What is Classification?

- A predictive modeling technique with categorical outputs (labels)
  - "orthogonal" (default) vs. hierarchically ordered labels.
  - Input: training records, each with a class label
  - Build: a model that can predict the label of future records of unknown class

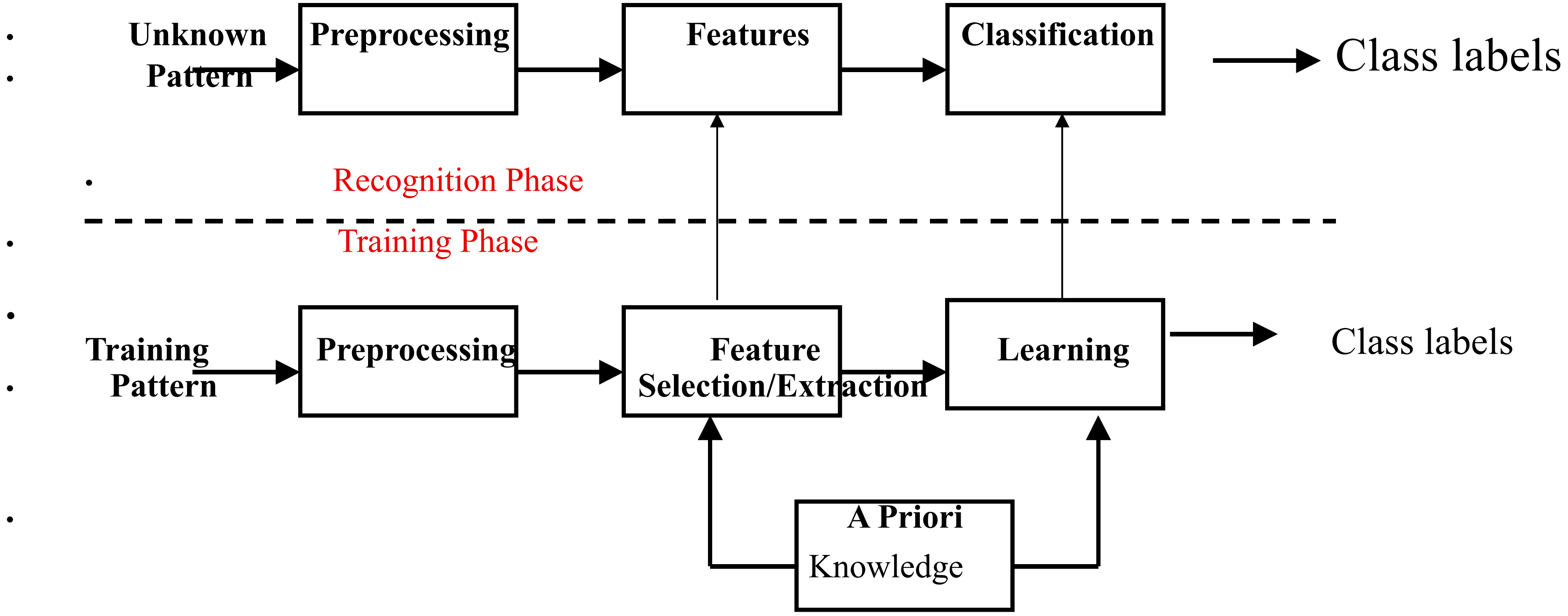
- **METHODS:**
  - decision trees (C4.5, CART, CHAID,...)
  - statistical/ Pattern recognition

    This philosophy "recognizes the probabilistic nature both of the information we seek to process, and of the form in which we should express the results".

    - Bayesian, k-nearest neighbor, logistic regression
    - neural networks

Training vs. Recognition

- divide given records into training, (validation), and test sets
  - "score" on future data
- true vs. estimated performance



A (Hard) Classification Model Partitions The Feature Space

- E.g. grading apples based on size and shine.

- Obtaining decision boundaries
  - Explicit
    - Youth if ht. < 5ft
  - vs. Implicit
    - Partitioning via discriminant functions $d_i(x)$
      - Object x belongs to class i iff $d_i(x) > d_j(x)$

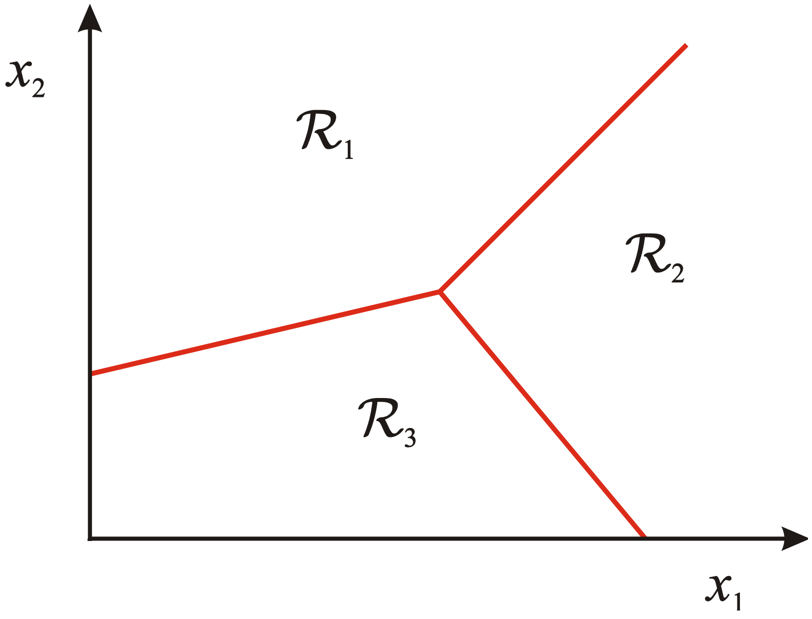
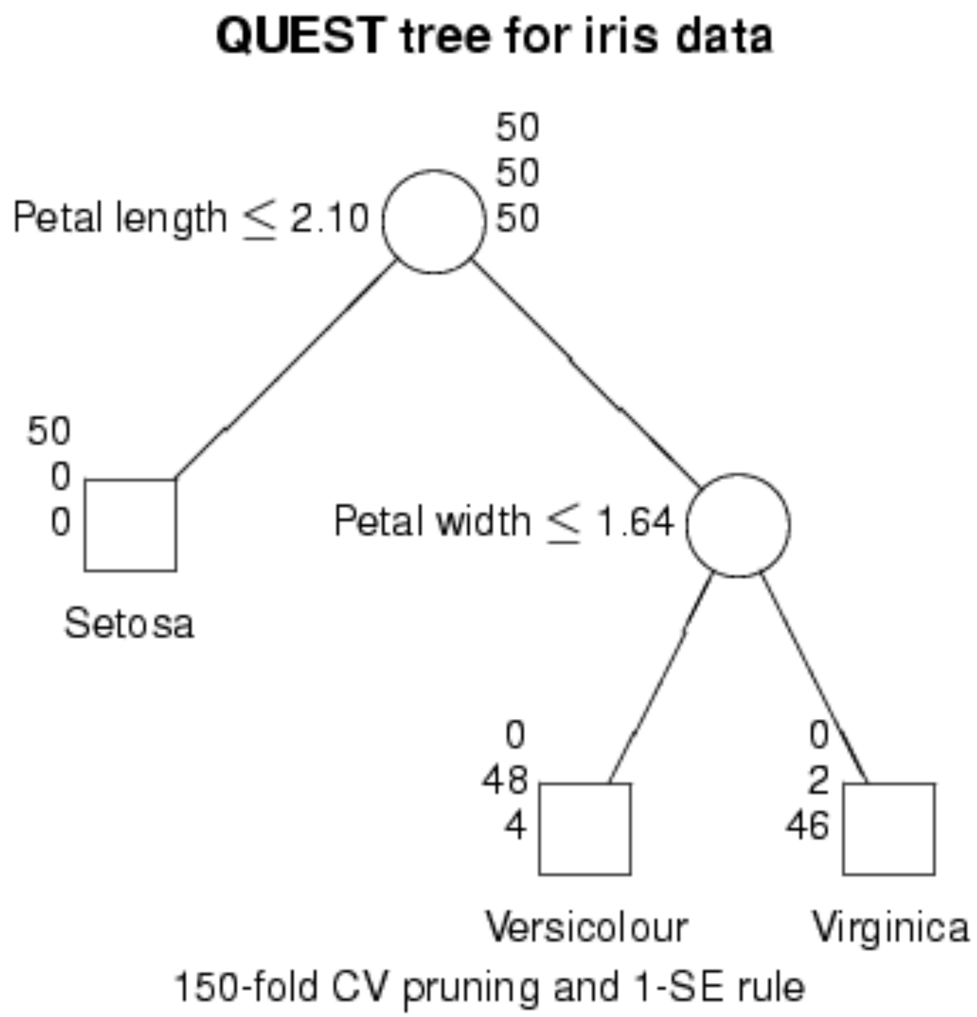


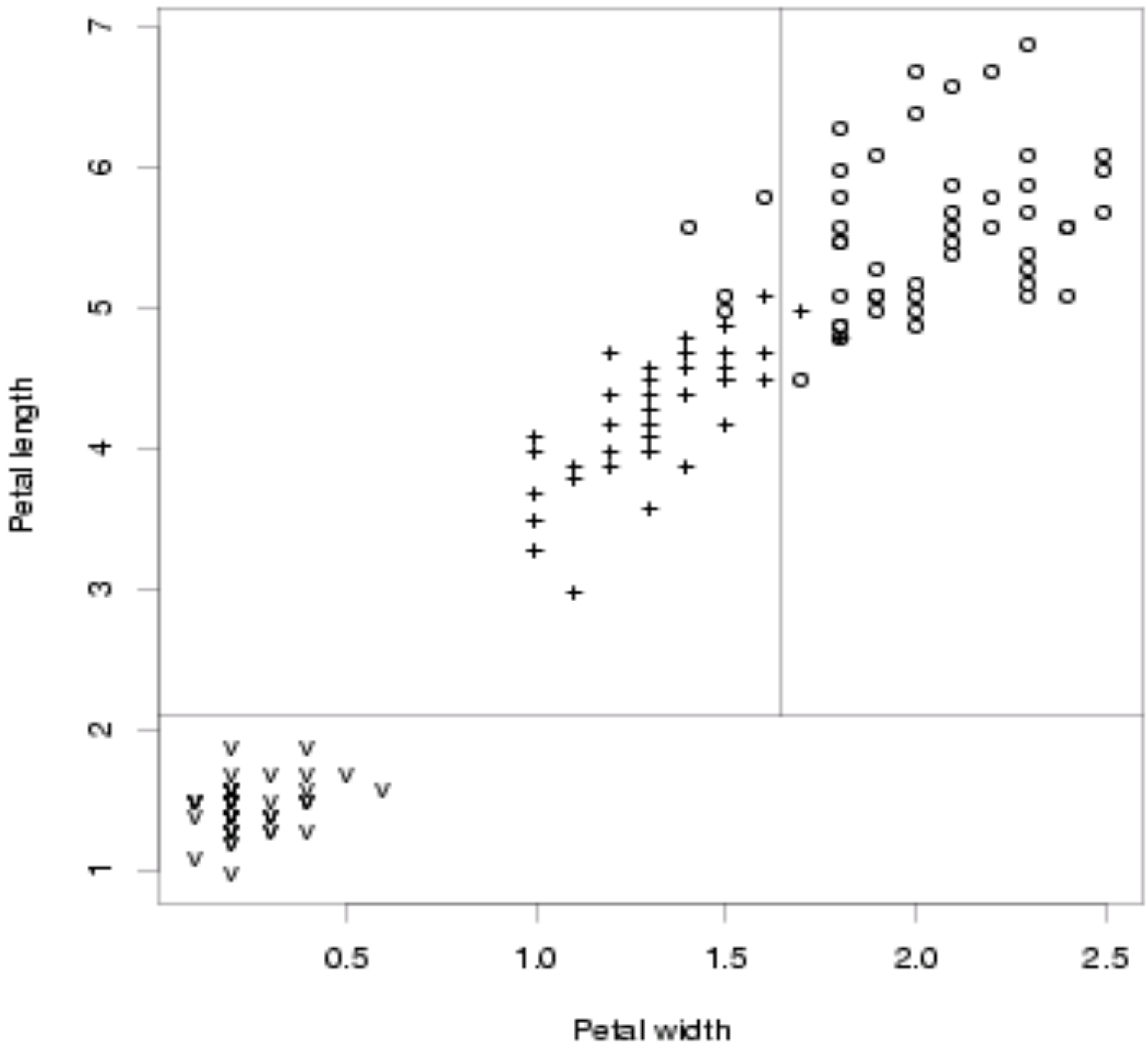
Fig: A 2-D input space partitioned into 3 regions for a 3 class problem

Decision Trees
(Simple but Popular Classifiers)

Decision Trees: Example



QUEST method



What Are Decision Trees?

- Hierarchical Classifier: breaks complex decision into series of simple decisions
  - each node performs a (single-variable) test to reduce uncertainty
    - For each input variable determine best split
    - Compare among the different variables to select best variable to split on
  - terminal nodes indicate samples (mostly) belonging to the same class (low uncertainty)
- goal: obtain small, shallow tree and low uncertainty (impurity) at the terminals

Decision Trees: Evaluation Functions

- Splitting: get "purer" children
  - (class probabilities (p_j' s) estimated empirically)

- Three popular evaluation functions:
  - entropy : $- \Sigma\, p_j \log p_j$

    tends to prefer attributes with many values
  - gini : $1 - \Sigma\, p_j^{\,2}$

  Both entropy and gini measure impurity at a node
  - use Impurity index: $\Delta\, i(n) = i(n) - p_{left}\, i(n_{left}) - p_{right}\, i(n_{right})$ to evaluate split, where $p_{left},\ p_{right}$ are also estimated empirically.

  - Chi-squared contingency table statistic Chi-Sq test with (r-1)x(c-1) degrees of freedom to test if two categorical variables are independent or not
    - Chi-Sq. stats: Σ_{cells} (Observed entry – Expected)^2 / Expected

  - Can generalize all three to k-ary splits

What size tree?

Question of Generalization

- Apriori termination criterion
- grow and prune

missing values?

  - Separate group
  - have secondary splitting variable

Decision Tree Packages

- C4.5 (Machine Learning) Uses entropy criterion for split
  - (maximizes information gain)
  - Commercial version is C5.0
- CART (Scientific/Stats): default is gini criterion
- CHAID (marketing/stats): uses Chi-sq; combines variables that are least discriminative,…
- …...

Note: CART also used for regression.

Evaluation of DTs:

- + simple, intuitive, often fast, explainable
- + integrates feature selection with classification
- + can "handle" missing values

- - often substantially poorer performance
- - limited: problems with complex decision boundaries, correlated features, continuous variables, …
- - unstable (partially addressed by bagging/boosting ensemble techniques)

Scalable, Parallelizeable Decision Trees*

- In **distributed** computing environments: For SPARK implementation, see the blog at

http://databricks.com/blog/2014/09/29/scalable-decision-trees-in-mllib.html

and its “Further Readings” section for slides and Video

Streaming (and Parallel) Decision Trees. See

www.jmlr.org/papers/v11/ben-haim10a.html

- The essence of the algorithm is to **quickly construct histograms at the processors**, **which compress the data to a fixed amount of memory.** A master processor uses this information to find near-optimal split points to terminal tree nodes. Our analysis shows that guarantees on the local accuracy of split points imply guarantees on the overall tree accuracy.

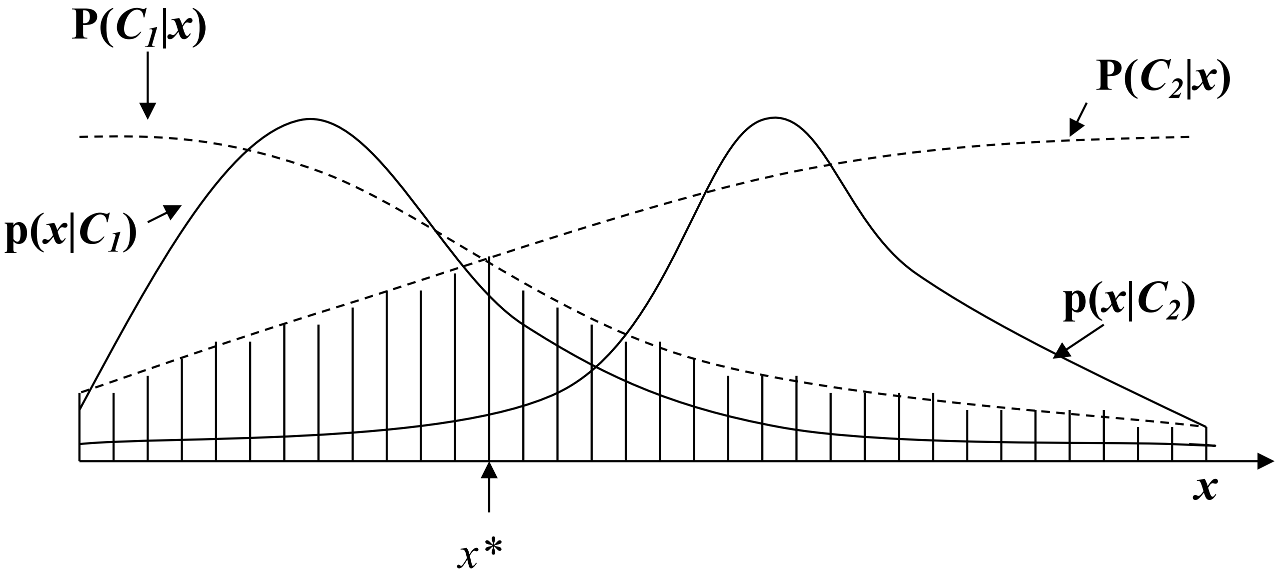
Pattern Recognition Approaches

Founded on Bayes Decision Theory

See Bishop Ch 1.5, and Chapter 2 of Richard O. Duda, Peter E. Hart, and David G. Stork (2001). *Pattern Classification*. Wiley. (DHS) e-book available from UT Libraries

http://www.ai.mit.edu/courses/6.891-f00/text/

Bayes Decision Theory



$P(C_i|x)$ – *a posteriori* probability

$p(x|C_i)$ – (class conditional) likelihood function

$P(C_i)$ – class priors

Bayes Classifier

The Bayesian classifier is a parametric method based on

- The *a priori* distributions of classes P(C_i)
- The probability distributions p(x |C_i)
- The *a posteriori* distributions of classes P(C_i|x)

The Bayesian classifier is a MAP classifier (*maximum a posteriori*) : an observed pattern x is classified as class C_i if

$$i = \underset{j=1 \ldots. K}{\operatorname{argmax}} \{P(C_j \mid x)\}$$

where K is # classes, and

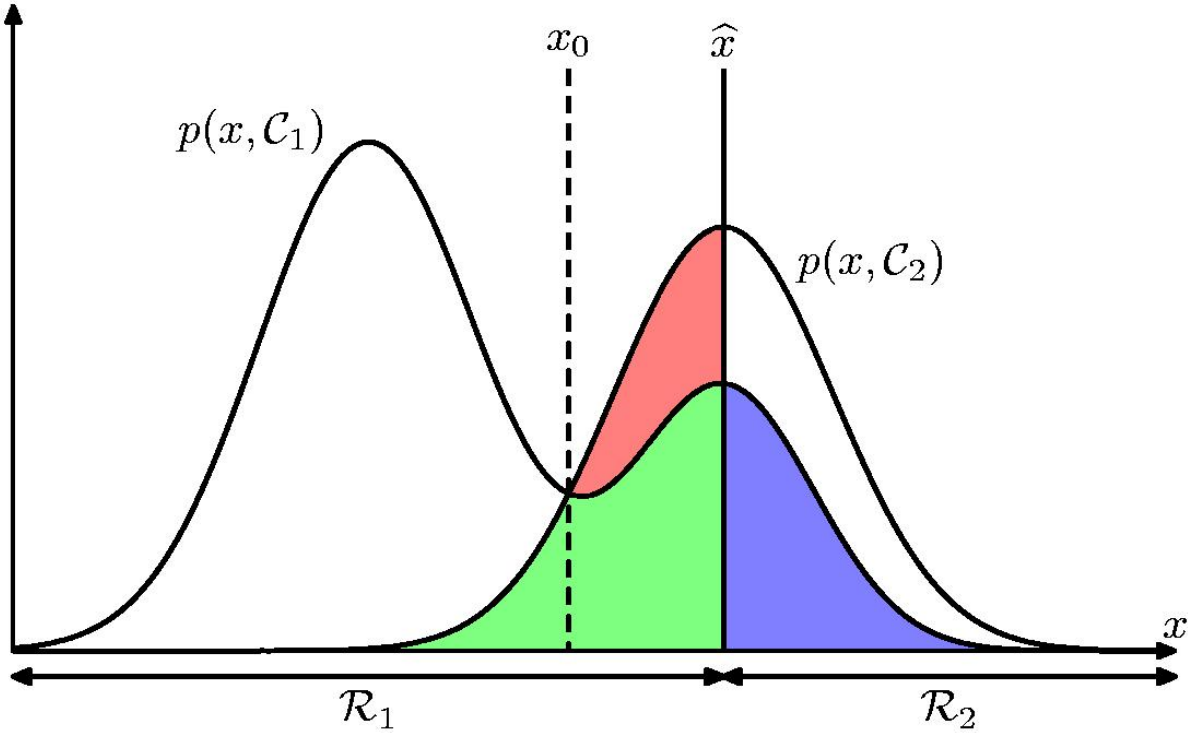
$$P(C_i | x) = \frac{P(C_i) p(x | C_i)}{p(x)}$$

Outcome: partitioning of the input space based on "discriminant functions"

The Bayesian classifier is optimal: it statistically minimizes the error rate

- Catch??

Misclassification Rate



$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x}. \end{aligned}$$

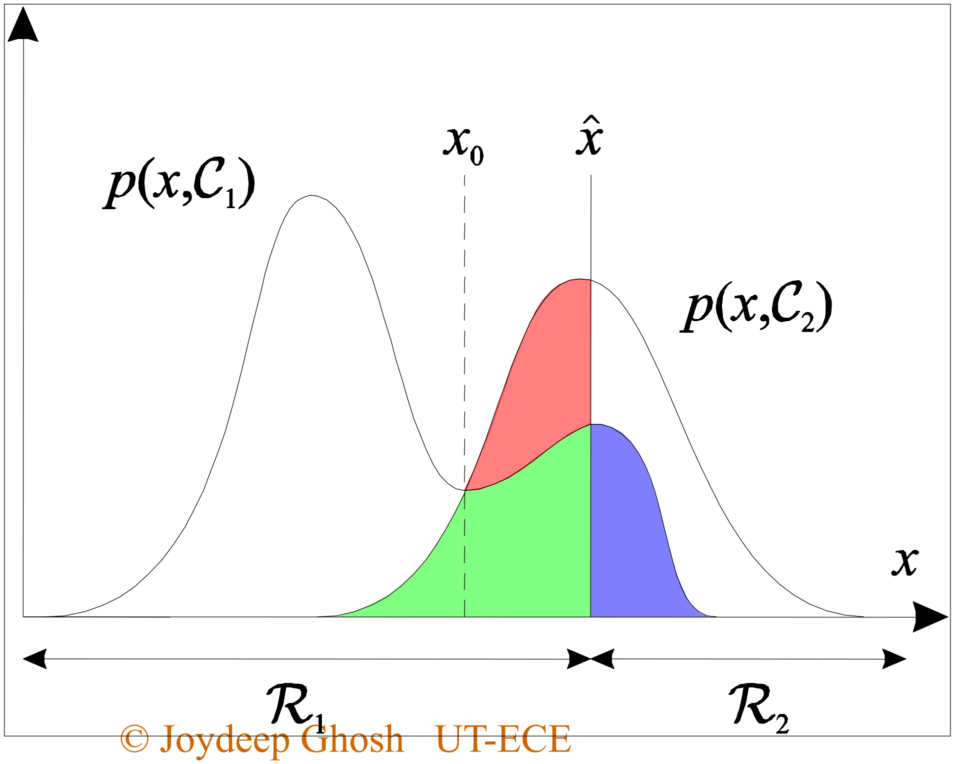
Optimal Decision for Min. Misclassification Rate

- Optimal decision boundary at x_0.
  - P(error) = P(class 2 -> class 1) + P(class 1 -> class 2)

    = (green area) + (blue area)

So extra error because of non-optimal design = red area.

Ques: show the extra error if boundary is chosen to left of x_0



Minimum Expected Loss

- **Example: classify medical images as ‘cancer’ or ‘normal’**

Decision

$$\text{Truth}\quad\begin{array}{l c} & \begin{array}{cc}\text{cancer} & \text{normal}\end{array} \\ \begin{array}{l}\text{cancer} \\ \text{normal}\end{array} & \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}$$

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}$$

Regions $\mathcal{R}_j$ are chosen to minimize

$$\mathbb{E}[L] = \sum_k L_{kj} p(\mathcal{C}_k | \mathbf{x})$$

How to Minimize Loss?

- **Example**: consider a loss matrix:

		Decision	
		C1	C2
Truth	C1	0	5
	C2	4	-2

Denote P(C1|x) as y(x) for convenience.

For what range of y(x) do we get lower expected loss by assigning to Class 1?

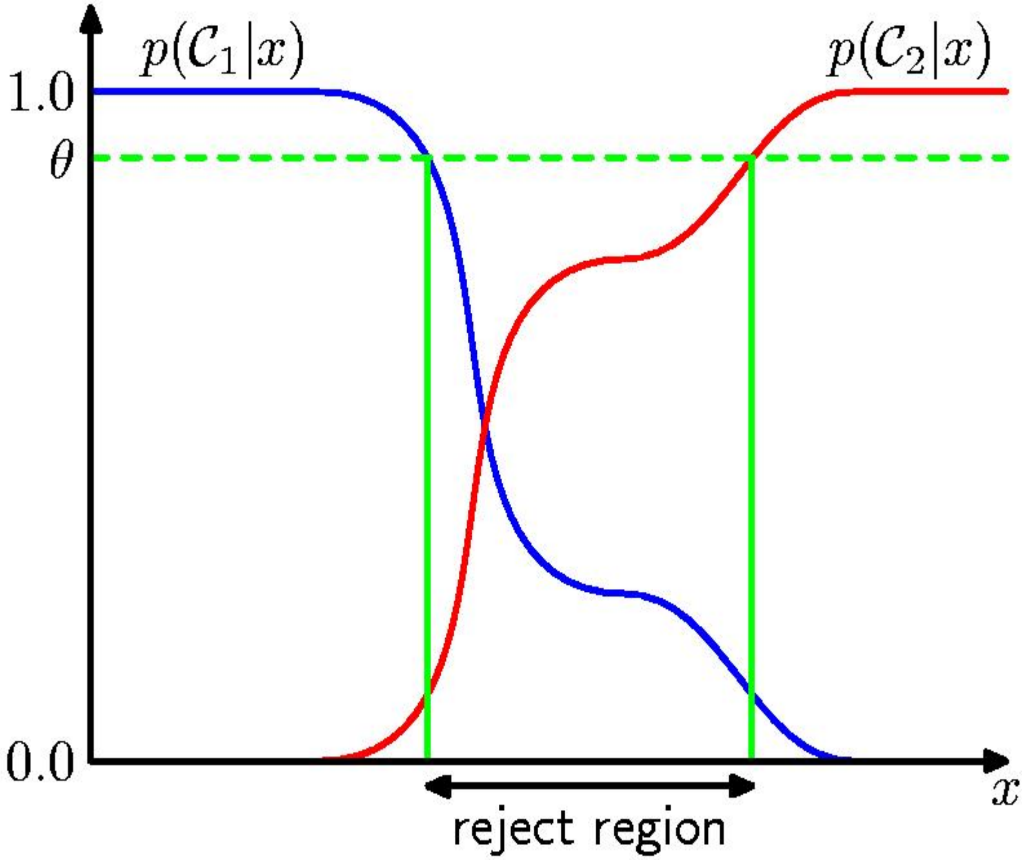
Solution sketch:

Expected loss if x is labeled as class 1 = 0. y(x) + 4 (1-y(x))

Compare with Expected loss if x is labeled as class 2

At the boundary both losses are equal.

Reject Option



Reject Option Example

- **Example**, consider a loss matrix:

		Decision		
		C1	C2	Reject
Truth	C1	0	M_1	R
	C2	M_2	0	R

WLOG assume $M_1 > M_2$.

For what range of P(C1|x) am I better off rejecting?

Why Separate Inference and Decision?

- Inference: estimate the $P(Ci|x)$ terms
- Decision: allocate a given x to a specific class.

- Minimizing risk (loss matrix may change over time)
- Reject option
- Changed class priors in scoring data
- Combining models (later in the course)

A Class can Span Disconnected Regions

- Example from DHS 2001(copy of Ch1-7 of DHS at http://www.ai.mit.edu/courses/6.891-f00/text/)

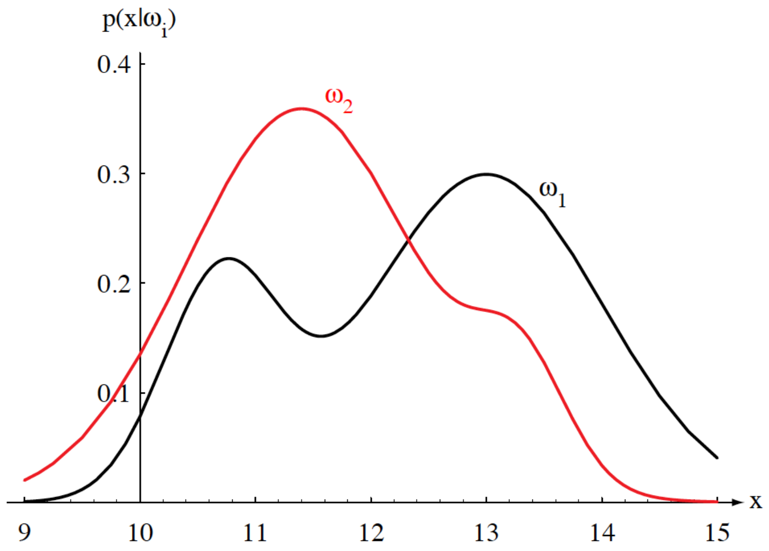


Figure 2.1: Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value x given the pattern is in category ω_i. If x represents the length of a fish, the two curves might describe the difference in length of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0.

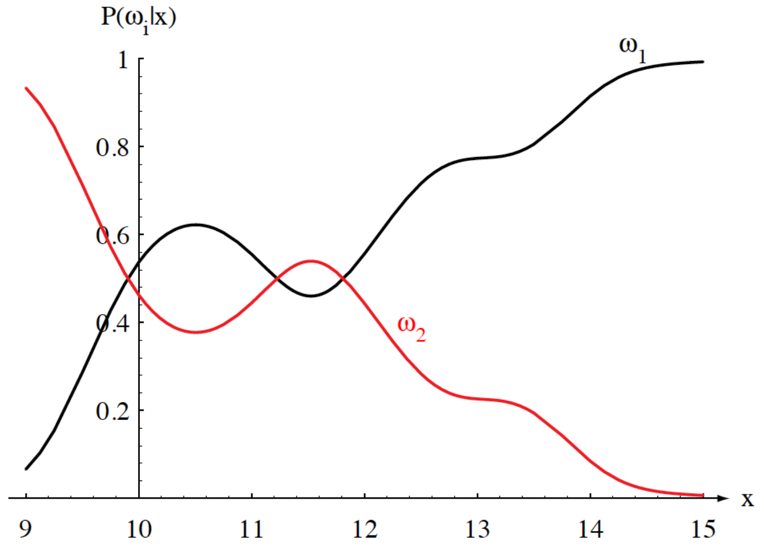


Figure 2.2: Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x, the posteriors sum to 1.0.

Solving for Optimal Boundaries

- In general it is difficult to get closed form expressions for optimal decision boundary
  - But possible for special cases, most notably when the (class-conditional likelihoods are normally distributed. (**READ DHS Sec 2.6**)
  - Formula for multivariate (d>1) normal distribution:

$$(1) \qquad p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Where $\boldsymbol{\Sigma}$ is the covariance matrix. Note both $\mathbf{x}$ and $\boldsymbol{\mu}$ are vectors.

Boundary between class i and j means $P(C_i\,|\,\mathbf{x}) = P(C_j\,|\,\mathbf{x})$

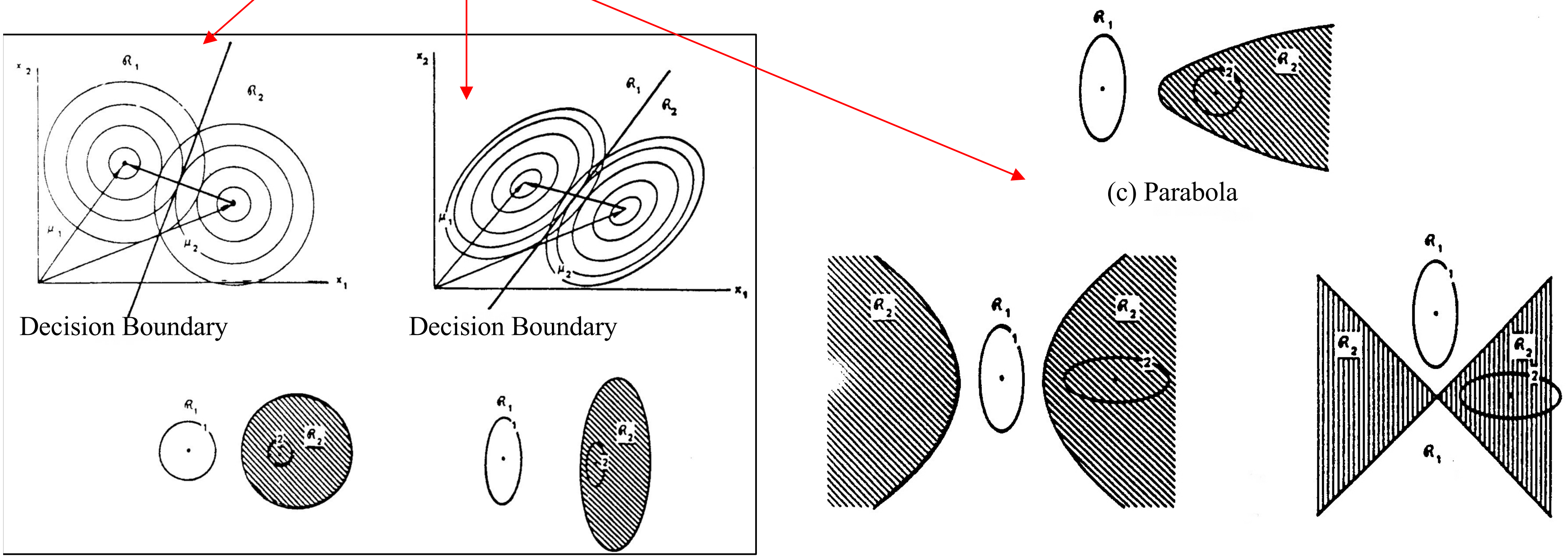
More conveniently, we have

$$\ln(p(\mathbf{x}|C_i)) + \ln(P(C_i)) = \ln(p(\mathbf{x}|C_j)) + \ln(P(C_j)) \qquad (2)$$

Decision boundaries

Forms for Decision Boundaries for Different Cases of Bivariate Gaussian Distributions

Case1: Identical covariance matrices (diagonal)
Case2: Identical covariance matrices (nondiagonal)
Case3: Different covariance matrices.



(c) Parabola

(a) Circle

(b) Ellipse

(d) Hyperbola

(e) Straight lines

2-D Example from DHS

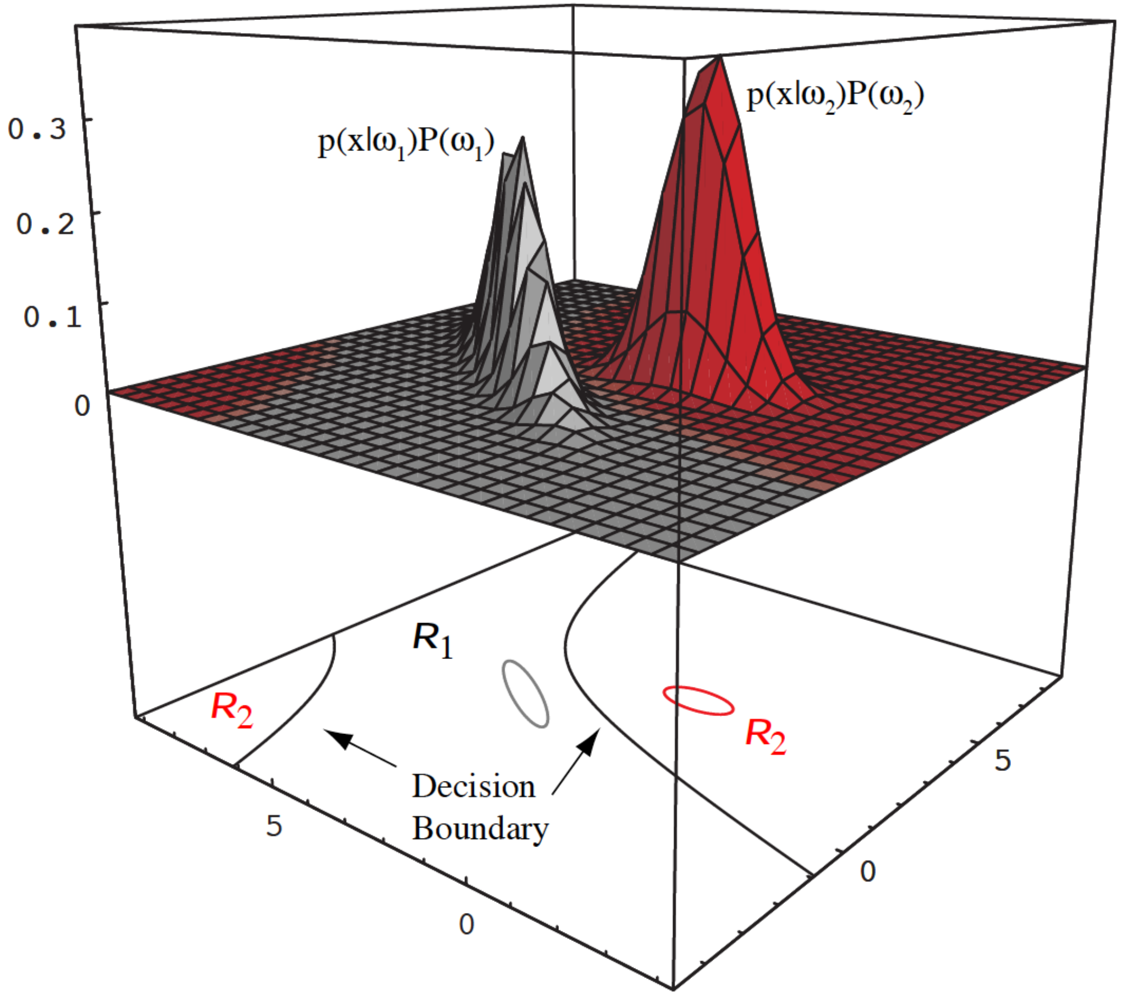


Figure 2.6: In this two-dimensional two-category classifier, the probability densities are Gaussian (with $1/e$ ellipses shown), the decision boundary consists of two hyperbolas, and thus the decision region $\mathcal{R}_2$ is not simply connected.

Visuals for Bivariate Gaussians (from DHS)

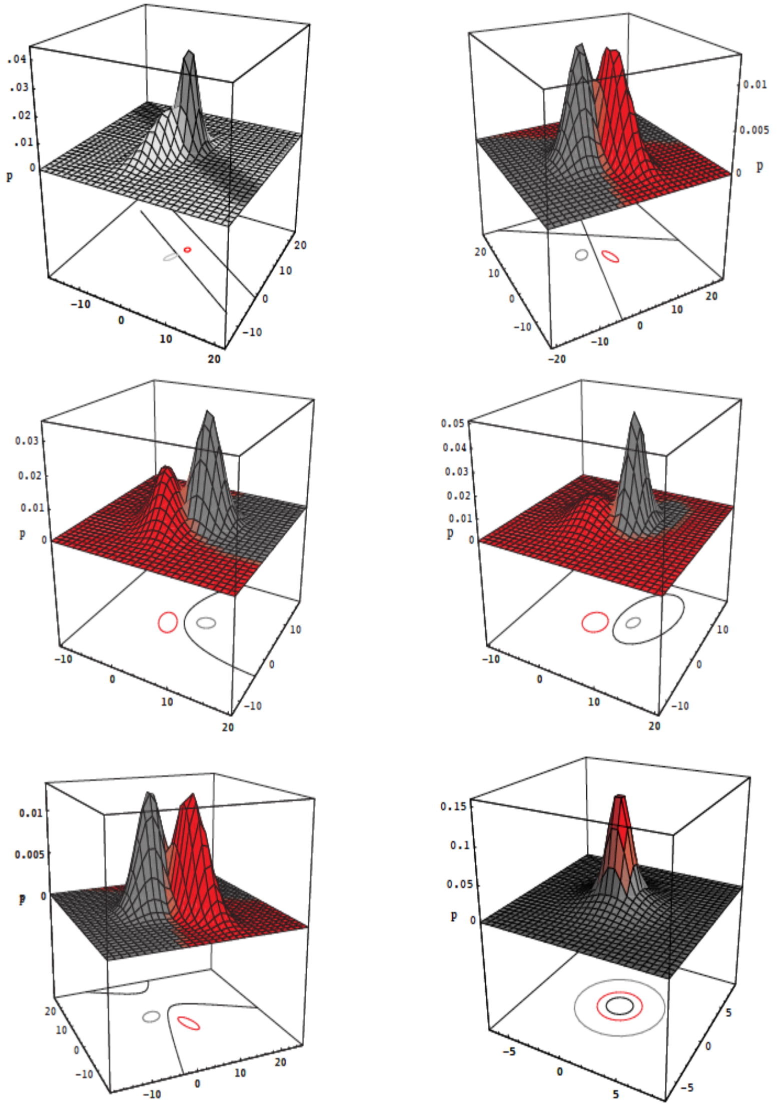


Figure 2.14: Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadratic, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric.

Discriminant Functions for Normal Distribution

From (1), we have

$$\ln\big(p(x)\big) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}) \; + \text{constants}$$

Substitute this form in (2) to get the solution!

=> quadratic boundaries in general. (**Quadratic Discriminant Analysis: estimate Σ separately for each class**)

Note: if $\Sigma_i = \Sigma_j$ = Then quadratic terms cancel, so get linear boundary!

(**Linear Discriminant Analysis or LDA: uses "pooled" covariance estimation**)

Also, if all priors are the same as well, then

Solution = assign to closest (Mahalonobis distance if Σ is diagonal) mean

—think of mean as a template/prototype

Why Bayes rate is not achieved ?

- Unknown distributions, priors
- Finite training set
  - Leads to estimation errors
- Noisy samples; mislabeled samples
- Missing values
- Symbolic vs. numerical attributes
- (lack of) constraints about problem domains

Bayes rate = function of features used

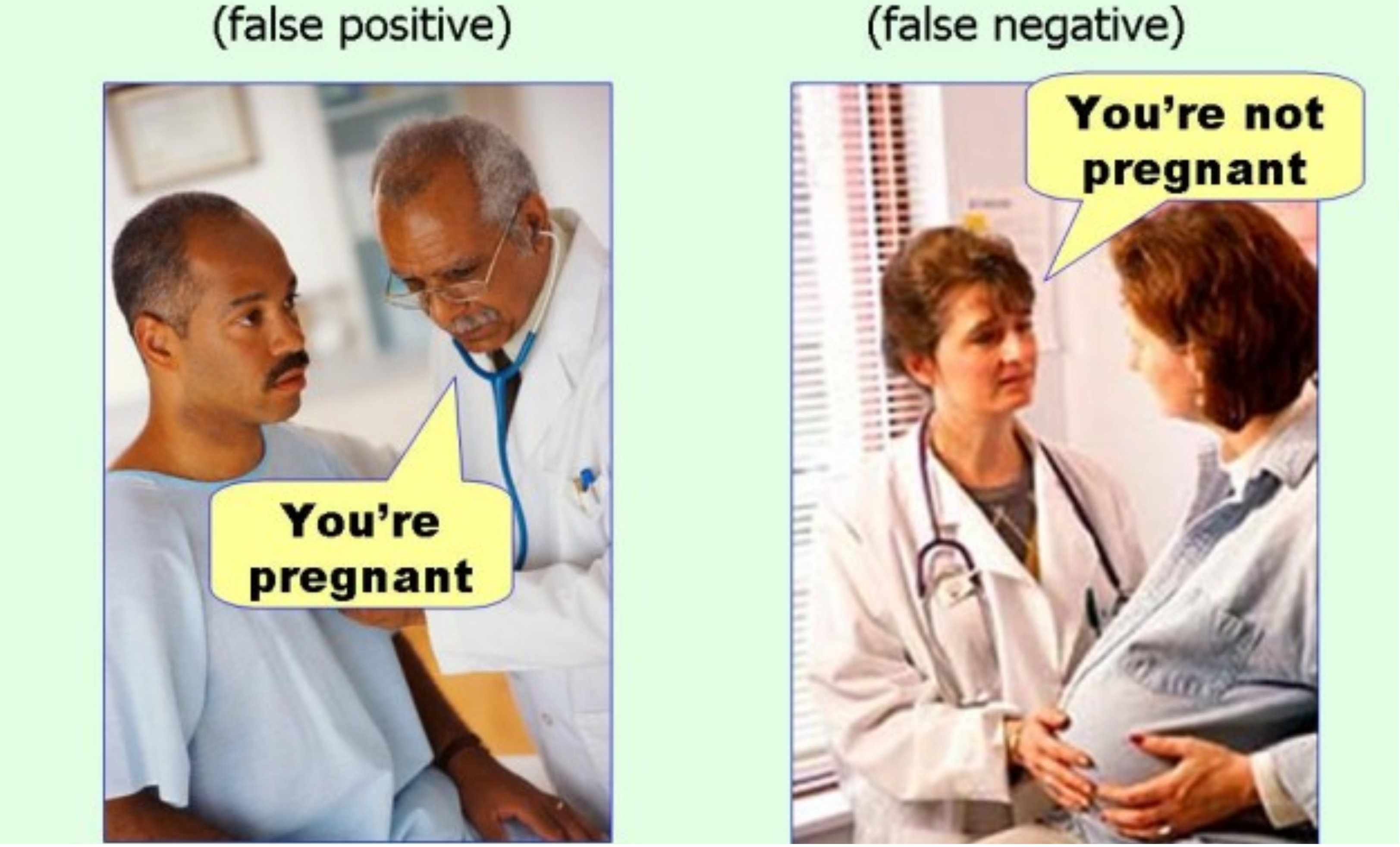
Evaluating Results (of any classifier)

- (correct/mis)-classification rate
  - estimate via holdout, cross-validation,...
- loss or profit
- confusion matrix
  - actual (rows) vs predicted class
  - for two class problems (+ve and –ve class)

we have:

		Predicted class	
		+ve	-ve
Actual class	+ve	True Positive (TP)	False Negative (FN)
	-ve	False positive (FP)	True Negative (TN)

Type I vs Type II Error

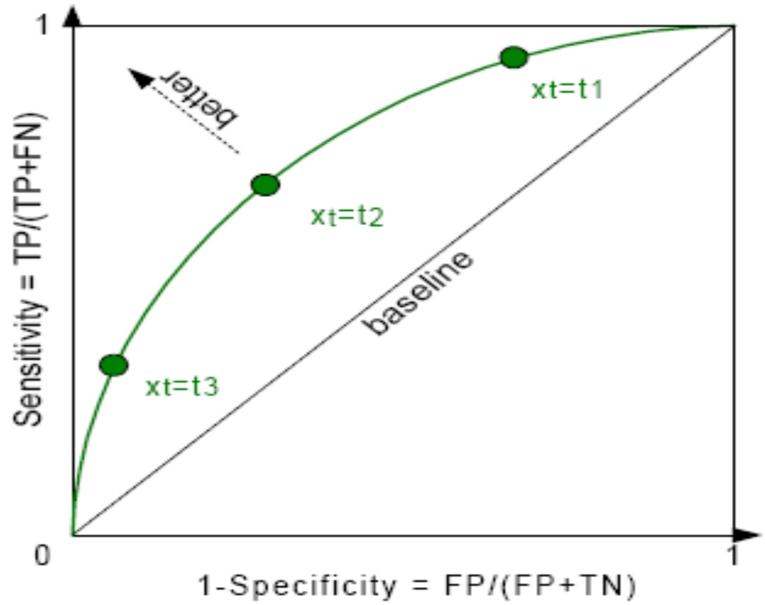


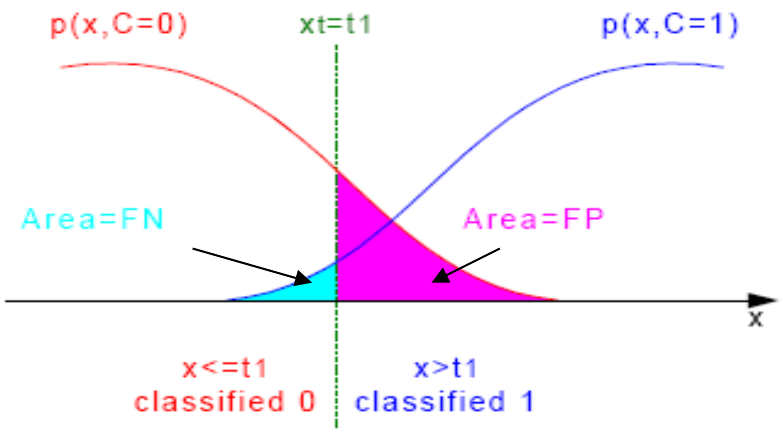
Type III error: when you get the right answer to the wrong question.

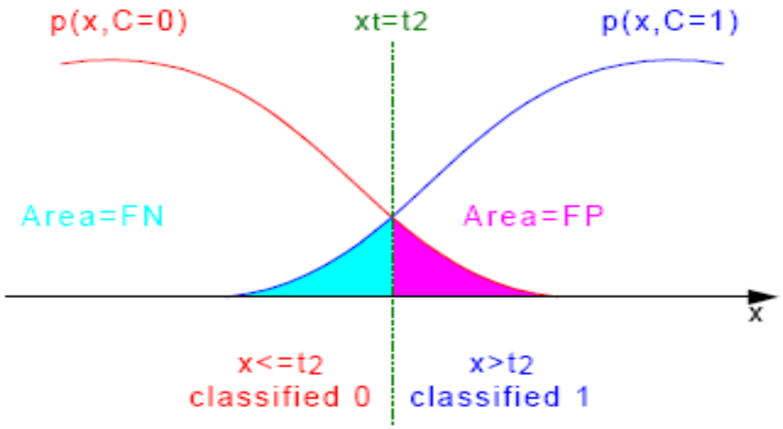
Assume C=1 denotes positive class.

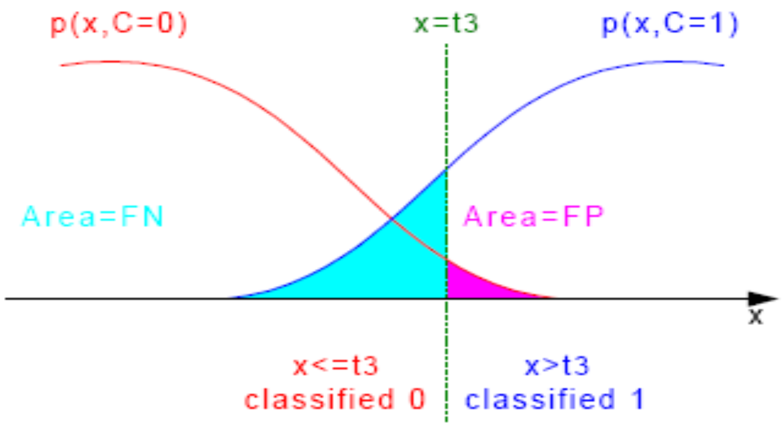
Both entries in a "predicted class" column increase or decrease together.

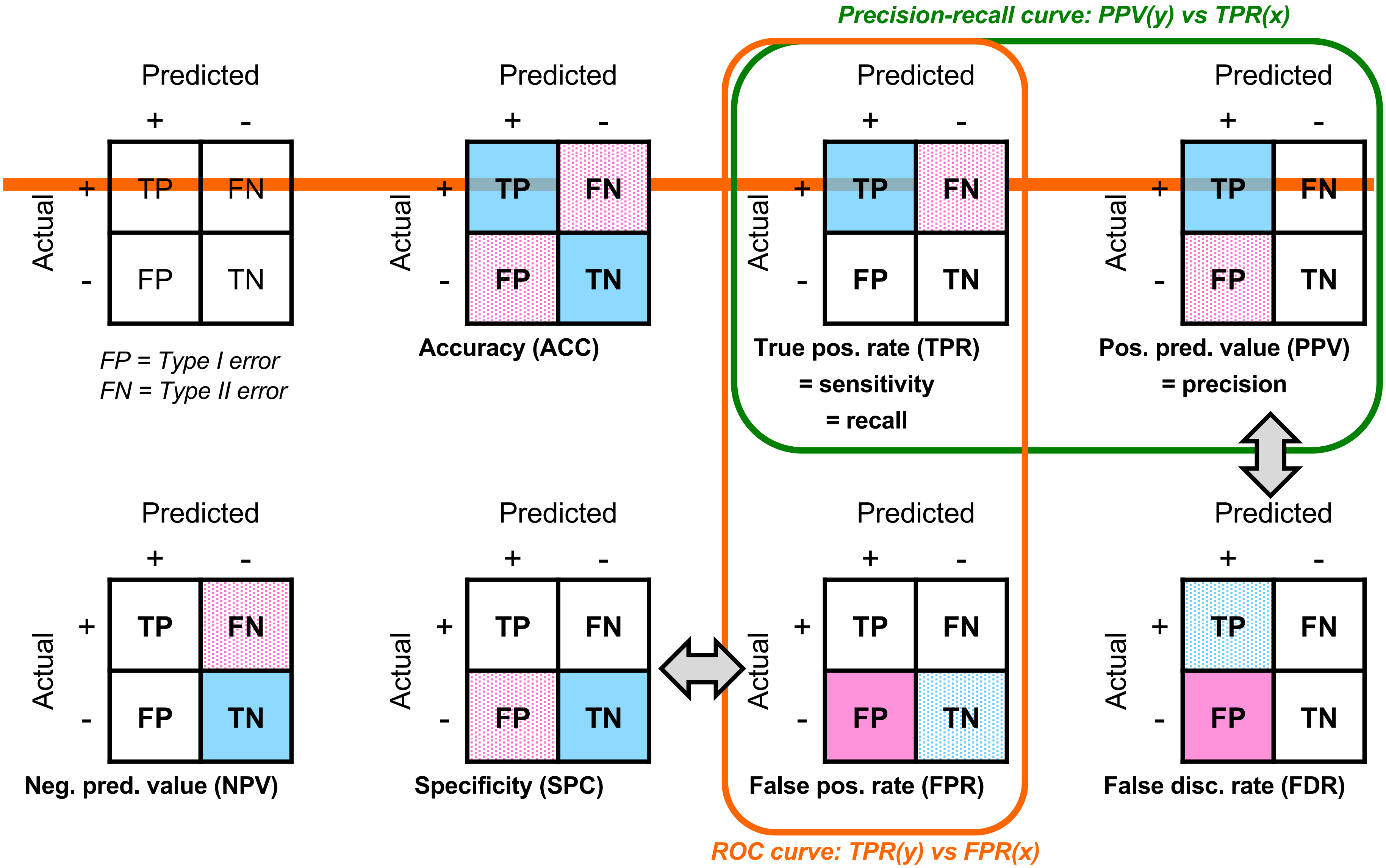
Since total number of samples is same, this means A trade-off between Type I and Type II errors











Value: between 0 and 1 (numerator/denominator)
Numerator = solid color shading
Denominator = solid + partial shading

⇔ *"one minus" relationship*

Relationship between ROC and precision-recall:

$$PPV = \frac{P(TPR)}{P(TPR)+N(FPR)}$$ *(ROC to P-R)*

$$FPR = \frac{P(1-PPV)(TPR)}{N(PPV)}$$ *(P-R to ROC)*

"P" = # of actual +ves; "N" = # of actual –ves.

Receiver Operating Characteristic (ROC) Charts

- CLASSICAL (Detectors): **% detected (TPR) vs. false alarm rate (FPR)**
  - **(roots in WW II)**
- Medicine: sensitivity vs. (1- specificity) for a range of cutoffs.
  - TP/(TP+FN) vs. FP/(FP+TN)
  - Also talk of Positive Predictive Value (PPV) and Negative Predictive Value (NPV)
- Bayesian Analogues:
  - Sensitivity (Specificity): Accuracy of positive (negative) call conditioned on positive (negative) class.
  - PPV (NPV): Unconditional accuracy of positive (negative) call
- Info. Retrieval: *recall* (fraction of relevant documents retrieved) vs *precision* (fraction of retrieved documents that are relevant)

What is a good looking ROC curve?

- **See an applet and a tutorial**

Calibration

- What is Calibration?
  - Good estimates of posterior probabilities
- When is it important?

- What happens to the ROC when all estimates of P(C|x) are multiplied by a constant?

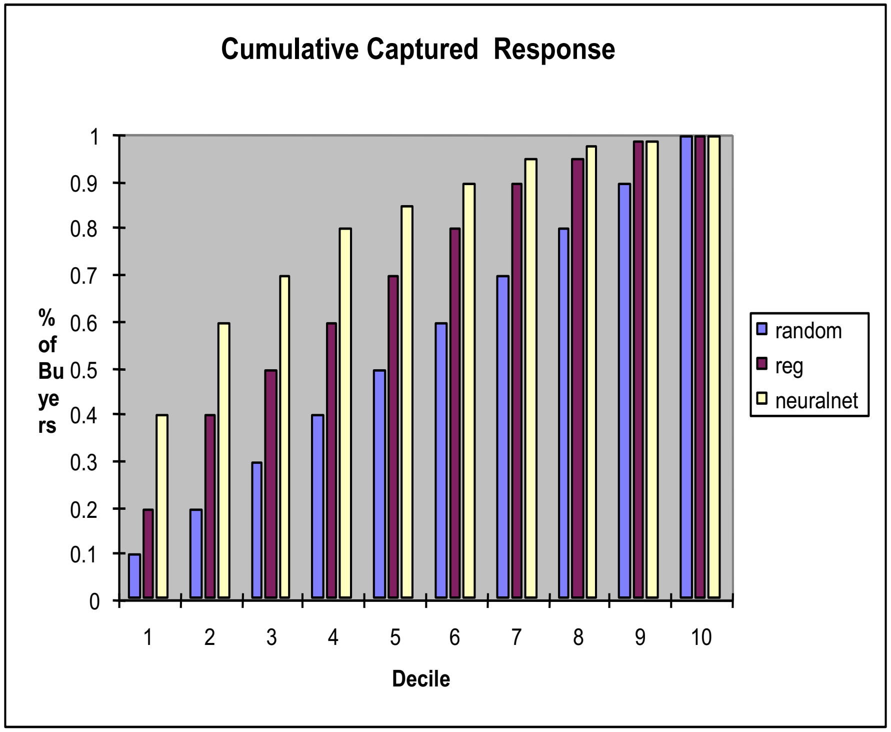
Lift Charts (gains chart)

- sort all observations
  - from highest expected profit to lowest expected profit (non-binary).
  - by the posterior probabilities of the target event (binary targets)
- group into deciles and plot
  - typically, cumulative plot of "% of target events in selection"
    - often normalized by "% in a random selection"
      e.g.: top decile provides a lift of 2.5

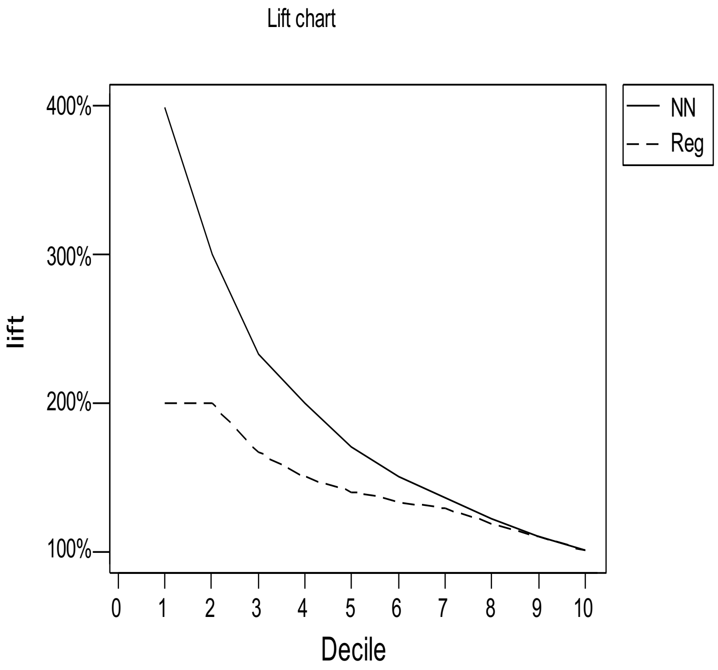
      Often only top 10% or top 25% is of interest

Lift Example (direct marketing)

Cumulated Lift Chart



Lift Chart



Example: Logistic Regression can capture 50% of the buyers by mailing to 30% of target audience; neural net can capture 70% of the buyers by mailing to 30% of the audience.

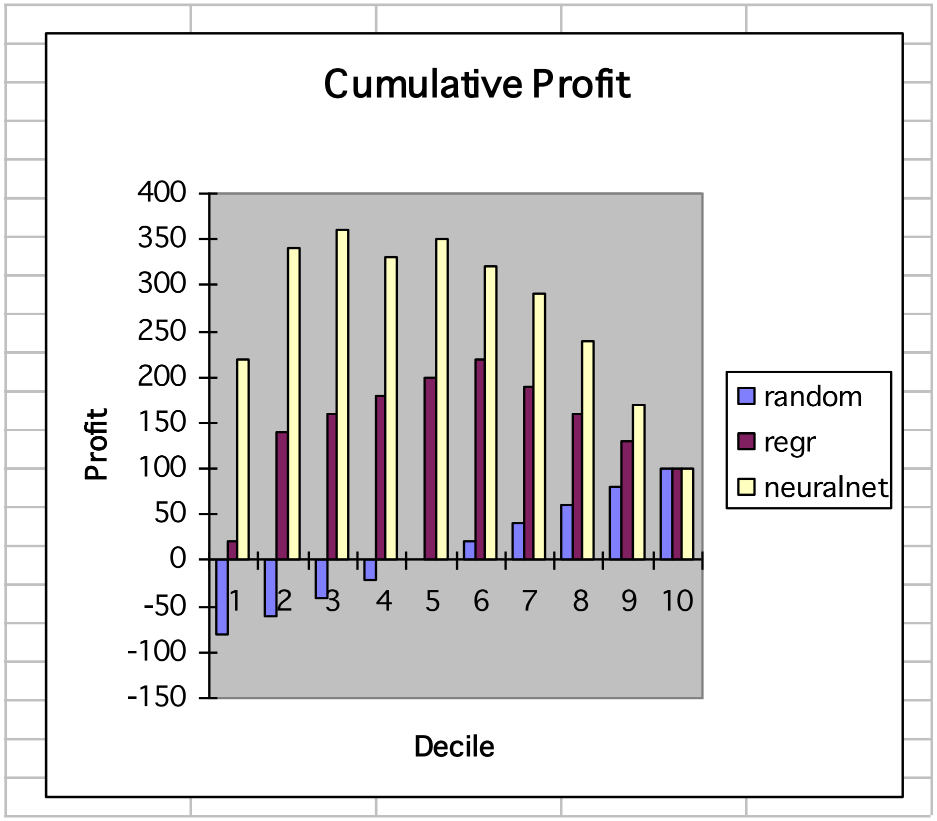
Profit and ROI Charts

- More important measure is expected profits or return on investment.
- This combines information about how well the campaign works with information on
  cost and value of customers to help with the decision of how to maximize profits.
  - Example: Recall \$4/target, 4% response rate, 200K individuals targeted.
    Suppose fixed cost = \$100K; value of each customer was \$125/yr.

    C = Fixed Cost + Per Customer Cost * mailing rate * Popn size
    E = expected # of buyers * Avg. Cust. Value
       = (Lift * mailing rate) * .04 * 200K * \$125

    Want to maximize Profit = E - C (or perhaps ROI = E/C)
  - How many should you send it to?
  - Suppose we choose to maximize Profit

Predicted Cumulative Profit



Shows: better modeling yields higher profits
The $360K earned by using neural network outperforms non-targeting mailing by 300%; and more profit is earned with a smaller target.

Backups

Building Decision Trees for Large Datasets

- Large database? Growing 99% of time; pruning 1%
- Time consuming operations (at each split node):
  - scan database (disk access) to update class counts
  - For each variable, sort records based on value of that variable

- Problems due to unstable model:
  - Sampling
    - problematic for exact answer but may be OK in practice
  - partition data into subsets; train multiple classifiers in parallel; combine outputs
    - reduce runtime significantly, but interpretability loss
  - Incremental learning?

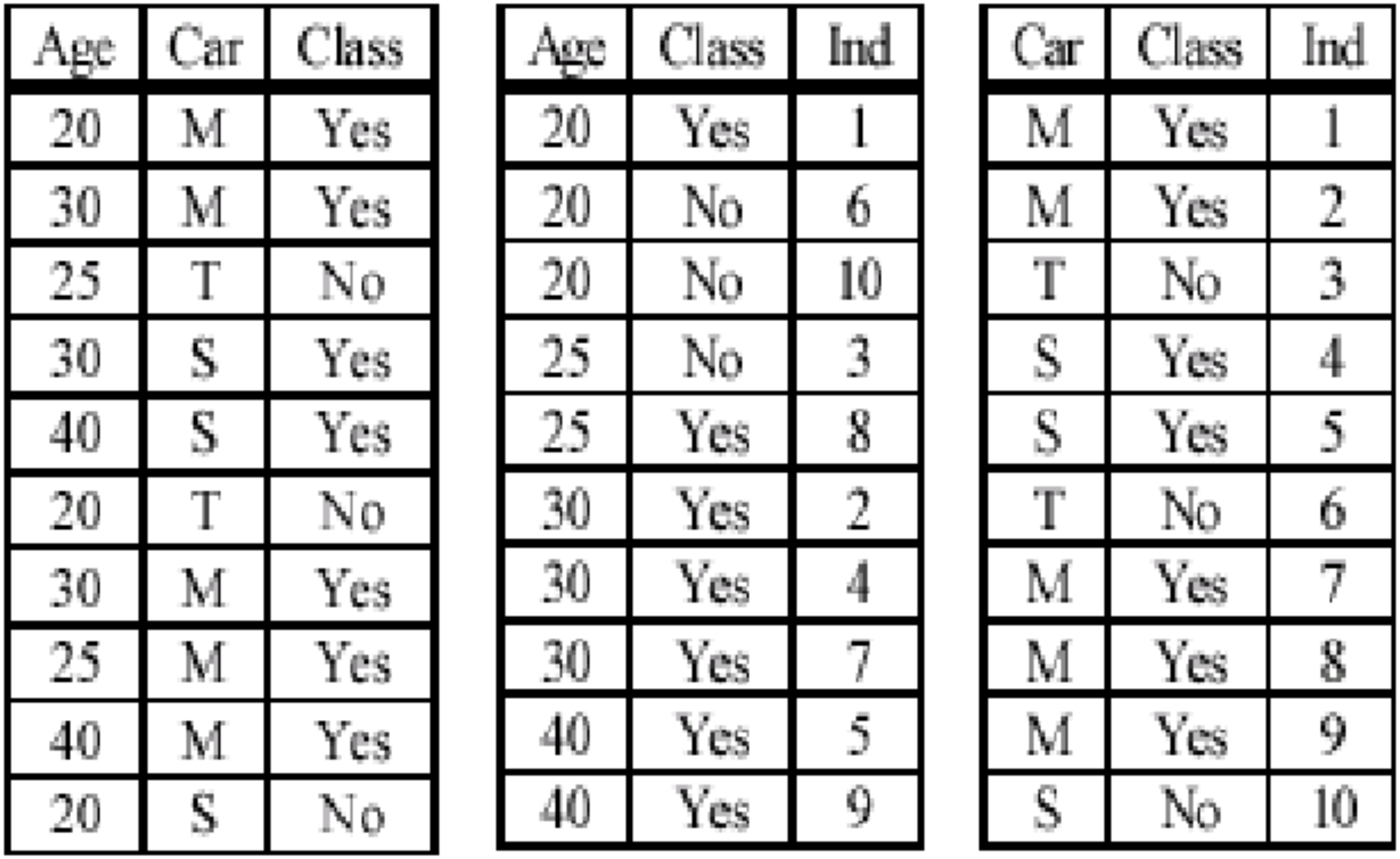
Scalable, Parallelizeable Decision Trees

- SLIQ: needs to **sort on each variable only once**
  - create a separate list of triplets (Attribute-value; Class; TID) for each attribute
    - (storage increases up to 3x)
  - keep location list (TID, class, tree-node-location) in main memory (bottleneck!)
  - sort each list initially
  - select best attribute - split-value combination
    - (using class histograms at each choice point)
    - Lists can be examined in parallel
  - Partition recorded by updating "tree-node-location" in location list
    - sequential partitioning of lists (hash join) preserves ordering!!
- PROBLEM: location list is memory-resident, shared data structure, and its size grows with # transactions
- SPRINT: refinement of SLIQ
  - uses a probe structure (hash table) to find L/R, instead of using location list - scalable!
  - Ref: Schafer, Agrawal, Mehta, Proc. VLDB, 1996.

SLIQ/SPRINT: Partitioning of Tables

- (from KDD2001 Tutorial by Gehrke & Loh)

Original | *Partitioned*

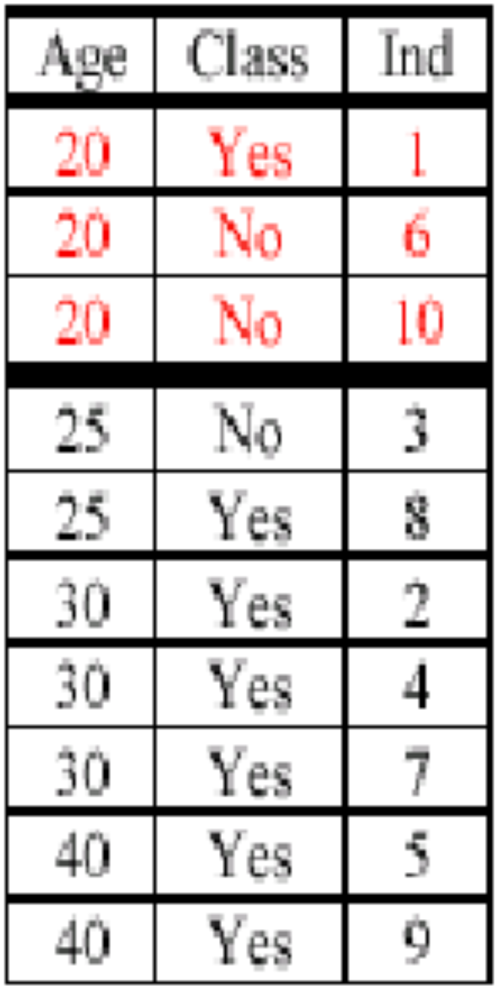


Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

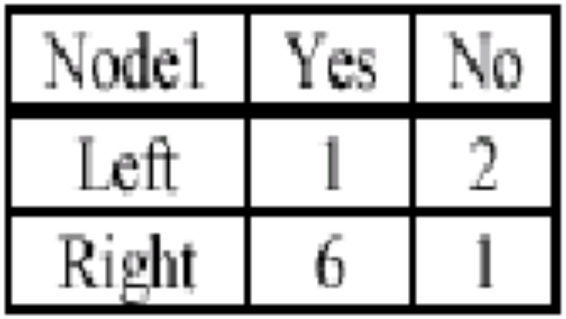
Age	Class	Ind
20	Yes	1
20	No	6
20	No	10
25	No	3
25	Yes	8
30	Yes	2
30	Yes	4
30	Yes	7
40	Yes	5
40	Yes	9

Car	Class	Ind
M	Yes	1
M	Yes	2
T	No	3
S	Yes	4
S	Yes	5
T	No	6
M	Yes	7
M	Yes	8
M	Yes	9
S	No	10

Evaluation of Splits



Age	Class	Ind
20	Yes	1
20	No	6
20	No	10
25	No	3
25	Yes	8
30	Yes	2
30	Yes	4
30	Yes	7
40	Yes	5
40	Yes	9



Node1	Yes	No
Left	1	2
Right	6	1

RainForest [Gehrke, Ramakrishnan, Ganti, VLDB 98]

- separate scalability aspects from algorithmic/ quality aspects
- split evaluation in all DT algorithms (allowing only univariate splits) are based solely on class counts for each attribute-value pair
  - AVC (attribute-value-classLabel) lists are sufficient statistics!!

Salary	Yes	No
30K	1	0
40K	3	0
60K	2	3

    If not too many attribute-values, can fit in main memory (independent of # of records)

  - For each tree level: scan D to construct AVC-lists in memory
  - find best split using AVC; partition D (one read-write scan); recurse
  - parallel implementation easy

Generative vs. Discriminative Approaches

- Generative: separately model class-conditional densities and priors
  (e.g. LDA)

- Discriminative: try to obtain class boundaries directly

  Through heuristic or through directly estimating posterior probabilities

  Just predict class label.

  (e.g. Decision Trees)

Pros and cons?