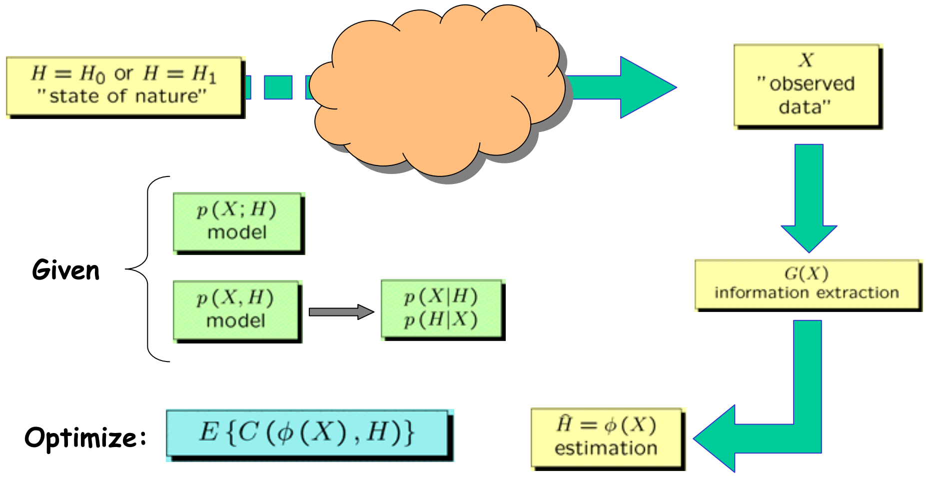
HST.582J / 6.555J / 16.456J Biomedical Signal and Image Processing
Spring 2007

Classification



Binary Hypothesis Testing (Neyman-Pearson)
(and a "simplification" of the notation)

- 2-Class problems are equivalent to the binary hypothesis testing problem.

$$
\begin{aligned}
H_1 &: x \sim p_{X|H_1}(x|H_1 \text{ is true}) \\
H_0 &: x \sim p_{X|H_0}(x|H_0 \text{ is true})
\end{aligned}
$$

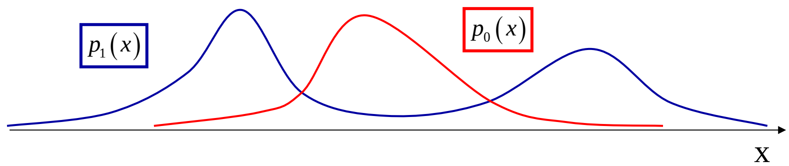
The goal is *estimate* which Hypothesis is true (i.e. from which class our sample came from).

- A minor change in notation will make the following discussion a little simpler.

$$
\begin{aligned}
p_1(x) &= p_{X|H_1}(x|H_1 \text{ is true}) \\
p_0(x) &= p_{X|H_0}(x|H_0 \text{ is true})
\end{aligned}
$$

Probability density models for the measurement x depending on which hypothesis is in effect.

Decision Rules

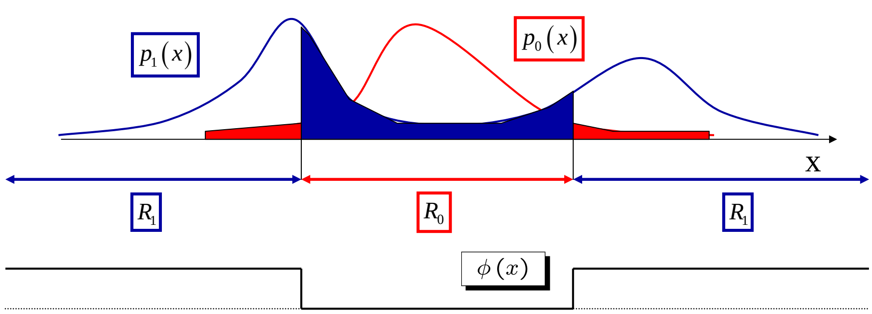


- Decision rules are functions which map measurements to choices.
- In the binary case we can write it as

$$\phi(x) = \begin{cases} 1 & ; \; x \in R_1 \\ 0 & ; \; x \in R_0 \end{cases}$$

where we need to designate R_0 and R_1.

Error Types



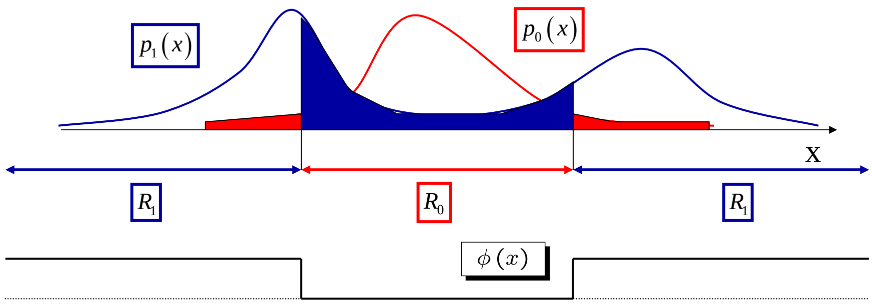
- There are 2 types of errors
- A "miss"

E_M: X falling in R_0 **AND** H_1 being correct

- A "false alarm"

E_F: X falling in R_1 **AND** H_0 being correct

Event Types



$$\begin{aligned} P_D &= E_{H_1}\{\phi(x)\} \\ &= \int p_1(x)\phi(x)\,dx \\ &= \beta \end{aligned}$$

$$\begin{aligned} P_M &= E_{H_1}\{1-\phi(x)\} \\ &= \int p_1(x)(1-\phi(x))\,dx \\ &= 1-\beta \end{aligned}$$

$$\begin{aligned} P_F &= E_{H_0}\{\phi(x)\} \\ &= \int p_0(x)\phi(x)\,dx \\ &= \alpha \end{aligned}$$

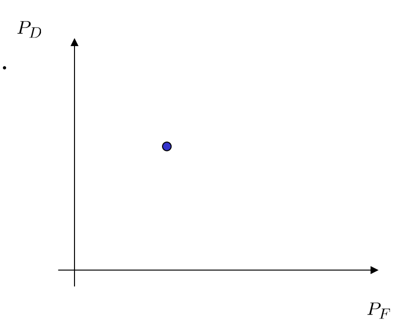
$$\begin{aligned} 1-P_F &= E_{H_0}\{1-\phi(x)\} \\ &= \int p_0(x)(1-\phi(x))\,dx \\ &= 1-\alpha \end{aligned}$$

Plotting β versus α

- For some decision rule we can plot the probability of detection versus the probability of a false alarm.

$$\begin{aligned} P_D &= E_{H_1}\{\phi(x)\} \\ &= \int p_1(x)\phi(x)\,dx \\ &= \beta \end{aligned}$$

$$\begin{aligned} P_F &= E_{H_0}\{\phi(x)\} \\ &= \int p_0(x)\phi(x)\,dx \\ &= \alpha \end{aligned}$$

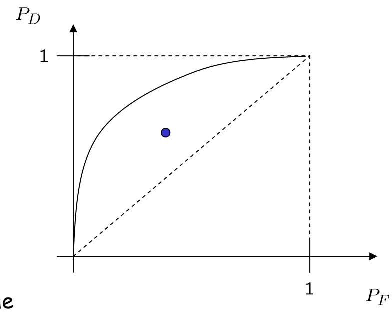


Receiver Operating Characteristic (ROC) Curve

- The form of the optimal decision function took the form of a likelihood ratio.

$$\phi(x) = \begin{cases} 1 & ; \; p_1(x) \geq \gamma p_0(x) \\ 0 & ; \; \text{otherwise} \end{cases}$$

$$\phi(x) = \begin{cases} 1 & ; \; \frac{p_1(x)}{p_0(x)} \geq \gamma \\ 0 & ; \; \text{otherwise} \end{cases}$$



- This test is optimal in the sense that for any setting of γ with a resulting P_D and P_F
  - *any* other decision rule with the same P_D (or β) has a P_F (or α) which is higher.
  - *any* other decision rule with the same P_F (or α) has a P_D (or β) which is lower.

Binary Hypothesis Testing (Bayesian)

- 2-Class problems are equivalent to the binary hypothesis testing problem.

$$\begin{aligned} H_1 &: x \sim p_{X|H_1}(x|H_1 \text{ is true}) \\ H_0 &: x \sim p_{X|H_0}(x|H_0 \text{ is true}) \end{aligned}$$

The goal is *estimate* which Hypothesis is true (i.e. from which class our sample came from).

- A minor change in notation will make the following discussion a little simpler.

$$\begin{aligned} P_1 &= \Pr(H = H_1) \\ P_0 &= \Pr(H = H_0) \end{aligned}$$

Prior probabilities of each class

$$\begin{aligned} p_1(x) &= p_{X|H_1}(x|H_1 \text{ is true}) \\ p_0(x) &= p_{X|H_0}(x|H_0 \text{ is true}) \end{aligned}$$

Class-conditional probability density models for the measurement x

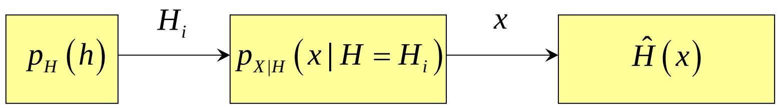
Marginal density of X

$$p_x(x) = P_1 p_1(x) + P_0 p_0(x)$$

Conditional probability of the hypothesis H_i given X

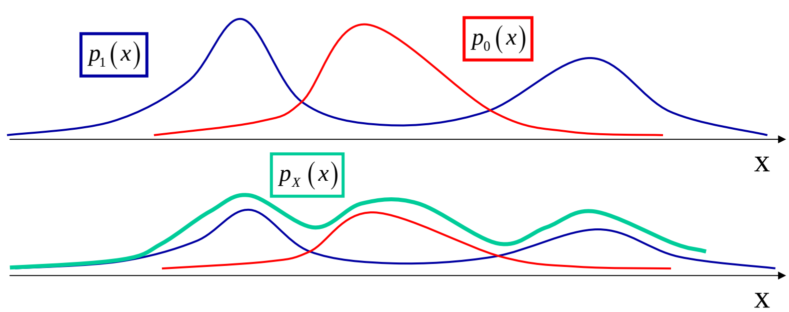
$$\begin{aligned} P_{H_i|x}(H_i|x) &= \frac{P_i p_i(x)}{p_x(x)} \\ &= \frac{P_i p_i(x)}{P_1 p_1(x) + P_0 p_0(x)} \end{aligned}$$

The Generative Model



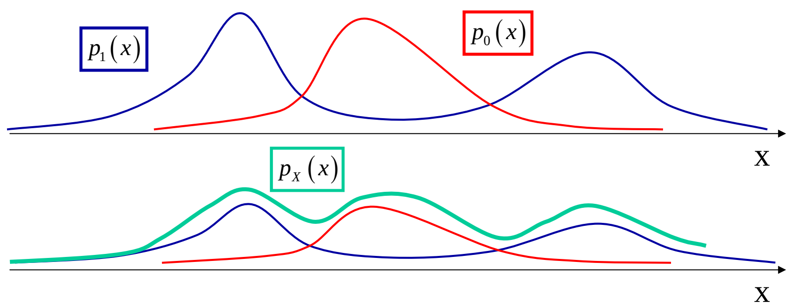
- A random process generates values of H_k, which are sampled from the probability mass function $p_H(H)$.
- We would like to ascertain the value of H_k, **unfortunately** we don't observe it directly.
- **Fortunately**, we observe a ***related*** random variable, x.
- From x, we can compute the **best** estimate of H_k.
- What is the nature of the ***relationship***, and what do we mean by ***best***.

A Notional 1-Dimensional Classification Example



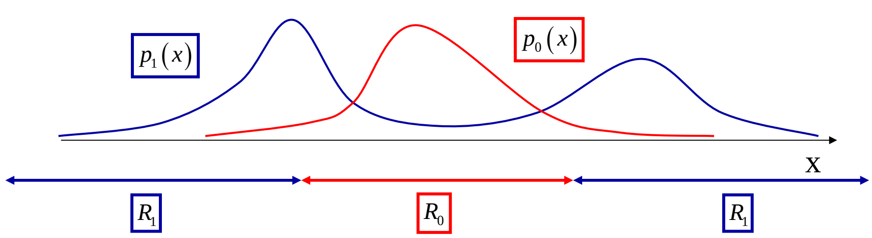
- So given observations of x, how should select our best guess of H_i?
- Specifically, what is a good criterion for making that assignment?
- Which H_i should we select before we observe x.

Bayes Classifier



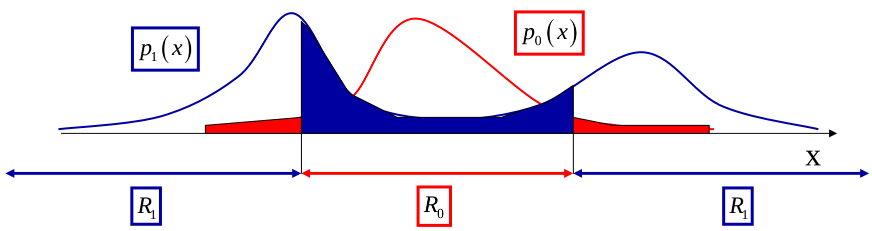
- A reasonable criterion for guessing values of H given observations of X is to minimize the probability of error.
- The classifier which achieves this minimization is the Bayes classifier.

Probability of Misclassification



- Before we derive the Bayes' classifier, consider the probability of misclassification for an **arbitrary** classifier (i.e. decision rule).
  - The first step is to assign regions of X, to each class.
  - An error occurs if a sample of x falls in R_i and we assume hypothesis H_j.

Probability of Misclassification



- An error is comprised of two events

E_1: X falling in R_0 **AND** H_1 being correct

E_0: X falling in R_1 **AND** H_0 being correct

- These are *mutually exclusive* events so their joint probability is the sum of their individual probabilities

$$\begin{aligned} P_E &= \Pr\{E_1\} + \Pr\{E_0\} \\ &= P_1 \Pr\{X \in R_0 | H_1\} + P_0 \Pr\{X \in R_1 | H_0\} \\ &= P_1 \int_{R_0} p_1(x)\, dx + P_0 \int_{R_1} p_0(x)\, dx \end{aligned}$$

fMRI example

- Noisy measurements
- Conditional predicted observations
- Quantifiable costs
- Tumor/Gray-White Matter Separation
- Eloquent/Non-Eloquent Cortex Discrimination

Risk Adjusted Classifiers

Suppose that making one type of error is more of a concern than making another. For example, it is worse to declare H_1 when H_2 is true then vice versa.

- This is captured by the notion of "cost".

$$C_{ij} = \text{cost of declaring } H_i \text{ when } H_j \text{ is correct}$$

- In the binary case this leads to a cost matrix.

$$\text{declared hypothesis} \left\{ \underbrace{\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix}}_{\text{correct hypothesis}} \right.$$

- The Risk Adjusted Classifier tries to minimize the expected "cost"

Derivation

- We'll simplify by assuming that $C_{11}=C_{22}=0$ (there is zero cost to being correct) and that all other costs are positive.
- Think of cost as a piecewise constant function of X.
- If we divide X into decision regions we can compute the expected cost as the cost of being wrong times the probability of a sample falling into that region.

$$\begin{aligned} E\{C(x,H)\} &= \int_{R_0} C_{01}P_1 p_1(x)\,dx + \int_{R_1} C_{10}P_0 p_0(x)\,dx \\ &= C_{01}P_1\left(1 - \int_{R_1} p_1(x)\,dx\right) + C_{10}P_0 \int_{R_1} p_0(x)\,dx \\ &= C_{01}P_1 + \int_{R_1}\left(\underbrace{C_{10}P_0 p_0(x)}_{\geq 0} - \underbrace{C_{01}P_1 p_1(x)}_{\geq 0}\right) dx \end{aligned}$$

Risk Adjusted Classifiers

Expected Cost is then

$$E\{C(x,H)\} = C_{01}P_1 + \int_{R_1}\left(\underbrace{C_{10}P_0 p_0(x)}_{\geq 0} - \underbrace{C_{01}P_1 p_1(x)}_{\geq 0}\right) dx$$

- As in the minimum probability of error classifier, we note that all terms are positive in the integral, so to minimize expected "cost" choose R_1 to be:

$$R_1 = \{x : C_{01}P_1 p_1(x) > C_{10}P_0 p_0(x)\}$$

- Alternatively

$$R_1 = \left\{x : \frac{p_1(x)}{p_0(x)} > \frac{C_{10}P_0}{C_{01}P_1}\right\}$$

- If $C_{10}=C_{01}$ then the risk adjusted classifier is equivalent to the minimum probability of error classifier.
- Another interpretation of "costs" is an adjustment to the prior probabilities.

$$\frac{P_0^{\text{adj}}}{P_1^{\text{adj}}} = \frac{C_{10}P_0}{C_{01}P_1}$$

- Then the risk adjusted classifier is equivalent to the minimum probability of error classifier with prior probabilities equal to P_1^{adj} and P_0^{adj}, respectively.

Okay, so what.

All of this is great. We now know what to do in a few classic cases if some nice person hands us all of the probability models.

- In general we aren't given the models – What do we do?

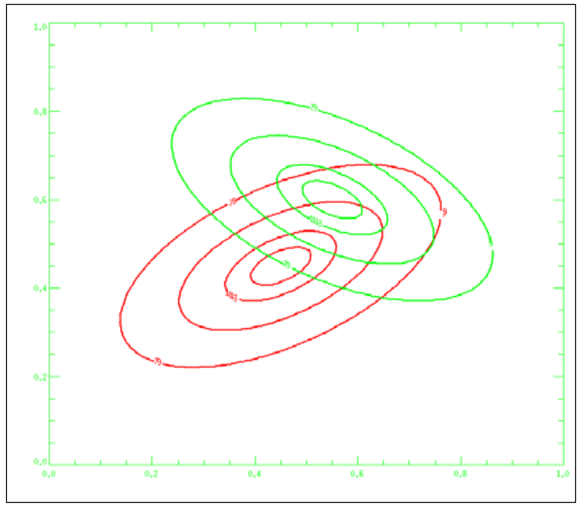
Density estimation to the rescue.

- While we may not have the models, often we do have a collection of labeled measurements, that is a set of $\{x,H_j\}$.
- From these we can estimate the class-conditional densities. Important issues will be:
  - How "close" will the estimate be to the true model.
  - How does "closeness" impact on classification performance?
  - What types of estimators are appropriate (parametric vs. nonparametric).
  - Can we avoid density estimation and go straight to estimating the decision rule directly? (generative approaches versus discriminative approaches)

Density Estimation

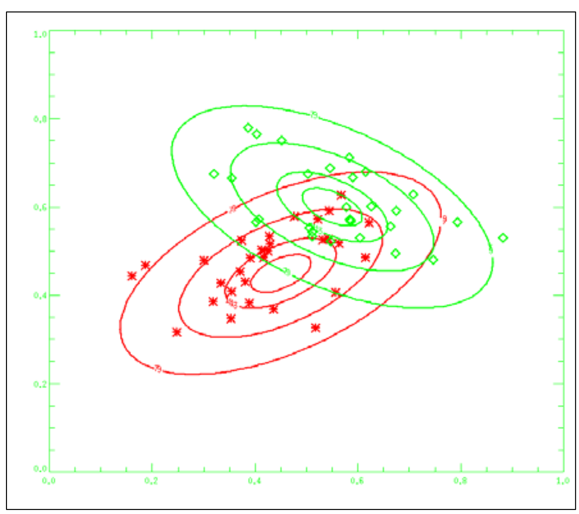
The Basic Issue

- All of the theory and methodology has been developed as if the model were handed to us.
- In practice, that is not what happens.



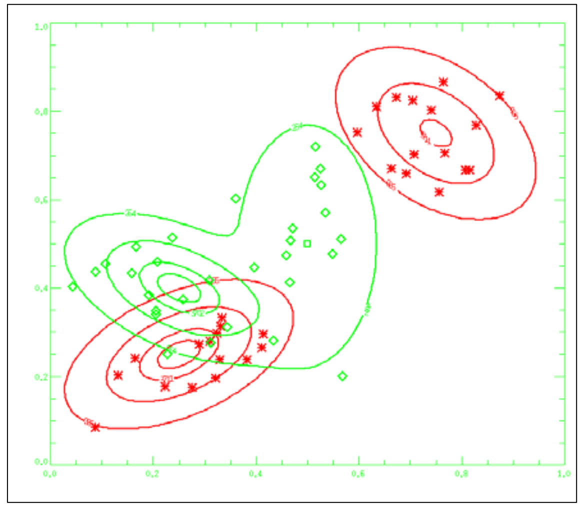
The Basic Issue

- All of the theory and methodology has been developed as if the model were handed to us.
- In practice, that is not what happens.

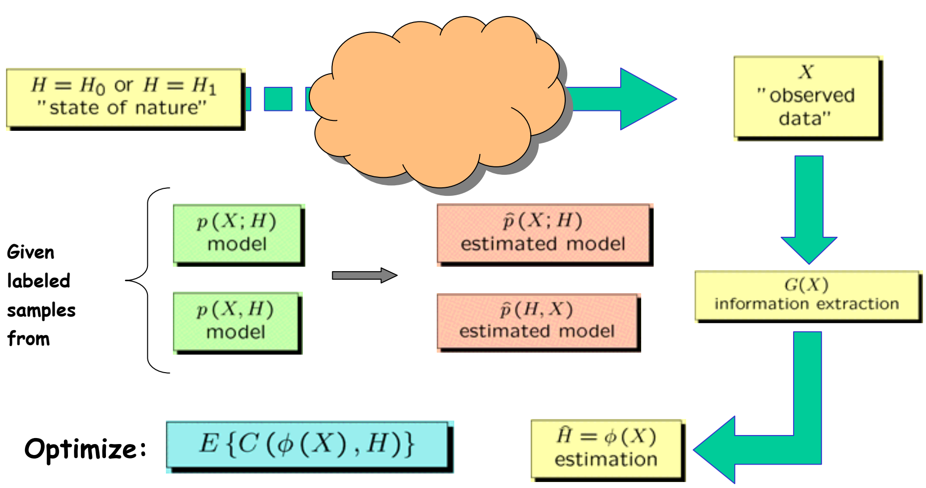


Even more challenging

- The model may not follow some convenient form.



Classification with Model Estimation



Density/Parameter Estimation

- We need to infer a density (or do we?) from a set of labeled samples.
- There are essentially 2 styles of density estimation
  - Parametric
  - Nonparametric

Primary Estimation Concepts

- While theoretical optimality of classifiers assumes known generative models, as a practical matter we rarely (if ever) know the true source density (or even its form).
- Methods by which we infer the class-conditional densities from a finite set of **labeled** samples.
- The sense in which a density estimate is "good".
- The difference between estimating a density and a "decision rule" for classification.

Parametric Estimation

- Assume the model has known functional form
- Estimate the parameters of the function from samples

Experiment (example)

- After tossing a coin 100 times you observe 56 heads and 44 tails.
- What probability model best explains our observations?
  - We'll need to define "best".
  - We might want to consider our prior experience/expectations.

Some Terms

- ***X*** is a set of *N* ***independent*** samples of an *M*-dimensional random variable.
  $$X = \{x_1, \ldots, x_N\} \qquad x_i \in \Re^M$$
- When appropriate we'll define a *P*-dimensional parameter vector.
  $$\theta \in \Re^P$$
- Denotes that samples are drawn from the probability density or mass function parameterized by θ.
  $$x_i \sim p(x;\theta)$$
- Denotes the "true" density from which samples are drawn.
  $$x_i \sim p_x(x)$$

Some Terms (con't)

Example:

- X are samples of an M-dimensional Gaussian random variable
  $$X = \{x_1, \ldots, x_N\} \qquad x_i \in \Re^M$$
- The set θ contains the mean vector and covariance matrix which completely specify the Gaussian density.
  $$\theta = \left\{\mu_x \in \Re^M, \Sigma_x \in \Re^{M \times M}\right\}$$
- P is the number of *independent* parameters which consists of M (for the mean vector) plus M (for the diagonal elements of the covariance matrix) plus (M^2-M)/2 (which is half of the off-diagonal elements – the other half are the same).
  $$\begin{aligned} P &= M + M + \frac{M^2 - M}{2} \\ &= \frac{3}{2}M + \frac{M^2}{2} \end{aligned}$$
- The parameterized (model) density is then the Gaussian form with mean and covariance as parameters.
  $$p(x;\theta) = \frac{1}{(2\pi)^{\frac{M}{2}}|\Sigma_x|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_x)^T \Sigma_x^{-1}(x-\mu_x)\right)$$

Measures of Goodness (L1)

- The L_1 variational distance

$$L_1(p_x, p_\theta) = \int_{\Omega_x} |p_x(x) - p(x;\theta)|\, dx$$

  - Related to how accurately your model computes the true probability of an event for ***any*** event A (where R_A is the region of X which defines the event A)

$$\begin{aligned} \Pr\{A\} &= \int_{R_A} p_x(x)\, dx \\ \frac{1}{2} L_1(p_x, p_\theta) &\geq \left| \int_{R_A} p_x(x)\, dx - \int_{R_A} p(x;\theta)\, dx \right| \end{aligned}$$

Measures of Goodness (KL)

- The Kullback-Leibler Divergence

$$\begin{aligned} D(p_x \| p_\theta) &= \int_{\Omega_x} p_x(x) \log\left(\frac{p_x(x)}{p(x;\theta)}\right) dx \\ &= E_X\left\{\log\left(\frac{p_x(x)}{p(x;\theta)}\right)\right\} \end{aligned}$$

  - It is the expectation of the log-likelihood function.
  - This is a directed measure – changing the order of arguments yields a different result.
  - Related to coding and quantization

Maximum Likelihood Density Estimation

- What do we do when we can't maximize the probability (e.g. when our samples come from a continuous distribution)?
- The maximum likelihood method chooses the parameter setting which maximizes the **likelihood** function (or some monotonically related function).

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(X;\theta) = \arg\max_{\theta} p(x_1,\ldots,x_N;\theta)$$

Maximum Likelihood Density Estimation

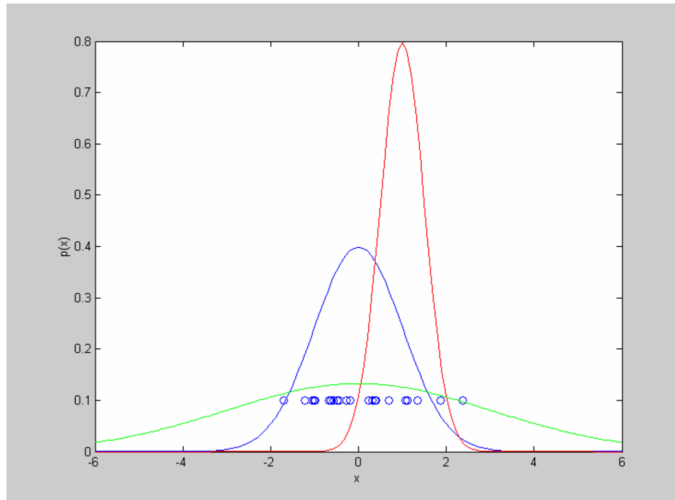
- If the samples are i.i.d. (independent and identically distributed) the likelihood function simplifies to

$$\begin{aligned}\hat{\theta}_{ML} &= \arg\max_{\theta} \prod_i p(x_i;\theta) \\ &= \arg\max_{\theta} \log\left(\prod_i p(x_i;\theta)\right) \\ &= \arg\max_{\theta} \sum_i \log\left(p(x_i;\theta)\right)\end{aligned}$$

- So why is this a good idea?

A Gaussian Example

Which density best explains the observed data?
Relate to K-L



Maximum Likelihood Estimate of Gaussian Density Parameters

Example:

- X are samples of an M-dimensional Gaussian random variable
- The set θ contains the mean vector and covariance matrix which completely specify the Gaussian density.
- P is the number of *independent* parameters which consists of M (for the mean vector) plus M (for the diagonal elements of the covariance matrix) plus (M^2-M)/2 (which is half of the off-diagonal elements – the other half are the same).
- The parameterized (model) density is then the Gaussian form with mean and covariance as parameters.

$$X = \{x_1, \ldots, x_N\} \qquad x_i \in \Re^M$$

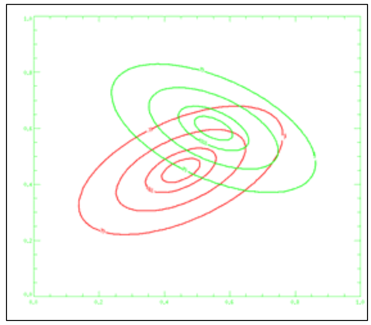
$$\theta = \left\{\mu_x \in \Re^M, \Sigma_x \in \Re^{M \times M}\right\}$$

$$\begin{aligned} P &= M + M + \frac{M^2 - M}{2} \\ &= \frac{3}{2}M + \frac{M^2}{2} \end{aligned}$$

$$p(x;\theta) = \frac{1}{(2\pi)^{\frac{M}{2}} |\Sigma_x|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_x)^T \Sigma_x^{-1} (x-\mu_x)\right)$$

2D Gaussian

- Gaussian Models



$$p(x;\theta_R) = \frac{1}{(2\pi)^{\frac{M}{2}} |\Sigma_R|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_R)^T \Sigma_R^{-1} (x-\mu_R)\right)$$

$$p(x;\theta_G) = \frac{1}{(2\pi)^{\frac{M}{2}} |\Sigma_G|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_G)^T \Sigma_G^{-1} (x-\mu_G)\right)$$

Maximum Likelihood Density Estimation

- The score function is the derivative of the (log) likelihood function with respect to the parameters.

$$S_\theta(X) = \frac{\partial}{\partial\theta} \sum_i \log\left(p(x_i;\theta)\right)$$

  - This derivative or gradient yields a system of equations, the solution to which gives the ML estimate of the density parameters
  - In the Gaussian case this results in a system of linear equations (woo hoo!).
  - More complicated models result in a nonlinear system of equations.

Maximum Likelihood Estimate of Gaussian Density Parameters

Example:

- The ML estimates of θ for the Gaussian turn out to be the sample mean and sample covariance.

$$\theta_{ML} = \left\{ \begin{aligned} \mu_{ML} &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ \Sigma_{ML} &= \frac{1}{N}\sum_{i=1}^{N} \left(x_i - \mu_{ML}\right)\left(x_i - \mu_{ML}\right)^T \end{aligned} \right\}$$

$$\hat{p}\left(x;\theta_{ML}\right) = \frac{1}{\left(2\pi\right)^{\frac{M}{2}} \left|\Sigma_{ML}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(x-\mu_{ML}\right)^T \Sigma_{ML}^{-1} \left(x-\mu_{ML}\right) \right)$$

- This is an example of a "Parametric" density estimate. First we compute some functions of the data (e.g. sample mean and covariance) and then plug the functions into some known form (e.g. the Gaussian).

Nonparametric Density Estimation

1. Normalized Histograms -> Convert to a PMF
2. Parzen Estimate
3. K-NN Estimate

- These methods are useful when the density exhibits more complex structure than a simple parameterized family.
- Convergence over a broader class of densities than any parametric density estimate (just more slowly).
- In contrast to Parametric estimates, nonparametric estimates are computed directly from our data samples.

Nonparametric Density Estimation

- Two common types are Parzen Windows and K-Nearest Neighbors (kNN)
  - consistency, bias, variance, convergence
  - quality measures
- They both exploit the following idea:

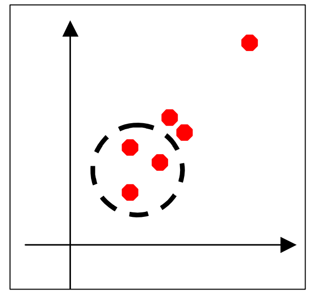
$$p(x) = \lim_{N \to \infty, V_x \to 0} \frac{k}{NV_x}$$

Nonparametric Estimation

- Assume the model has arbitrary form
- Estimate the function directly from samples
  - In some sense the model is parameterized directly from the samples

Nonparametric Density Estimation

- Generally, such estimates are "local" estimates.
  - Consequently the estimate at point x_1 is relatively unaffected by a "distant" point x_2
- Issues
  - need more samples for estimation at some points
  - uniform convergence rates are not always possible (i.e. the estimate is better in some regions of X than others).



$$p(x) = \lim_{N \to \infty, V_x \to 0} \frac{k}{NV_x}$$

Nonparametric Density Estimation

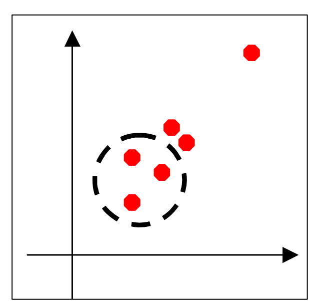
- Let's estimate the density function in the following way:
  - define a region L(x) about some point x.
  - estimate the probability of samples appearing in that region as

$$\hat{p}(x)v = k(x)/N$$

  - or

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

  - if v is fixed then k is a random variable, if k is fixed that v is a random variable



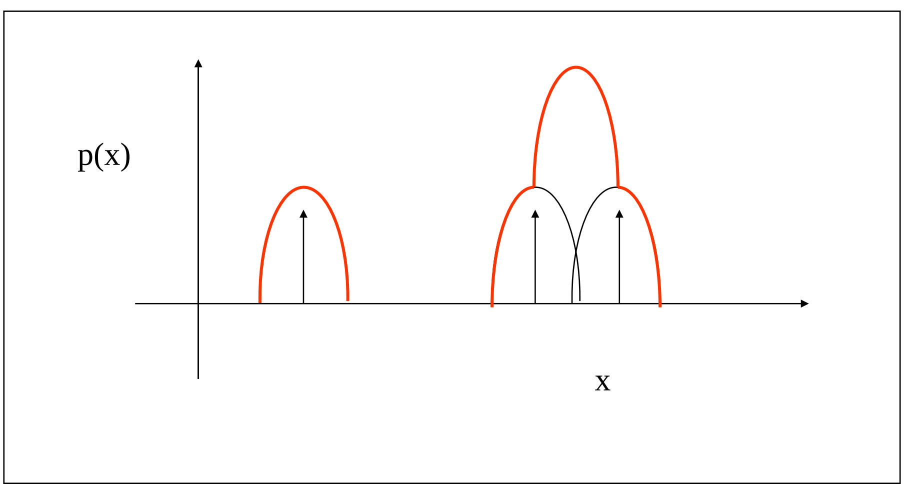
Use of Nonparametric Statistics

- The Parzen Density estimator

$$\hat{p}(x) = \frac{1}{Nh_N} \sum_{i=1}^{N} k\left(\frac{x - x_i}{h_N}\right)$$

- Convolution of a kernel with the data
- Kernel encapsulates “local” and “distance”
- Note that the kernel function is not necessarily constant which is a slight deviation from the “counting” argument on the previous slide.

Parzen Density Estimate



Variance and Bias (Parzen)

$$E\{\hat{p}(x)\} = \int p(u)k(x-u)du$$
$$= p(x) * k(x)$$

$$\lim_{N\to\infty} Nh \operatorname{var}[\hat{p}(x)] = p(x)\int k(x)^2 dx$$

Consistency Conditions (Parzen)

- k(x) is a density
- k(x) is "local".

$$\int k(x)dx = 1$$
$$k(x) > 0$$
$$\lim_{x\to\pm\infty} |xk(x)| = 0$$

Consistency Conditions (Parzen)

- These conditions ensure that the Parzen estimate is asymptotically unbiased

$$\lim_{N\to\infty} h_N = 0$$
$$\lim_{N\to\infty} Nh_N = \infty$$

where h_N loosely indicates that h is a function of N

k Nearest Neighbors Density Estimate

- The Parzen density fixed the volume (via the kernel).
- The kNN estimate varies the the volume (via k).

$$p(x) = \frac{k}{Nv(x)}$$

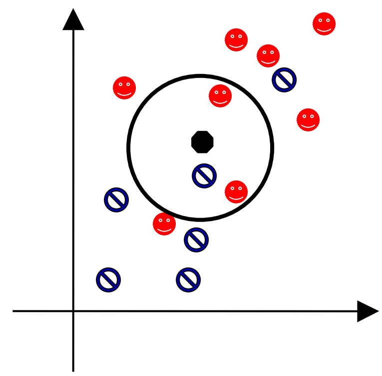
- The volume, v(x), is set such that at any point x it encloses k sample points.
- The Parzen density integrates to unity, the kNN density estimate does not.
- Early convergence results for classification.

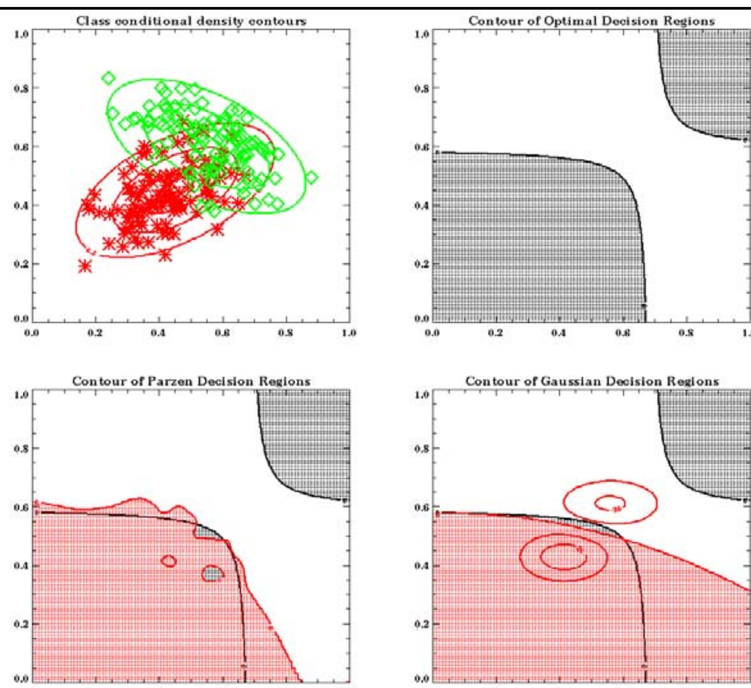
k Nearest Neighbors Classification

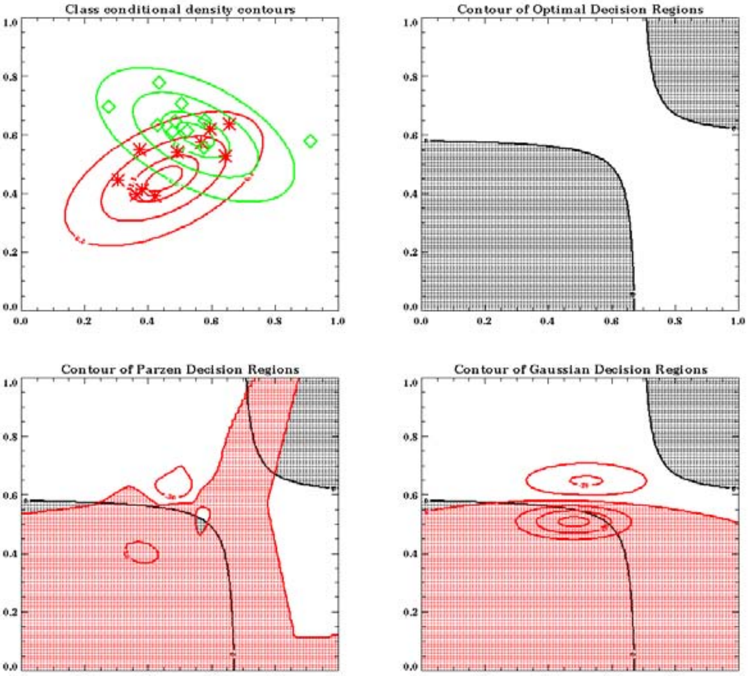
- One approach to classification might be to plug the density estimate directly into the Bayes' decision rule.
- However, K-NN provides a method for estimating the class directly (without the intermediate density estimate)

<u>K-NN Classification Procedure</u>

1. Given a new sample x_o increase the volume $v(x_o)$ until it encloses k sample points.
2. The class then corresponds to the majority.

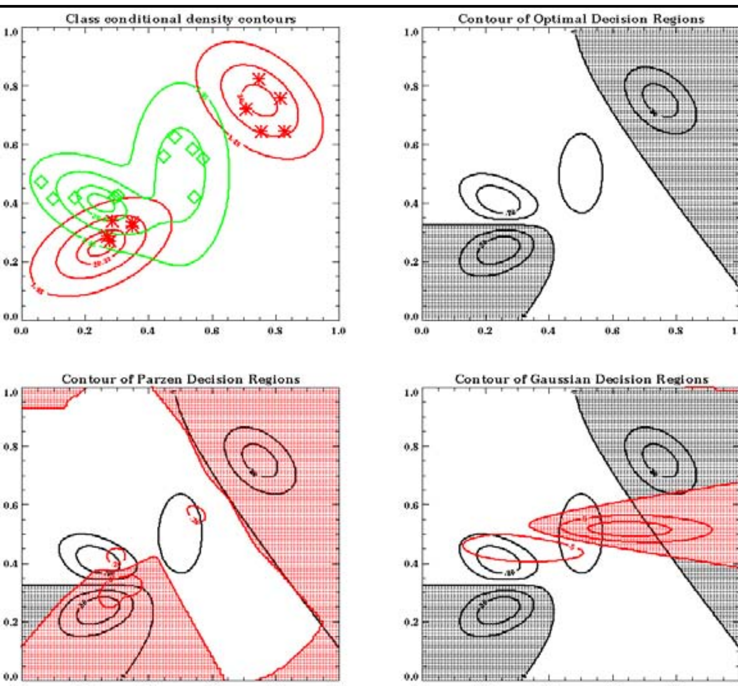






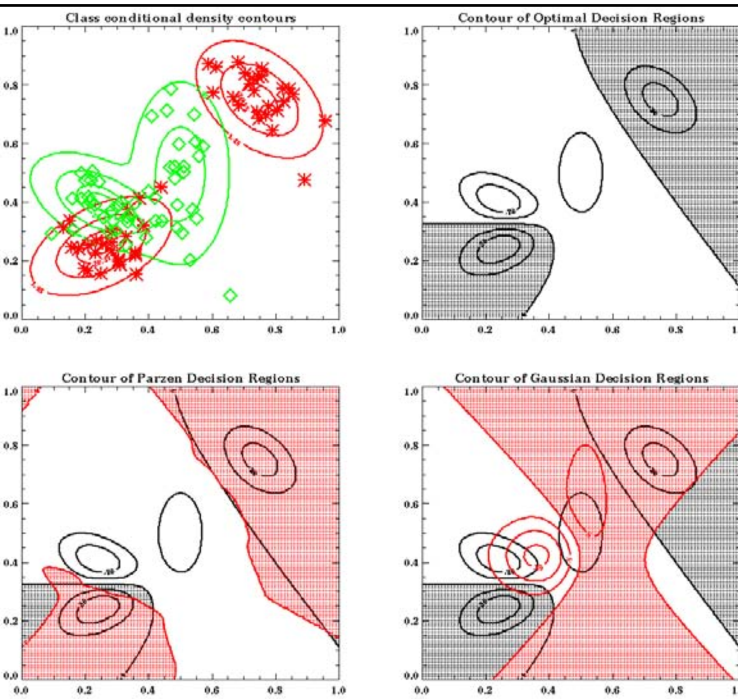
Class conditional density contours
Contour of Optimal Decision Regions
Contour of Parzen Decision Regions
Contour of Gaussian Decision Regions

49



Class conditional density contours
Contour of Optimal Decision Regions
Contour of Parzen Decision Regions
Contour of Gaussian Decision Regions

50


Class conditional density contours
Contour of Optimal Decision Regions
Contour of Parzen Decision Regions
Contour of Gaussian Decision Regions