location to be changed, now 6-7pm here!
- Did you receive our announcements?
- Logistic questions?
Classification

- Representing data:

\[ X = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^K \end{bmatrix} \]

- Hypothesis (classifier)

\[ f(x_1^n, x_2^n, \cdots, x_K^n) \]

Outline

- What is theoretically the best classifier
  - Probabilistic theory of classification
  - Discrete density estimation and Bayesian theorem
  - Bayesian decision rule for Minimum Error

- Nonparametric Classifier (Instance-based learning)
  - Nonparametric density estimation
  - K-nearest-neighbor classifier
  - Optimality of kNN
  - Problem of kNN
Decision-making as dividing a high-dimensional space

- Distributions of samples from normal and abnormal cells

\[ h_1(x) = p(x_1 | x) \]
\[ h_2(x) = p(x_2 | x) \]

Density Estimation

- A Density Estimator learns a mapping from a set of attributes to a Probability

- Often know as parameter estimation if the distribution form is specified
  - Binomial, Gaussian ...

- Four important issues:
  - Nature of the data (iid, correlated, ...)
  - Objective function (MLE, MAP, ...)
  - Algorithm (simple algebra, gradient methods, EM, ...)
  - Evaluation scheme (likelihood on test data, predictability, consistency, ...)
Density Estimation Schemes

- **Data**
  - \((x_1^i, \ldots, x_n^i)\)
  - \((x_2^i, \ldots, x_m^i)\)
  - \(\ldots\)
  - \((x_M^i, \ldots, x_N^i)\)
  - iid
  - correlated

- **Objective functions**
  - **Maximum likelihood**
  - **Bayesian**
  - **Conditional likelihood**
  - **Margin**

- **Algorithm**
  - Analytical
  - Gradient
  - EM
  - Sampling

- **Score param**
  - Analytical: \(10^{-5}\)
  - Gradient: \(10^{-3}\)
  - EM: \(10^{-15}\)

Parameter Learning from iid Data

- Goal: estimate distribution parameters \(\theta\) from a dataset of \(N\) independent, identically distributed (iid), fully observed, training cases

\[ D = \{x_1, \ldots, x_N\} \]

- **Maximum likelihood estimation (MLE)**
  1. One of the most common estimators
  2. With iid and full-observability assumption, write \(L(\theta)\) as the likelihood of the data:

\[
L(\theta) = P(x_1, x_2, \ldots, x_N; \theta) = P(x_1; \theta)P(x_2; \theta)\ldots, P(x_N; \theta) = \prod_{i=1}^{N} P(x_i; \theta)
\]

3. pick the setting of parameters most likely to have generated the data we saw:

\[
\theta^* = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} \log L(\theta)
\]
Example: Bernoulli model

- **Data:**
  - We observed \( N \) iid coin tossing: \( D = \{1, 0, 1, \ldots, 0\} \)

- **Representation:**
  Binary r.v.
  \( x = \{0, 1\} \)

- **Model:**
  \[
  P(x) = \begin{cases} 
  1 - \theta & \text{for } x = 0 \\
  \theta & \text{for } x = 1 
  \end{cases} 
  \Rightarrow P(x) = \theta^x (1 - \theta)^{1-x}
  \]

- **How to write the likelihood of a single observation \( x_i \)?**
  \[
  P(x_i) = \theta^{x_i} (1 - \theta)^{1-x_i}
  \]

- **The likelihood of dataset \( D = \{x_1, \ldots, x_N\} \):**
  \[
  P(x_1, x_2, \ldots, x_N | \theta) = \prod_{i=1}^{N} P(x_i | \theta) = \prod_{i=1}^{N} \left( \theta^{x_i} (1 - \theta)^{1-x_i} \right) = \theta^{\sum_{i=1}^{N} x_i} (1 - \theta)^{N - \sum_{i=1}^{N} x_i} = \theta^{\text{heads}} (1 - \theta)^{\text{tails}}
  \]

---

Maximum Likelihood Estimation

- **Objective function:**
  \[
  \ell(\theta; D) = \log P(D | \theta) = \log \theta^{n_h} (1 - \theta)^{n_t} = n_h \log \theta + (N - n_h) \log (1 - \theta)
  \]

- **We need to maximize this w.r.t. \( \theta \)**

- **Take derivatives wrt \( \theta \)**
  \[
  \frac{\partial \ell}{\partial \theta} = \frac{n_h}{\theta} - \frac{N - n_h}{1 - \theta} = 0 \Rightarrow \hat{\theta}_{MLE} = \frac{n_h}{N} \quad \text{or} \quad \hat{\theta}_{MLE} = \frac{1}{N} \sum_i x_i
  \]

- **Sufficient statistics**
  - The counts, \( n_h \), where \( n_h = \sum x_i \), are sufficient statistics of data \( D \)
Overfitting

- Recall that for Bernoulli Distribution, we have

\[ \hat{\theta}_{ML}^{\text{head}} = \frac{n^{\text{head}}}{n^{\text{head}} + n^{\text{tail}}} \]

- What if we tossed too few times so that we saw zero head? We have \( \hat{\theta}_{ML}^{\text{head}} = 0 \), and we will predict that the probability of seeing a head next is zero!!!

- The rescue: "smoothing"
  - Where \( n' \) is known as the pseudo-(imaginary) count

\[ \hat{\theta}_{ML}^{\text{head}} = \frac{n^{\text{head}} + n'}{n^{\text{head}} + n^{\text{tail}} + n'} \]

- But can we make this more formal?

Bayesian Parameter Estimation

- Treat the distribution parameters \( \theta \) also as a random variable
- The a posteriori distribution of \( \theta \) after seeing the data is:

\[
p(\theta | D) = \frac{p(D | \theta) p(\theta)}{p(D)} = \frac{p(D | \theta) p(\theta)}{\int p(D | \theta) p(\theta) d\theta}
\]

This is Bayes Rule

\[
\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}
\]


The prior \( p(.) \) encodes our prior knowledge about the domain
Frequentist Parameter Estimation

Two people with different priors \( p(\theta) \) will end up with different estimates \( p(\theta | D) \).

- Frequentists dislike this “subjectivity”.
- Frequentists think of the parameter as a fixed, unknown constant, not a random variable.
- Hence they have to come up with different "objective" estimators (ways of computing from data), instead of using Bayes’ rule.
  - These estimators have different properties, such as being "unbiased", "minimum variance", etc.
  - The maximum likelihood estimator is one such estimator.

Discussion

\( \theta \) or \( p(\theta) \), this is the problem!
Discussion

\( \theta \) or \( p(\theta) \), this is the problem!

Bayesian estimation for Bernoulli

- **Beta distribution:**
  \[
  P(\theta; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1} = B(\alpha, \beta)\theta^{\alpha-1}(1-\theta)^{\beta-1}
  \]
  - When \( x \) is discrete
    \[\Gamma(x + 1) = x\Gamma(x) = x!\]

- **Posterior distribution of \( \theta \):**
  \[
  P(\theta | x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n | \theta) p(\theta)}{P(x_1, \ldots, x_n)} \propto \theta^n (1-\theta)^n \times \theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{\alpha+n-1}(1-\theta)^{\beta+n-1}
  \]
  - Notice the isomorphism of the posterior to the prior,
  - such a prior is called a **conjugate prior**
  - \( \alpha \) and \( \beta \) are hyperparameters (parameters of the prior) and correspond to the number of "virtual" heads/tails (pseudo counts)
Bayesian estimation for Bernoulli, con’d

- Posterior distribution of $\theta$:
  $$P(\theta | x_1, \ldots, x_N) = \frac{p(x_1, \ldots, x_N | \theta) \times \theta^n (1 - \theta)^{n - 1}}{p(x_1, \ldots, x_N)} = \theta^{n+\alpha-1} (1 - \theta)^{n + \beta - 1}$$

- Posterior mean estimation:
  $$\theta_{\text{mean}} = \int \theta p(\theta | D) d\theta = C \int \theta \times \theta^{n+\alpha-1} (1 - \theta)^{n + \beta - 1} d\theta = \frac{n_h + \alpha}{N + \alpha + \beta}$$

- Maximum a posteriori (MAP) estimation:
  $$\theta_{\text{MAP}} = \arg \max_{\theta} \log P(\theta | x_1, \ldots, x_N)$$

- Prior strength: $A = \alpha + \beta$
  - $A$ can be interoperated as the size of an imaginary data set from which we obtain the pseudo-counts.

Effect of Prior Strength

- Suppose we have a uniform prior ($\alpha = \beta = 1/2$), and we observe $\tilde{n} = (n_h = 2, n_r = 8)$
- Weak prior $A = 2$. Posterior prediction:
  $$p(x = h | n_h = 2, n_r = 8, \tilde{\alpha} = \tilde{\alpha} \times 2) = \frac{1 + 2}{2 + 10} = 0.25$$

- Strong prior $A = 20$. Posterior prediction:
  $$p(x = h | n_h = 2, n_r = 8, \tilde{\alpha} = \tilde{\alpha} \times 20) = \frac{10 + 2}{20 + 10} = 0.40$$

- However, if we have enough data, it washes away the prior. e.g., $\tilde{n} = (n_h = 200, n_r = 800)$. Then the estimates under weak and strong prior are $\frac{1 + 200}{2 + 1000}$ and $\frac{10 + 200}{20 + 1000}$, respectively, both of which are close to 0.2.
Continuous Distributions

- Uniform Probability Density Function
  \[ p(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{elsewhere} \end{cases} \]

- Normal (Gaussian) Probability Density Function
  \[ p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]
  - The distribution is symmetric, and is often illustrated as a bell-shaped curve.
  - Two parameters, \( \mu \) (mean) and \( \sigma \) (standard deviation), determine the location and shape of the distribution.
  - The highest point on the normal curve is at the mean, which is also the median and mode.
  - The mean can be any numerical value: negative, zero, or positive.

- Multivariate Gaussian
  \[ p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu) \right) \]

Class-Conditional Probability

- Classification-specific Dist.: \( P(X|Y) \)
  \[ p(X | Y = 1) = p_1(X; \mu_1, \Sigma_1) \]
  \[ p(X | Y = 2) = p_2(X; \mu_2, \Sigma_2) \]

- Class prior (i.e., “a priori weight”): \( P(Y) \)
The Bayes Rule

What we have just did leads to the following general expression:

\[ P(Y \mid X) = \frac{P(X \mid Y)p(Y)}{P(X)} \]

This is Bayes Rule

---

The Bayes Decision Rule for Minimum Error

- The \textit{a posteriori} probability of a sample
  \[ P(Y = i \mid X) = \frac{P(X \mid Y = i)P(Y = i)}{P(X)} = \frac{\pi_i p_i(X \mid Y = i)}{\sum_i \pi_i p_i(X \mid Y = i)} = q_i(X) \]
- Bayes Test:
  \[ q_i(x) \geq q_j(x) \quad \rightarrow \quad \frac{\pi_i p_i(x)}{\pi_j p_j(x)} \leq \frac{\pi_j p_j(x)}{\pi_i p_i(x)} \]
- Likelihood Ratio:
  \[ \hat{\ell}(X) = \frac{\pi_i p_i(x)}{\pi_j p_j(x)} \]
- Discriminant function:
  \[ h(X) = \ln \hat{\ell}(X) = \ln \left( \frac{\pi_i p_i(x)}{\pi_j p_j(x)} \right) \geq \mathbf{w}_i^T \mathbf{a} - \mathbf{w}_j^T \mathbf{a} \]
Example of Decision Rules

- When each class is a normal ...

- We can write the decision boundary analytically in some cases ... homework!!

Bayes Error

- We must calculate the probability of error
  - the probability that a sample is assigned to the wrong class
- Given a datum $X$, what is the risk?
  \[ r(X) = \min \{ q_1(X), q_2(X) \} \]

- The Bayes error (the expected risk):
  \[
  \epsilon = E[r(X)] = \int r(x)p(x)dx = \int \min[p_1(x), p_2(x)]dx = \pi_1 \int_{L_1} p_1(x)dx + \pi_2 \int_{L_2} p_2(x)dx = \pi_1 p_{L_1} + \pi_2 p_{L_2}
  \]
More on Bayes Error

- Bayes error is the lower bound of probability of classification error

\[ E_{\text{Bayes}} = \frac{1}{2} \log \left( \frac{p_1}{p_2} \right) \]

- Bayes classifier is the theoretically best classifier that minimizes probability of classification error

- Computing Bayes error is in general a very complex problem. Why?
  - Density estimation:

  Integrating density function:

  \[ r_1 = \int_{-\infty}^{+\infty} p_1(x)dx \]
  \[ c_3 = \int_{-\infty}^{+\infty} p_3(x)dx \]

Learning Classifier

- The decision rule:

\[ h(X) = -\ln p_1(X) + \ln p_2(X) > \ln \frac{\pi_1}{\pi_2} \]

- Learning strategies
  - Generative Learning
    - Parametric
    - Nonparametric
  - Discriminative Learning
    - Parametric
    - Nonparametric
  - Instance-based Learning (Store all past experience in memory)
    - A special case of nonparametric classifier
Supervised Learning

- K-Nearest-Neighbor Classifier:
  where the \( h(X) \) is represented by all the data, and by an algorithm

Recall: Vector Space Representation

- Each document is a vector, one component for each term (= word).

<table>
<thead>
<tr>
<th></th>
<th>Doc 1</th>
<th>Doc 2</th>
<th>Doc 3</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word 1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>Word 2</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>Word 3</td>
<td>12</td>
<td>1</td>
<td>10</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>

- Normalize to unit length.
- High-dimensional vector space:
  - Terms are axes, 10,000+ dimensions, or even 100,000+
  - Docs are vectors in this space
Classes in a Vector Space

Test Document = ?
K-Nearest Neighbor (kNN) classifier

kNN Is Close to Optimal

- Cover and Hart 1967
- Asymptotically, the error rate of 1-nearest-neighbor classification is less than twice the Bayes rate [error rate of classifier knowing model that generated data]
- In particular, asymptotic error rate is 0 if Bayes rate is 0.
- Decision boundary:
Where does kNN come from?

- How to estimation \( p(X) \)?
- Nonparametric density estimation
  - Parzen density estimate
    
    E.g. (Kernel density est.):  
    
    \[
    \hat{p}(X) = \frac{1}{N} \sum_{i=1}^{N} k(X - x_i),
    \]
    
    More generally:  
    
    \[
    \hat{p}(X) = \frac{1}{N} \frac{k(X)}{V}
    \]

Where does kNN come from?

- Nonparametric density estimation
  - Parzen density estimate  
    
    \[
    \hat{p}(X) = \frac{1}{N} \frac{k(X)}{V}
    \]
  - kNN density estimate  
    
    \[
    \hat{p}(X) = \frac{1}{N} \frac{(k - 1)}{V(X)}
    \]

- Bayes classifier based on kNN density estimator:
  
  \[
  \frac{h(X)}{p_1(X)} = -\ln \frac{p_1(X)}{p_2(X)} = -\ln \frac{(k_1 - 1)N_2V_2(X)}{(k_2 - 1)N_1V_1(X)} > \ln \frac{N_1}{N_2}
  \]

- Voting kNN classifier
  
  Pick \( K_1 \) and \( K_2 \) implicitly by picking \( K_1 + K_2 = K, V_1 = V_2, N_1 = N_2 \)
Voting kNN

- The procedure

Asymptotic Analysis

- Condition risk: $r_k(X, X_{NN})$
  - Test sample $X$
  - NN sample $X_{NN}$
  - Denote the event $X$ is class I as $X \rightarrow I$

  Assuming $k = 1$

  \[
  r_1(X, X_{NN}) = Pr \left\{ \begin{array}{ll} 
  X \rightarrow 1 & \text{or} \ X \rightarrow 2 & \text{or} \ X_{NN} \rightarrow 1 \end{array} \right| X_{NN}
  \]

  \[
  = Pr \left\{ \begin{array}{ll} 
  X \rightarrow 1 & \text{or} \ X_{NN} \rightarrow 2 \end{array} \right| X_{NN} \right| X_{NN}
  \]

  \[
  = q_1(X)q_2(X_{NN}) + q_2(X)q_1(X_{NN})
  \]

- When an infinite number of samples is available, $X_{NN}$ will be so close to $X$

  \[
  r_1^*(X) = 2q_1(X)q_2(X) = 2\zeta(X)
  \]
Asymptotic Analysis, cont.

- Recall conditional Bayes risk:

\[
\hat{r}^*(X) = \min[q_1(X), q_2(X)]
\]

\[
\frac{1}{2} - \frac{1}{2} \sqrt{1 - 4\hat{r}_2(X)}
\]

\[
\sum_{i=1}^{\infty} \frac{1}{i(i-1)} \epsilon_i(X)
\]

This is called the MacLaurin series expansion.

- Thus the asymptotic condition risk

\[
r_1^*(X) = 2\epsilon(X) \leq 2\epsilon^*(X)
\]

- It can be shown that \( \epsilon_1^* \leq 2\epsilon^* \)

- This is remarkable, considering that the procedure does not use any information about the underlying distributions and only the class of the single nearest neighbor determines the outcome of the decision.

In fact

\[
\frac{1}{2} \epsilon^* \leq \epsilon_{2NN}^* \leq \epsilon_{4NN}^* \leq \ldots \leq \epsilon^* \leq \ldots \leq \epsilon_{3NN}^* \leq \epsilon_{NN}^* \leq 2\epsilon^*
\]

- Example:
Nearest-Neighbor Learning Algorithm

- Learning is just storing the representations of the training examples in $D$.
- Testing instance $x$:
  - Compute similarity between $x$ and all examples in $D$.
  - Assign $x$ the category of the most similar example in $D$.
- Does not explicitly compute a generalization or category prototypes.
- Also called:
  - Case-based learning
  - Memory-based learning
  - Lazy learning

kNN is an instance of Instance-Based Learning

- What makes an Instance-Based Learner?
  - A distance metric
  - How many nearby neighbors to look at?
  - A weighting function (optional)
  - How to relate to the local points?
Euclidean Distance Metric

\[ D(x, x') = \sqrt{\sum_i \sigma_i^2 (x_i - x_i')^2} \]

- Or equivalently,

\[ D(x, x') = \sqrt{(x - x')^T \Sigma (x - x')} \]

- Other metrics:
  - L_1 norm: \|x-x'\|
  - L_\infty norm: \max \|x-x'\| (elementwise ...)
  - Mahalanobis: where \Sigma is full, and symmetric
  - Correlation
  - Angle
  - Hamming distance, Manhattan distance
  - ...

1-Nearest Neighbor (kNN) classifier

- Sports
- Science
- Arts
2-Nearest Neighbor (kNN) classifier

3-Nearest Neighbor (kNN) classifier
5-Nearest Neighbor (kNN) classifier

Case Study: kNN for Web Classification

- Dataset
  - 20 News Groups (20 classes)
  - Download: (http://people.csail.mit.edu/jrennie/20Newsgroups/)
  - 61,118 words, 18,774 documents
  - Class labels descriptions

<table>
<thead>
<tr>
<th>comp.graphics</th>
<th>rec.autos</th>
<th>sci.crypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>comp.os.ms-windows</td>
<td>rec.motorcycles</td>
<td>sci.electronics</td>
</tr>
<tr>
<td>misc.comp.sys</td>
<td>rec.sport.baseball</td>
<td>sci.med</td>
</tr>
<tr>
<td>comp.sys.x86</td>
<td>rec.sport.hockey</td>
<td>sci.space</td>
</tr>
<tr>
<td>comp.windows.x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>misc.forsale</th>
<th>talk.politics.misc</th>
<th>talk.religion.misc</th>
</tr>
</thead>
<tbody>
<tr>
<td>talk.politics.guns</td>
<td>talk.politics.mideast</td>
<td>alt.atheism</td>
</tr>
<tr>
<td>soc.religion.christian</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experimental Setup

- Training/Test Sets:
  - 50%-50% randomly split.
  - 10 runs
  - report average results
- Evaluation Criteria:
  \[
  \text{Accuracy} = \frac{\sum_{i \in \text{test set}} I(\text{predict}_i = \text{true label}_i)}{\# \text{ of test samples}}
  \]

Results: Binary Classes

Accuracy vs. k

- alt.atheism vs. comp.graphics
- rec.autos vs. rec.sport.baseball
- comp.windows.x vs. rec.motorcycles
Results: Multiple Classes

Random select 5-out-of-20 classes, repeat 10 runs and average

Accuracy

All 20 classes

Is kNN ideal?
Is kNN ideal? … more later

Effect of Parameters

- Sample size
  - The more the better
  - Need efficient search algorithm for NN
- Dimensionality
- Density
  - How smooth?
- Metric
  - The relative scalings in the distance metric affect region shapes.
- Weight
  - Spurious or less relevant points need to be downweighted
- K
Sample size and dimensionality

\[ E[\epsilon_{NN}] \approx \epsilon_{NN} + \beta_1 f(X) \]

From page 316, Fukumaga

Neighborhood size

From page 350, Fukumaga
Summary

- **Bayes classifier** is the best classifier which minimizes the probability of classification error.

- Nonparametric and parametric classifier

- A nonparametric classifier does not rely on any assumption concerning the structure of the underlying density function.

- A classifier becomes the **Bayes classifier** if the density estimates converge to the true densities
  - when an infinite number of samples are used
  - The resulting error is the **Bayes error**, the smallest achievable error given the underlying distributions.