Statistical Approaches to Classification and Regression

Professor Michael C. Mozer
CSCI 3202
Correlation

Measure of relation between 2 or more variables

Pearson $r$: linear relation

$r$ in range $-1 \rightarrow +1$ (see figure)

Strength of relationship determined by $r^2$ - proportion of variance accounted for

Artifacts (see figure)

Nonlinear relationships

Can transform data so that it becomes linear, given some hypotheses about nature of nonlinearity

Importance of graphing data
\[ \rho = \frac{\sum x_i y_i - \frac{1}{n} \sum x_i \sum y_i}{\sqrt{\left( \sum x_i^2 - \frac{1}{n} (\sum x_i)^2 \right) \cdot \left( \sum y_i^2 - \frac{1}{n} (\sum y_i)^2 \right)}} \]
BEFORE THE OUTLIER HAS BEEN REMOVED, $r = .88$
Linear regression

Use one variable to predict the value of another

\[ \hat{y} = a + bx \]

- **dependent variable**
- **independent variable**
- **regression coefficients**

Least squares regression: minimize \( \sum (\hat{y}_i - y_i)^2 \)

- **residual**

Correlation coefficient: how small are residuals?

\[ r = (1 - \text{residual variability/original variance})^{1/2} \]

Multiple linear regression

\[ \hat{y} = a + b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots \]
Parametric nonlinear regression

If nature of nonlinear relationship is known, can create new variables that are linearly related to the independent variable.

E.g., polynomial regression
\[ y = f(x, x^2) \]
\[ \Rightarrow x_1 = x \quad x_2 = x^2 \]
\[ \hat{y} = a + b_1 x_1 + b_2 x_2 \]

E.g., exponential regression
\[ y = \exp(a + bx) \]
\[ \Rightarrow y' = \log y \]
\[ \hat{y}' = a + bx \]

E.g., basis function regression
\[ x_1 = f_1(x), \quad x_2 = f_2(x), \quad x_3 = f_3(x) \]
\[ \hat{y} = a + b_1 x_1 + b_2 x_2 + b_3 x_3 \]

\[ y_i(x) = e^{-(x - m_i)^2} \]
Parameteric nonlinear regression (cont.)

e.g., logit regression

dependent variable in range 0-1
(e.g., probabilities)

\[ y' = \log \left( \frac{n}{1-n} \right) \]

\[ \hat{y}' = a + bx \quad \Rightarrow \quad \hat{y} = \frac{1}{1 + e^{-(a+bx)}} \]

Same as neural net with no hidden units

e.g., piecewise linear models
Classification

Treat as a regression problem

\[ y = a + b_1 x_1 + b_2 x_2 + \ldots \] where \( y = +1 \) for class 1
-1 for class 2

Fischer linear discriminant analysis

Project data onto a line that minimizes within-class variability and maximizes between-class variability

\[ F = b_1 x_1 + b_2 x_2 \]

Fisher is same as linear regression if we use these targets for the regression:

\[ y = \frac{n_2 + n_i}{n_2} \] for class 1

\[ = -\frac{n_2 + n_i}{n_2} \] for class 2
**Gaussian Noise Distribution**

Assume data in a class is lumped according to a Gaussian probability distribution.

\[ p(x | c_i) = \frac{1}{(2\pi \sigma_i^2)^{d/2}} \exp \left[-\frac{1}{2\sigma_i^2} (x - \bar{x}_i)^T \Sigma_i^{-1} (x - \bar{x}_i) \right] \]

- **Simple case - uniform spread**: \( \Sigma_i \) is a scalar
- **General case**: \( \Sigma_i \) is a covariance matrix

where \( \bar{x}_i \) is the mean and \( \Sigma_i \) is the covariance matrix for class \( c_i \).
Bayes classifier

Choose class $c_1$ if $p(c_1|x) > p(c_2|x)$

Equivalently, $\frac{p(c_1|x)}{p(c_2|x)} > 1$

\[
\frac{p(c_1|x)}{p(c_2|x)} = \frac{\left[ p(x|c_1)p(c_1)/p(x) \right]}{\left[ p(x|c_2)p(c_2)/p(x) \right]}
\]

\[
= \frac{p(x|c_1)}{p(x|c_2)} \frac{p(c_1)}{p(c_2)} > 1
\]

$p(x|c_1)$ and $p(x|c_2)$ computed assuming Gaussian distribution, w.r. mean & covariance estimated from data

$p(c_1)$ and $p(c_2)$ estimated from data

Recognition becomes the problem of density estimation — $p(x|c_1)$
Two category example.

2i: decision to label $x$ instance of class 1
8ii: decision to label $x$ instance of class 2

$$R(a_1, x) = \lambda_1 \cdot p(C_1 | x) + \lambda_{21} \cdot p(C_2 | x)$$

$$R(a_2, x) = \lambda_{12} \cdot p(C_1 | x) + \lambda_2 \cdot p(C_2 | x)$$

Bayes decision rule: Choose $a_2$ if

$$R(a_1, x) < R(a_2, x)$$

$$\lambda_1 \cdot p(C_1 | x) + \lambda_{21} \cdot p(C_2 | x) < \lambda_{12} \cdot p(C_1 | x) + \lambda_2 \cdot p(C_2 | x)$$

$$(\lambda_{21} - \lambda_2) \cdot p(C_1 | x) < (\lambda_{12} - \lambda_1) \cdot p(C_1 | x)$$

$$\frac{(\lambda_{21} - \lambda_2)}{(\lambda_{12} - \lambda_1)} \cdot \frac{p(C_2 | x)}{p(C_1 | x)} < 1$$

$$(\lambda_{21} - \lambda_2) \cdot \frac{p(x | C_2)}{p(x | C_1)} \cdot \frac{p(C_2)}{p(C_1)} < 1$$

If $\lambda_1 = \lambda_2 = 0$ and $\lambda_{12} = \lambda_{21}$, this reduces to minimum misclassification scheme described earlier.

$(\lambda_{21} - \lambda_2) / (\lambda_{12} - \lambda_1)$ is like a threshold
Minimum risk classification

\[ p(x|c_a)p(c_a) = p(c_a|x) \]

- **instance of \( c_i \) labeled as \( c_a \)
- **instance of \( c_a \) labeled as \( c_i \)

If two types of errors are equally bad use criterion: choose class \( k \) if \( p(c_k|x) \geq p(c_j|x) \) \( \forall j \neq k \)

This will minimize the number of misclassifications.

What if errors have different costs?

Action \( a_j \): label instance as belonging to class \( j \)

Loss \( \lambda(c_i, a_j) \): cost associated with labeling an instance of class \( i \) as class \( j \)

Risk associated with action \( a_j \):

\[ R(a_j|x) = \sum \lambda(c_i, a_j) p(c_i|x) \]

Bayes decision rule: choose \( a_k \) if \( R(a_k|x) \leq R(a_j|x) \) \( \forall j \neq k \)
Density estimation

Classification problems can be reduced to class-conditional density estimation

- examine members of one class or the other
- estimate $p(x|c_i)$
- determine $p(c_i|x)$ from the $p(x|c_i)$

Parametric density estimation: form of density function is known; simply need to determine parameters of function (GMM)

E.g., Gaussian $\hat{p}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

Nonparametric density estimation: form of density function is unknown (Parzen windows, KNN)

Challenge: high dimensional data
Gaussian mixture model

Appropriate when data comes in several "clumps".

\[ p(x | \text{model}) = \bar{\pi}_x = \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right) \]

\[ = \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right) \]

- \( n \): number of Gaussians
- \( \pi_i \): weighting (mixture coefficient) for Gaussian \( i \)
- \( \mu_i \): mean of Gaussian \( i \)
- \( \sigma_i^2 \): variance of Gaussian \( i \)

How do we find model parameters given a set of examples \( \{ x \in \mathbb{R} \} \)?

Maximize likelihood that model generated the data:

\[ L = \prod_{x \in \{ x \}} \bar{\pi}_x \]

Equivalent to finding parameters that maximize

\[ \log L = \sum \log (\bar{\pi}_x) \]

Can do this with gradient ascent:

\[ \delta \pi_i = \frac{\partial \log L}{\partial \pi_i} \]
Nonparametric techniques

Histograms specify fixed size bins and count # instances in each bin

\[ \hat{p}(x) = \frac{N \cdot b(x)}{N} \]

Problems: loss of resolution (big bins), noisy (small bins) or curse of dimensionality

Equalized histograms Build variable sized bins, each containing

\[ \hat{p}(x) = \frac{N}{L \cdot b(x)} \]

Nice feature of histograms: can discard data once histogram has been constructed
Nonparametric techniques

Parzen windows

Given $x$, estimate $p(x)$

Assume fixed window around $x$ (hypercube)

Count number of data points in window ($k$)

$$\hat{p}(x) = \frac{k}{NL}$$

$N$: total # data pts

$L$: area (volume) of window

density of samples in window

Another way of thinking abt problem:

Each data point suggests a probability density function

height $= \frac{1}{L}$

width $= L$
Compute mean density suggested by all of the data

\[ \phi_j(x) = \begin{cases} 1 & \text{if } x \text{ is in window around data point } j \\ 0 & \text{otherwise} \end{cases}, \quad \| x - y_i \| < \frac{L}{2} \forall i \in d \]

where \( L = \frac{1}{N} \)

\[ \hat{\phi}(x) = \frac{1}{N} \sum_{i=1}^{N} \phi_j(x) \]

\[ = \frac{K}{NL} \]

Can use other window functions, e.g.,

\[ \phi_j(x) = \exp\left( -\frac{1}{2 \sigma^2} \| x - y_i \|^2 \right) \]

\[ L = \frac{1}{\sqrt{(2\pi)^d \sigma^d}} \]

As amount of data increases, make window smaller, e.g.,

\[ L \sim \frac{1}{\sqrt{N}} \]

\[ \sigma^2 \sim \frac{1}{N} \]
Example

Data drawn from a Rayleigh distribution:

Using Gaussian window function $w(x) = \frac{e^{-x^2}}{\sqrt{\pi}}$

Histogram for $N=256$

- $N=80$
  - $\lambda_0=4$

- $N=256$
  - $\lambda_0=4$
  - $\lambda_0=0.25$
  - $\lambda_0=8$
KNN (K-Nearest Neighbor)

With Parzen density estimate, we fix the window and count the number of samples in the window.

With KNN density estimate, we fix the number of samples and determine what volume window is required to encompass K samples.

\[ \hat{p}(x) = \frac{K-1}{N \cdot L(x)} \]

Variance of KNN density estimate > Variance of Parzen density estimate

(subtract 1 from K to obtain unbiased estimate)
Mixture of Gaussian density

3D data points

Histogram

Passen window

KNN
Clustering

Partition data to obtain compact description of data

K Means

![Diagram of data points and summary points]

Given N data points

Split data into K disjoint sets \( S_i \) so as to minimize

\[
J = \frac{1}{K} \sum_{i=1}^{K} \sum_{x_j \in S_i} |x_j - \mu_i|^2
\]

where \( \mu_i \) is the mean of data points in \( S_i \), given by

\[
\mu_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j
\]

Algorithm

Assign points at random to K sets

Loop

Compute mean \( \mu_i \) for each set \( i \)

Repeat

Reassign each point \( j \) according to which mean is nearest
- Algorithm guaranteed to improve clustering fit (reduce J) on each iteration.
- Algorithm eventually converges
- Related to Gaussian mixture model in which all Gaussians have equal variance
- Can be used to discard "unimportant" (?) variability in data
- How to pick K? Some clustering algorithms do this automatically

Hierarchical clustering

Algorithm

Loop
  Find two data points that are closest to one another
  Merge points

Repeat
Hierarchical clustering of hidden unit representations of a neural network performing a language task.