Statistical Approaches to Classification and Regression

Professor Michael C. Mozer

CSCI 3202

Correlation

Measure of relation between 2 or more variables

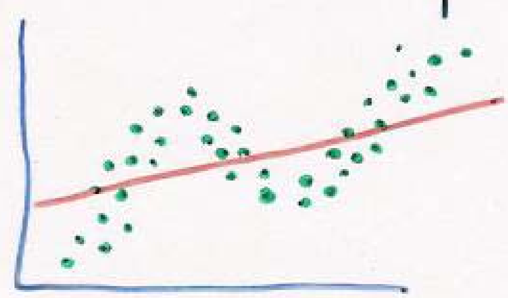
Pearson ρ: linear relation

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

strength of relationship determined by ρ^2 – proportion of variance accounted for

Artifacts (see figure)

Nonlinear relationships



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

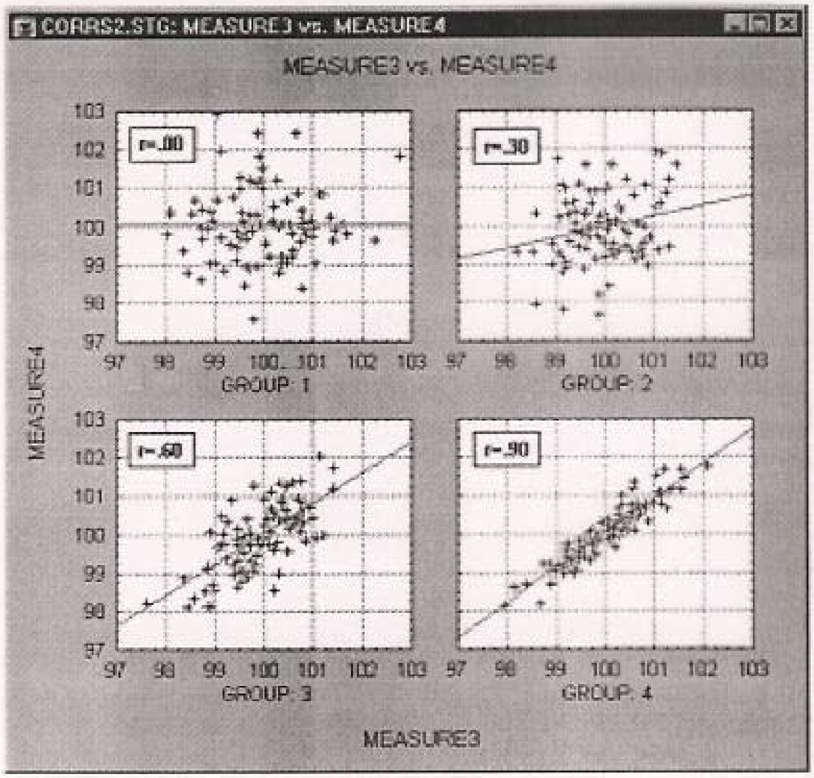
Importance of graphing data

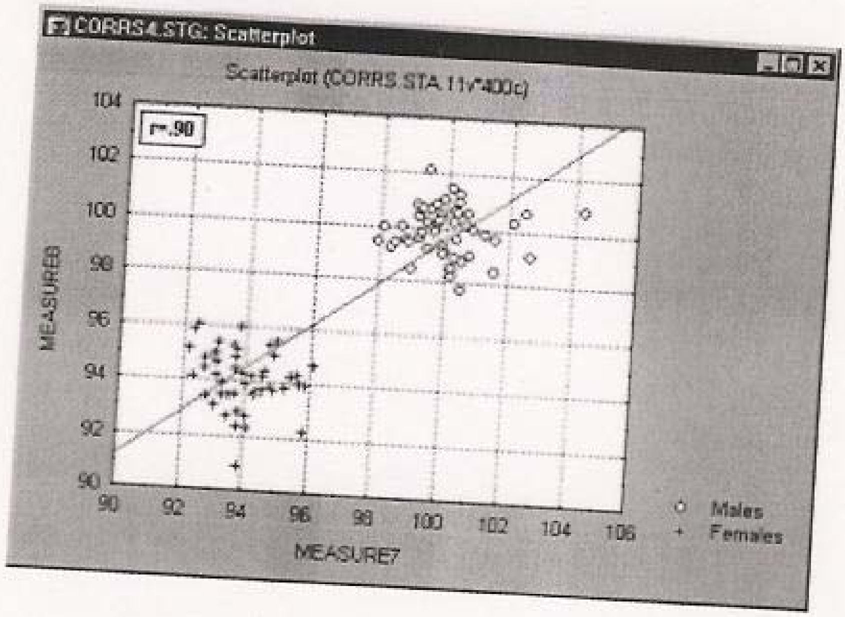
covariance of x & y

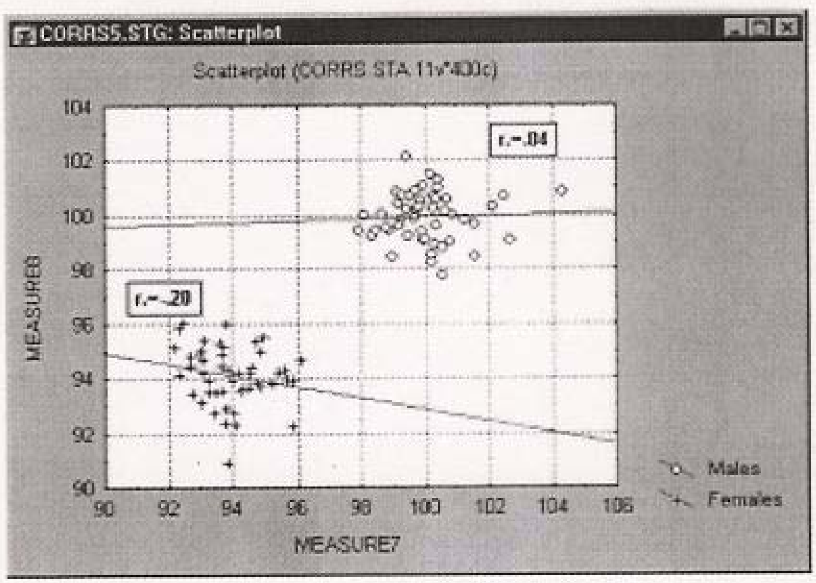
$$\rho = \frac{\sum_i x_i y_i - \frac{1}{n}\sum_i x_i \sum_i y_i}{\sqrt{\sum x_i^2 - (\sum x_i)^2/n}\;\sqrt{\sum y_i^2 - (\sum y_i)^2/n}}$$

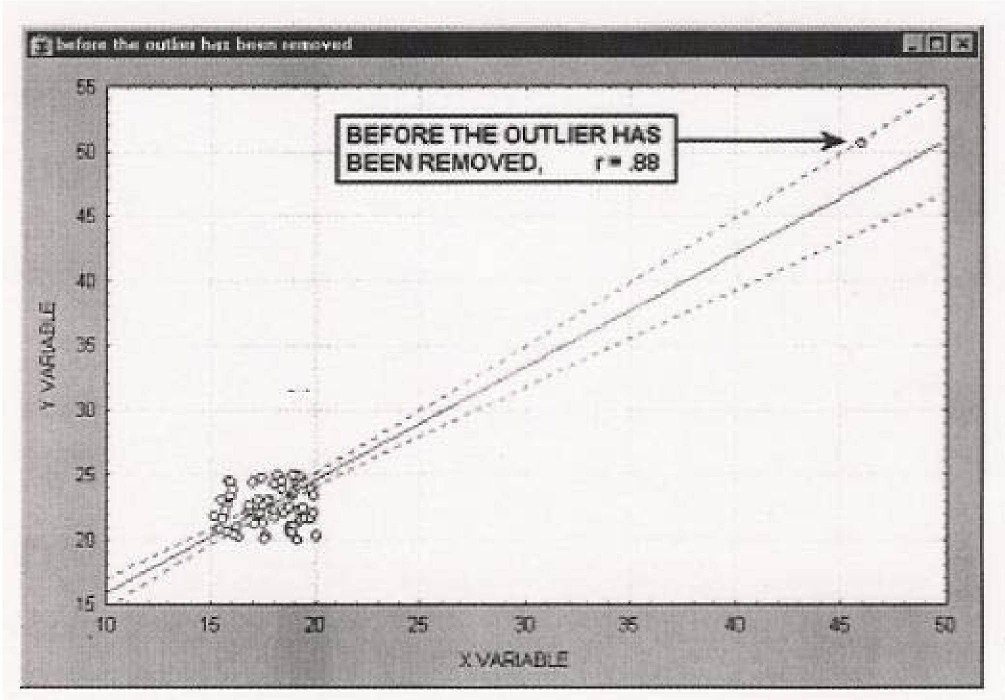
standard dev. of x

standard dev. of y




CORRS4.STG: Scatterplot
Scatterplot (CORRS.STA 11v*400c)
r=.90
MEASURE8
MEASURE7
Males
Females


CORRS5.STG: Scatterplot
Scatterplot (CORRS.STA 11v*400c)
r.=.04
r.=-.20
MEASURE8
MEASURE7
Males
Females



before the outlier has been removed
BEFORE THE OUTLIER HAS
BEEN REMOVED, r = .88
Y VARIABLE
X VARIABLE
55
50
45
40
35
30
25
20
15
10
15
20
25
30
35
40
45
50

Linear regression

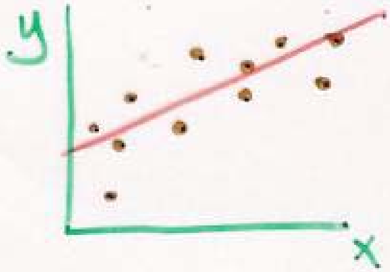
Use one variable to predict the value of another

$$\hat{y} = a + bx$$

dependent variable

regression coefficients

independent variable



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

Correlation coefficient: how small are residuals?

$\rho = (1 - \text{residual variability} / \text{original variance})^{1/2}$

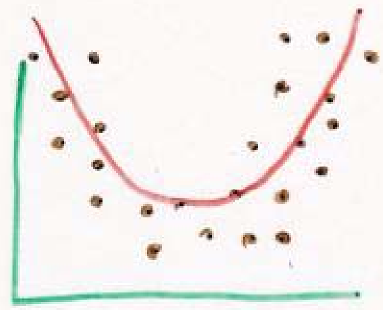
multiple linear regression

$$\hat{y} = a + b_1 x_1 + b_2 x_2 + b_3 x_3 + \dots$$

Parametric nonlinear regression

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

E.g., polynomial regression



$$y = f(x, x^2)$$

$$\Rightarrow \quad x_1 = x \qquad x_2 = x^2$$

$$\hat{y} = a + b_1 x_1 + b_2 x_2$$

E.g., exponential regression

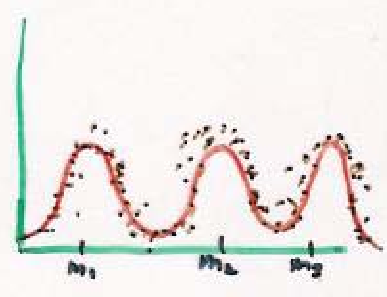
(b negative)

$$y = \exp(a + bx)$$

$$\Rightarrow \quad y' = \log y$$

$$\hat{y}' = a + bx$$

E.g., basis function regression



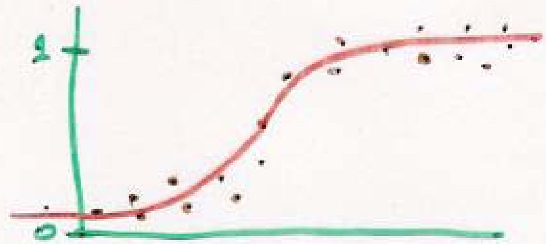
$$x_1 = f_1(x), \quad x_2 = f_2(x), \quad x_3 = f_3(x)$$

$$\hat{y} = a + b_1 x_1 + b_2 x_2 + b_3 x_3$$

$$y_i(x) = e^{-(x-m_i)^2}$$

Parametric nonlinear regression (cont.)

e.g., Logit regression   dependent variable in range 0–1 (e.g., probabilities)

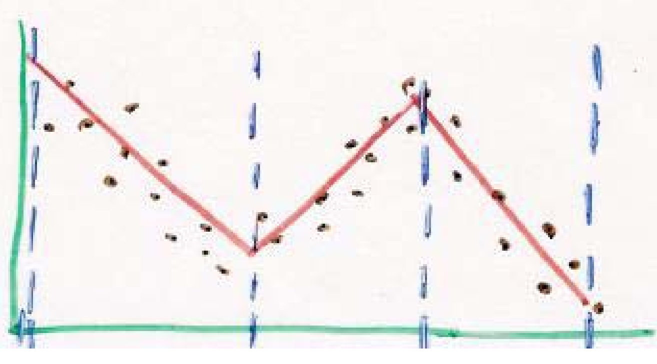


$$y' = \log\left(\frac{y}{1-y}\right)$$

$$\hat{y}' = a + bx \qquad \equiv \qquad \hat{y} = \frac{1}{1+e^{-(a+bx)}}$$

Same as neural net with no hidden units

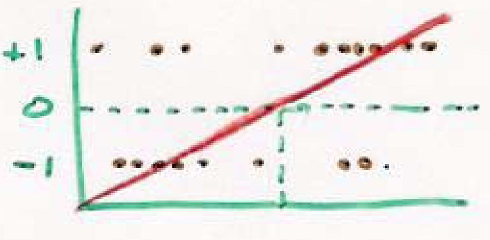
e.g. piecewise linear models



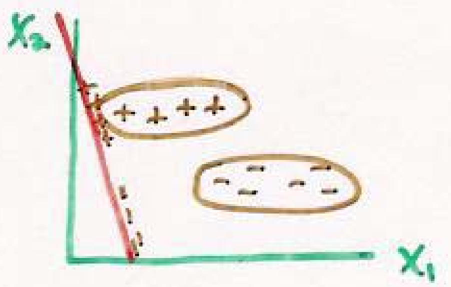
Classification

Treat as a regression problem

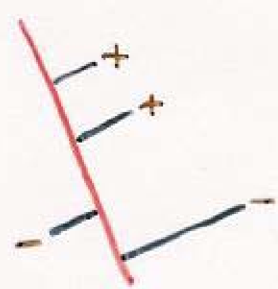
$\hat{y} = a + b_1 x_1 + b_2 x_2 + \ldots$ where $y = +1$ for class 1
-1 for class 2



Fischer linear discriminant analysis

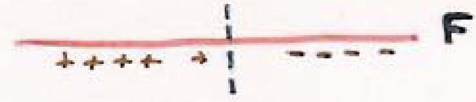


projection
$b^T x$



project data on to a line that minimizes within-class variability and maximizes between-class variability

$$F = b_1 x_1 + b_2 x_2$$



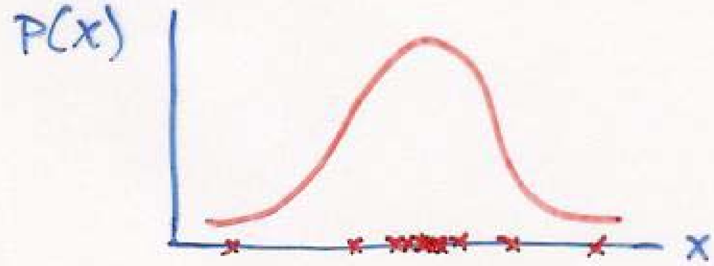
Fisher is same as linear regression if we use these targets for the regression:

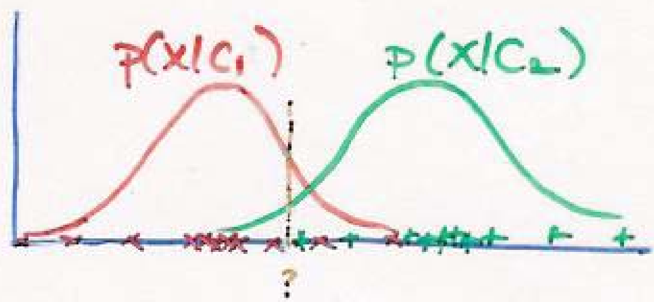
$$y = \frac{n_2 + n_1}{n_1} \text{ for class 1}$$

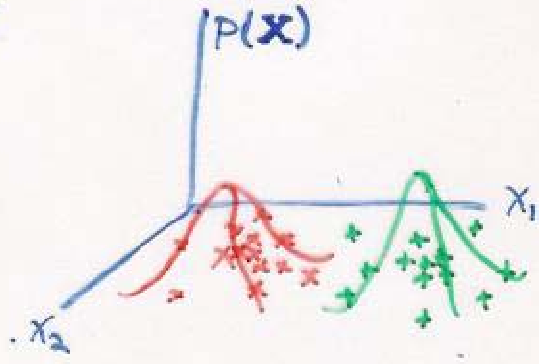
$$= -\frac{n_2 + n_1}{n_2} \text{ for class 2}$$

Gaussian Noise Distribution

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



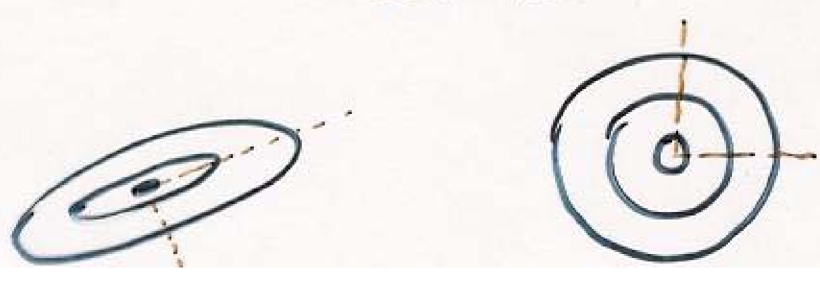




$$p(\mathbf{x} \mid c_i) = \frac{1}{(2\pi\sigma_i^2)^{d/2}} \exp\left[- \frac{|\mathbf{x} - \bar{\mathbf{x}}_i|^2}{2\sigma_i^2} \right]$$ ← simple case – uniform spread

general case

$$= \frac{1}{(2\pi)^{d/2} \underbrace{|\Sigma_i|}_{\text{determinant}}{}^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}}_i)^T \underbrace{\Sigma_i^{-1}}_{\text{covariance matrix}} (\mathbf{x} - \bar{\mathbf{x}}_i) \right]$$



Bayes classifier

Choose class 1 if $p(C_1|x) > p(C_2|x)$

Equivalently, $$\frac{p(C_1|x)}{p(C_2|x)} > 1$$

$$\frac{p(C_1|x)}{p(C_2|x)} = \frac{[\,p(x|C_1)\,p(C_1) / \cancel{p(x)}\,]}{[\,p(x|C_2)\,p(C_2) / \cancel{p(x)}\,]}$$

$$= \underbrace{\frac{p(x|C_1)}{p(x|C_2)}}_{\text{likelihood ratio}} \; \underbrace{\frac{p(C_1)}{p(C_2)}}_{\text{prior ratio}} \overset{?}{>} 1$$

$p(x|C_1)$ and $p(x|C_2)$ computed assuming Gaussian distribution, w/ mean & covariance estimated from data

$p(C_1)$ and $p(C_2)$ estimated from data

Recognition becomes the problem of <u>density estimation</u> — $p(x|C_i)$

Two category example.

a_1: decision to label x instance of class 1
a_2: decision to label x instance of class 2

$$R(a_1|x) = \lambda_{11}\, p(c_1|x) + \lambda_{21}\, p(c_2|x)$$
$$R(a_2|x) = \lambda_{12}\, p(c_1|x) + \lambda_{22}\, p(c_2|x)$$

Bayes decision rule: Choose a_1 if

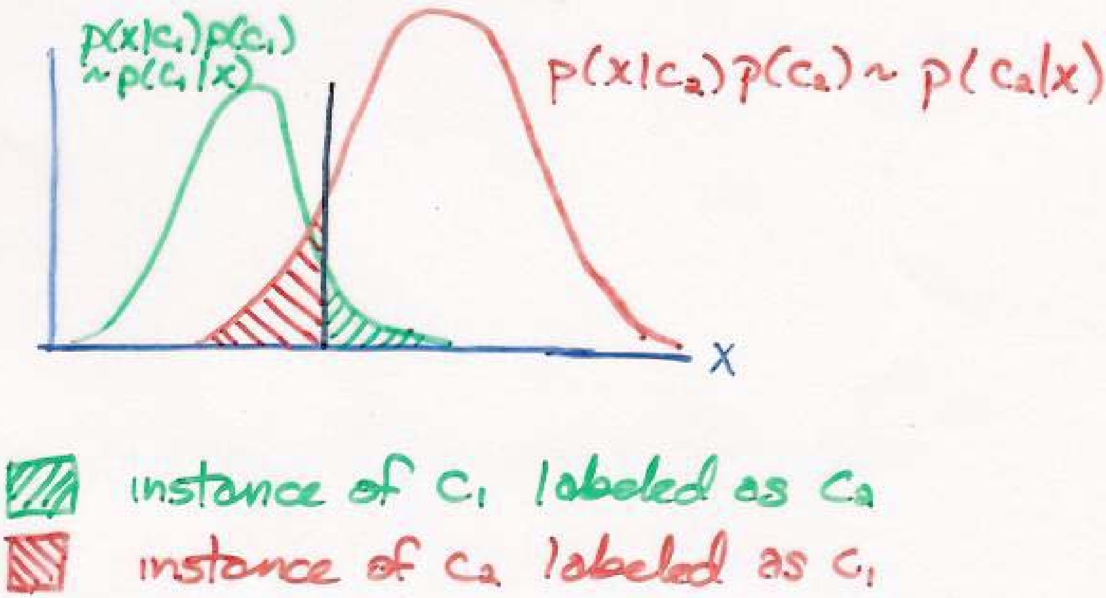
$$R(a_1|x) < R(a_2|x)$$
$$\lambda_{11}\, p(c_1|x) + \lambda_{21}\, p(c_2|x) < \lambda_{12}\, p(c_1|x) + \lambda_{22}\, p(c_2|x)$$
$$(\lambda_{21} - \lambda_{22})\, p(c_2|x) < (\lambda_{12} - \lambda_{11})\, p(c_1|x)$$
$$\frac{(\lambda_{21} - \lambda_{22})}{(\lambda_{12} - \lambda_{11})} \frac{p(c_2|x)}{p(c_1|x)} < 1$$

$$\frac{(\lambda_{21} - \lambda_{22})}{(\lambda_{12} - \lambda_{11})} \frac{p(x|c_2)}{p(x|c_1)} \frac{p(c_2)}{p(c_1)} < 1$$

If $\lambda_{11} = \lambda_{22} = 0$ and $\lambda_{12} = \lambda_{21}$, this reduces to minimum misclassification scheme described earlier.

$(\lambda_{21} - \lambda_{22})/(\lambda_{12} - \lambda_{11})$ is like a threshold

Minimum risk classification



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

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

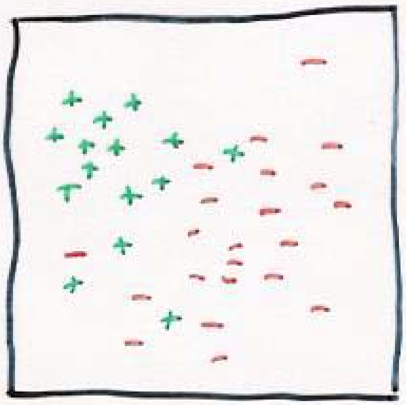
Risk associated with action a_j:

$$R(a_j|x) = \sum_i \lambda(c_i, a_j)\, p(c_i|x)$$

Bayes decision rule: choose a_k if $R(a_k|x) \leq R(a_j|x) \quad \forall_{j \neq k}$

Density estimation

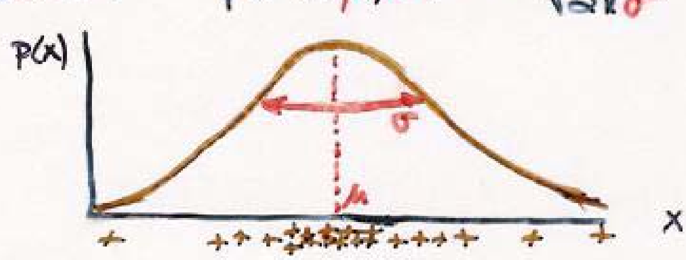
Classification problems can be reduced to class-conditional density estimation



- examine members of one class or the other
- estimate $p(x \mid c_i)$ — x: data, c_i: class
- determine $p(c_i \mid x)$ from the $p(x \mid c_j)$

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

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

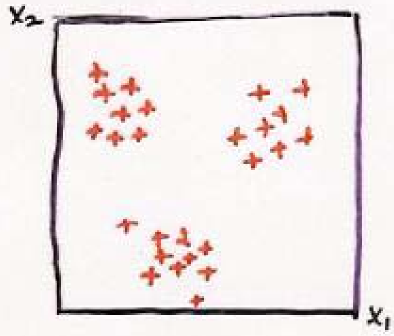


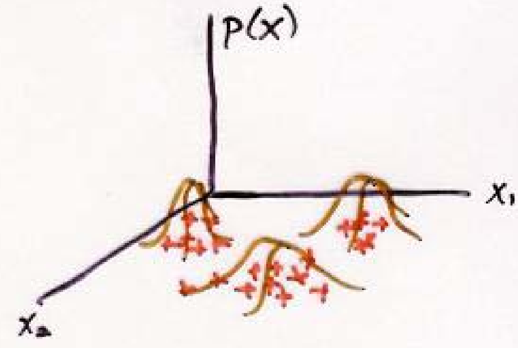
nonparametric density estimation: form of density function is unknown (Parzen windows, KNN)

Challenge: high dimensional data

Gaussian mixture model

Appropriate when data comes in several "clumps"





$$p(x\,|\,\text{model}) = \gamma_x = \sum_{i=1}^{n} s_i\, p(x\,|\,\text{Gaussian}_i)$$

$$= \sum_{i=1}^{n} s_i \frac{1}{\sqrt{2\pi\sigma_i^2}^{\,d}} \exp\left(-\frac{|x-\mu_i|^2}{2\sigma_i^2}\right)$$

n: # Gaussians

$\{s_i\}$: 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 $\{x_e\}$?

Maximize likelihood that model generated the data:

$$L = \prod_{e=1}^{E} \gamma_{x_e}$$

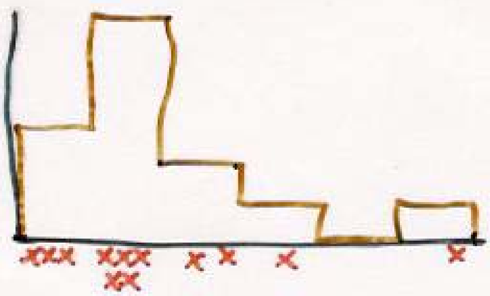
Equivalent to finding parameters that maximize

$$\log L = \sum \log(\gamma_{x_e})$$

Can do this with gradient ascent: $\Delta s_i \sim \frac{\partial \log L}{\partial s_i}$ etc.

Nonparametric techniques

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



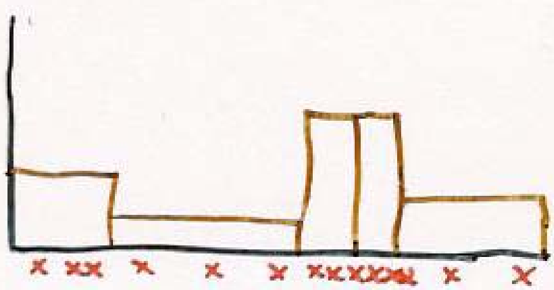
$$\hat{p}(x) = \frac{n_{b(x)}}{N}$$

data points in bin of x

data points

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

Equalized histograms Build variable sized bins, each containing



$$\hat{p}(x) = \frac{n}{L_{b(x)}}$$

data points per bin

size of bin for x

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

Nonparametric techniques.

Parzen windows

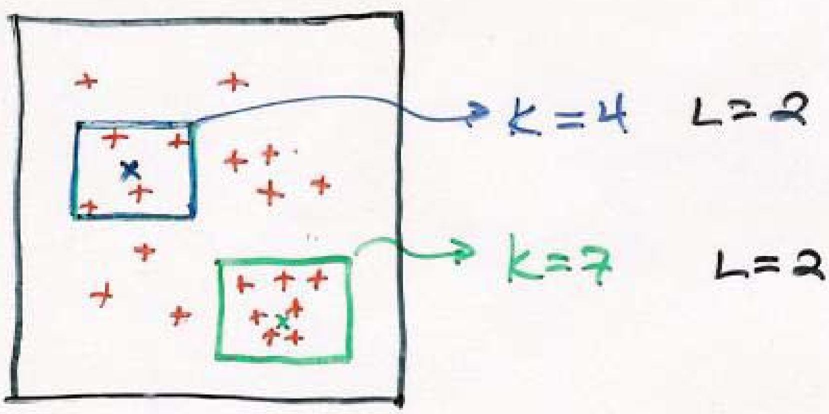
Given x, estimate $p(x)$

Assume fixed <u>window</u> 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

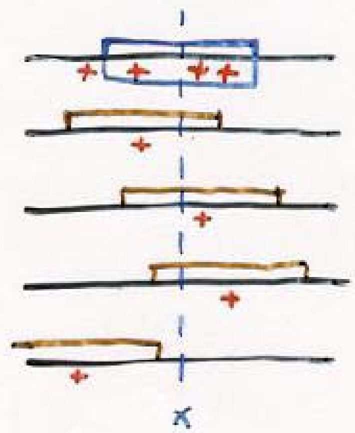
density of samples in window



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

$$\phi_j(x) = \begin{cases} 1 & \text{if } x \text{ is in window around data point}_j \\ 0 & \text{otherwise.} \end{cases}$$

$|x_i - y_{ji}| < \frac{l}{2} \ \forall i \in d$

where $L = l^d$

$$\hat{p}(x) = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{L} \phi_j(x)$$

$$= \frac{\Sigma \phi_j(x)}{NL} = \frac{K}{NL}$$

Can use other window functions, e.g.,

$$\phi_j(x) = \exp\left(-\frac{|x - y_j|^2}{2\sigma^2}\right)$$

$$L = \frac{1}{(2\pi\sigma^2)^{d/2}}$$

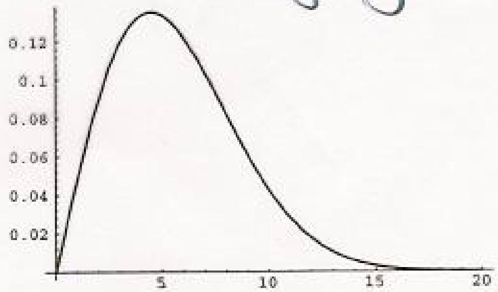
As amount of data increases, make window smaller, e.g.,

$$l \sim \frac{1}{\sqrt{N}}$$

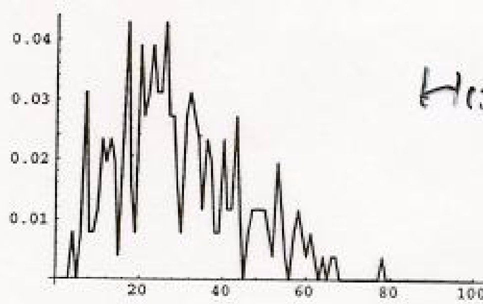
$$\sigma^2 \sim \frac{1}{\sqrt{N}}$$

Example

Data drawn from a Rayleigh distribution:

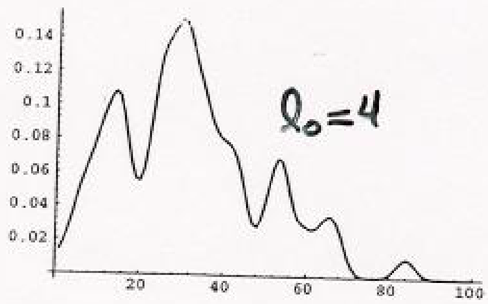


Using Gaussian window function w/ $\sigma^2 = \frac{l_0}{\sqrt{N}}$

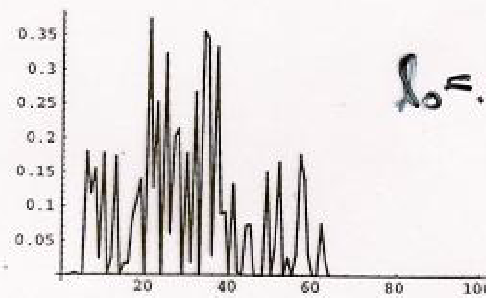


Histogram for $N=256$

$N=80$

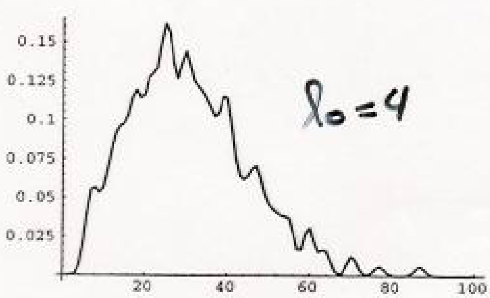


$l_0=4$

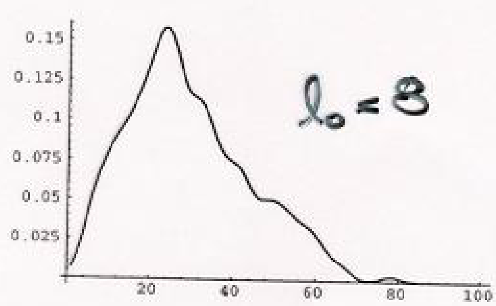


$l_0=.25$

$N=256$



$l_0=4$

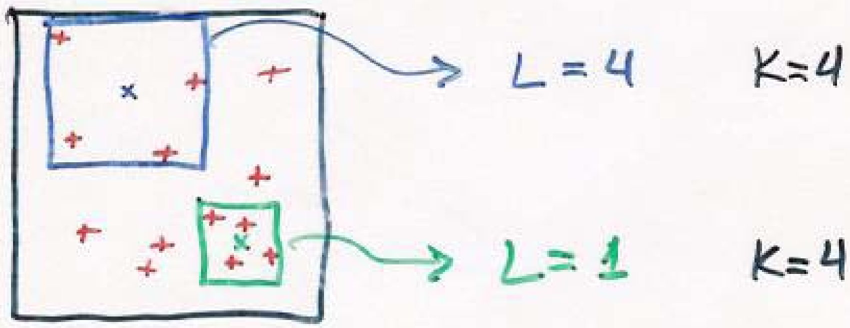


$l_0=8$

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 (K) and determine what volume window is required to encompass K samples.



$$\hat{p}(x) = \frac{K-1}{N\,L(x)}$$

(subtract 1 from K to obtain unbiased estimator)

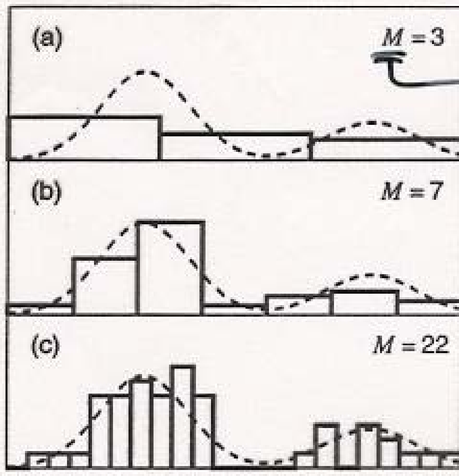
Variance of KNN density estimate > Variance of Parzen density estimate

Mixture of Gaussians density $\mu_1 = .5$ $\mu_2 = .8$ $\sigma_1 = \sigma_2 = .1$

$s_1 = .7$ $s_2 = .3$

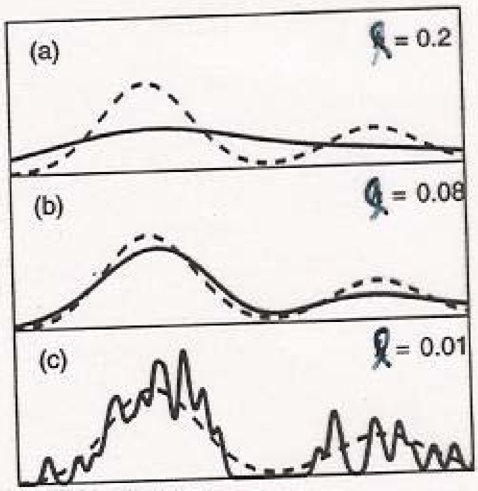
30 data points

Histogram

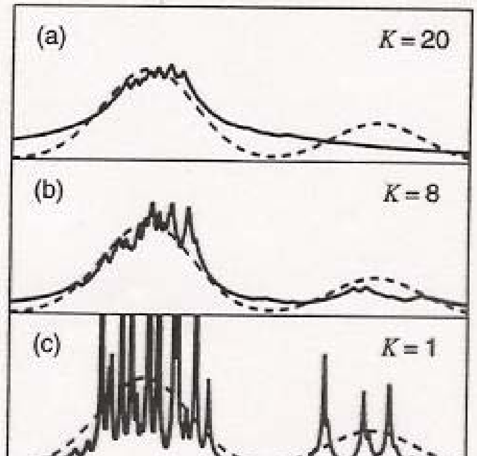


bins

Parzen window



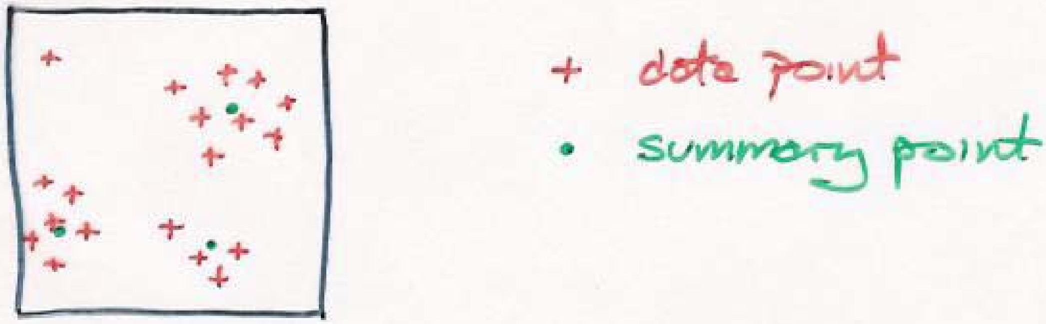
KNN



Clustering

Partition data to obtain compact description of data

K means



Given N data points
Split data into K disjoint sets S_i so as to minimize

$$J = \sum_{i=1}^{k} \sum_{j \in S_i} | x_j - \mu_i |^2$$

where μ_i is the mean of data points in S_i, given by

$$\mu_i = \frac{1}{|S_i|} \sum_{j \in S_i} x_j$$

Algorithm

Assign points at random to K sets

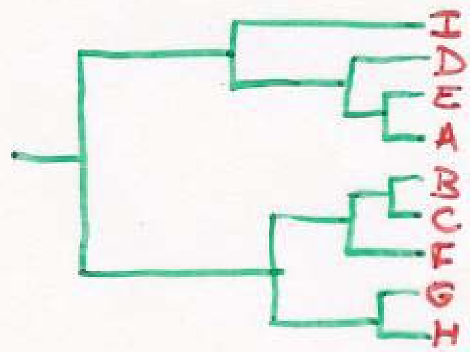
Loop
  Compute mean μ_i for each set i
  Reassign each point j according to which mean is nearest
Repeat

- 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