ECE7252
Statistical Learning for Signal Processing

Lecture 5: Overview on Supervised Learning

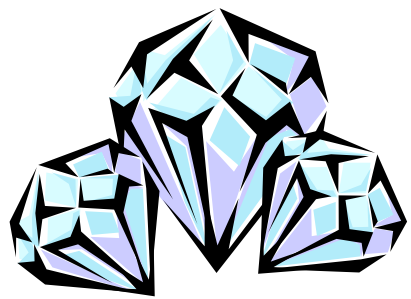
Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
chl@ece.gatech.edu

Outline of the Lecture

- Data mining: general concept
- Statistical learning: general concept
- Supervised learning: learning with a teacher
- Regression and classification problems
  - More detail in Chapters 3 and 4
- Model selection, feature selection, data selection, model complexity and generalization
- Statistical decision theory
  - Bayes decision rules, minimum error: more later

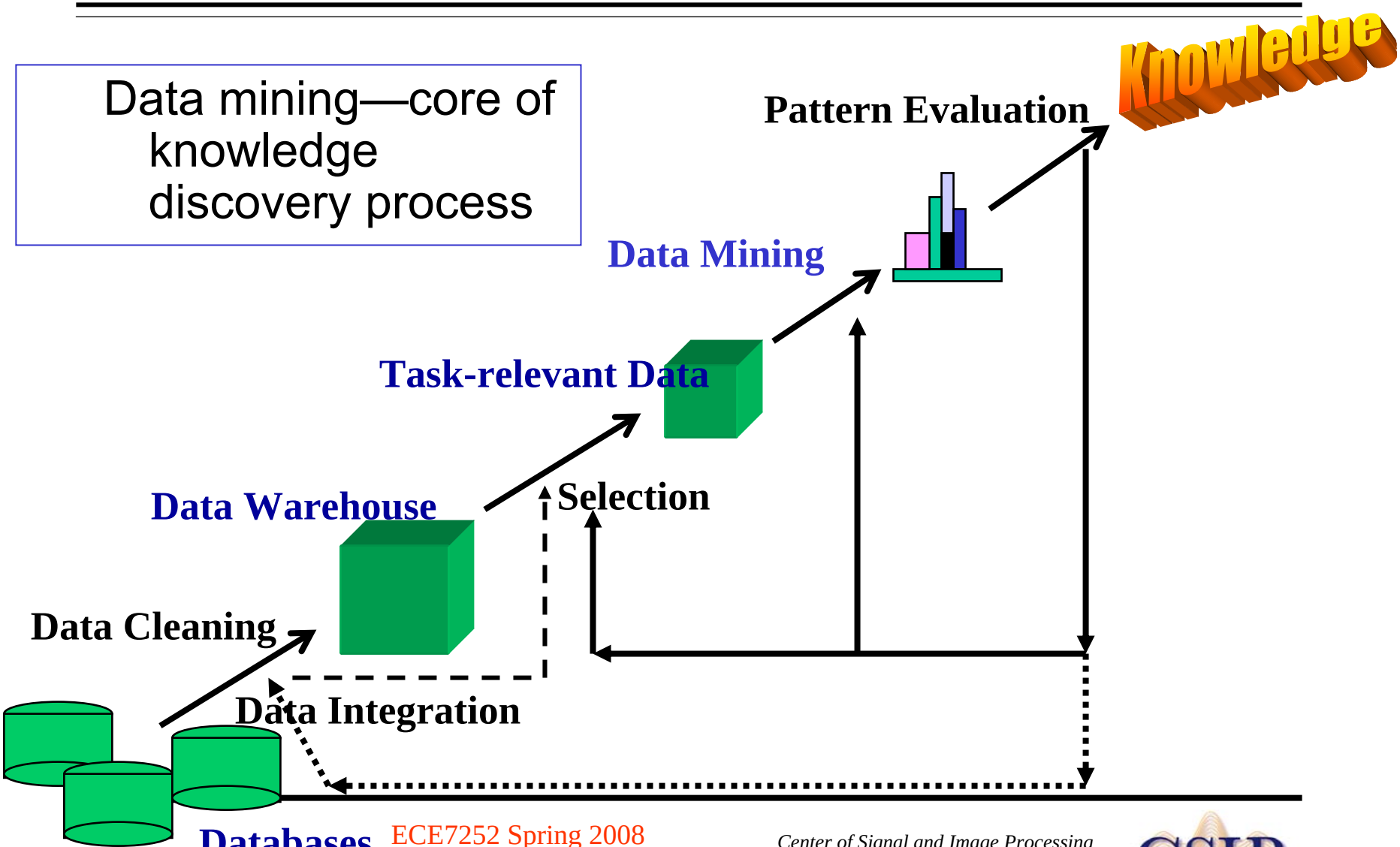
What Is Data Mining?

- Data mining (knowledge discovery from data)
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data
- Alternative names
  - Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.
- Watch out: Is everything "data mining"?
  - Simple search and query processing
  - (Deductive) expert systems

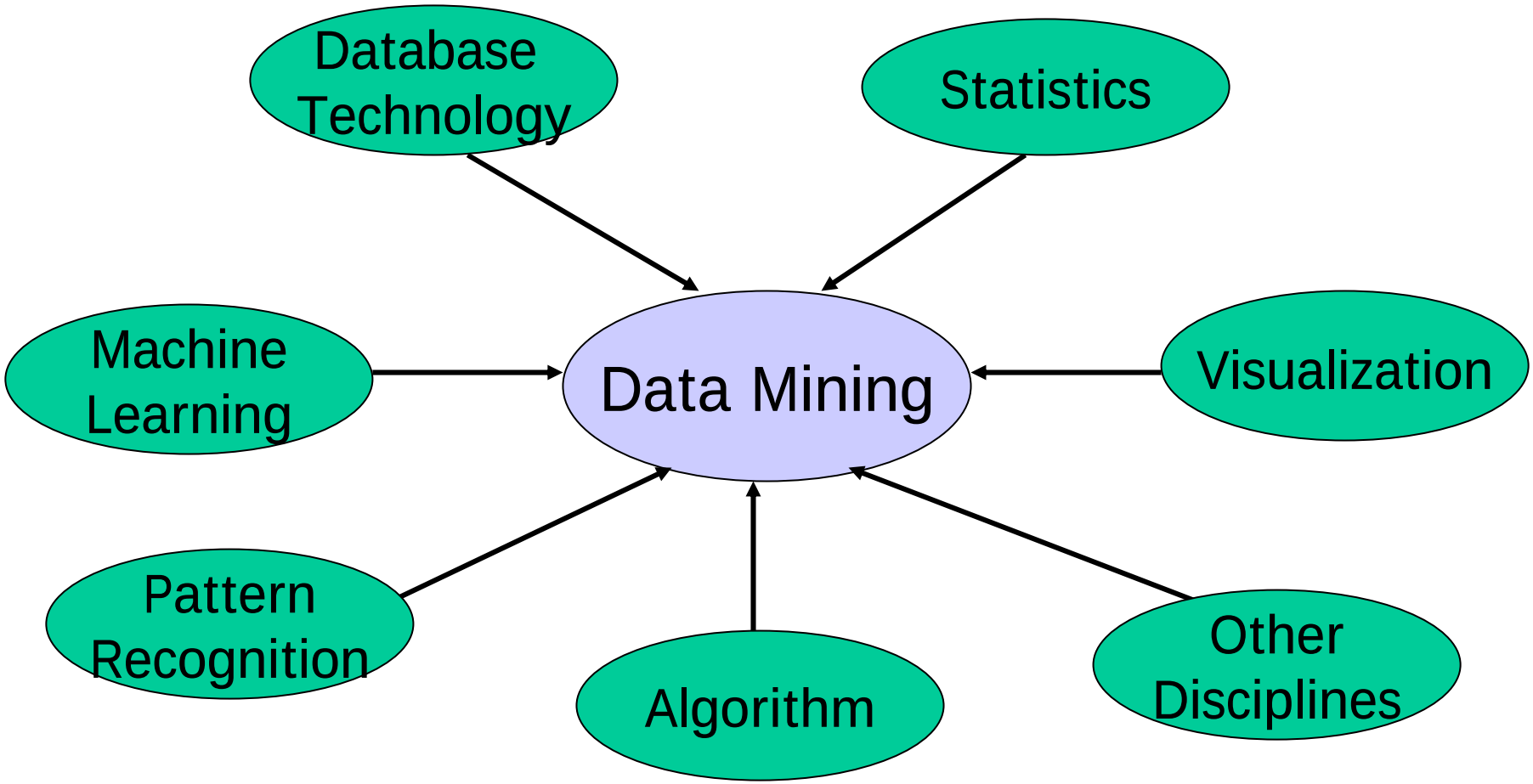


Knowledge Discovery (KDD) Process

Data mining—core of knowledge discovery process



Data Mining: Confluence of Multiple Disciplines



Statistical Learning

- Supervised learning: classification & prediction
  - – Given a set of training data, in which we observe the outcome and feature measurements for a set of objects
  - – Using this data we build a prediction model, or *learner*, which will enable us to predict the outcome for new unseen objects
  - – Example: (1) predict tomorrow's electricity consumption, from weather forecasts and calendar records (season, weekday, holiday); (2) Identify the numbers in a handwritten ZIP code, from a digitized image
- Unsupervised learning: association analysis & clustering
  - – Observe only the features but with no specific outcome
  - – Describe how the data are organized and clustered
  - – Example: Identify buying patterns that can be used to design sales promotions

Regression

- Prediction of quantitative output given one or more inputs: given examples $\{(\boldsymbol{x}, y), \ldots\}$, predict $y=f(\boldsymbol{x})$, $\boldsymbol{x}$ can be a vector
- Sample algorithms:
  - Linear and nonlinear regression
  - Artificial neural networks
  - Support vector machines
  - Auto-regression of time series (e.g. speech)
  - Decision trees

Classification

- Prediction of qualitative output given one or more inputs: given samples *{(**x**,C), …}*, predict *C=f(**x**)*, ***x*** can be a vector, and *C* is a class label": email spam: "is this a junk mail?"
- Sample algorithms:
  - Support vector machines (SVM)
  - Artificial neural networks (ANN)
  - Bayesian networks (BN)
  - K-nearest neighbor (KNN)
  - Linear discriminant function (LDF)
  - Hidden Markov model (HMM)
  - Decision trees (DT)

Association Rules

- Association rule: $X \rightarrow Y$
- Support $(X \rightarrow Y)$:

$$P(X,Y) = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers}\}}$$

- Confidence $(X \rightarrow Y)$:

$$P(Y \mid X) = \frac{P(X,Y)}{P(X)} = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers who bought } X\}}$$

Prediction by Learning from Data

Assume that we have a data set

x_{11}	x_{21}	.	.	.	x_{p1}	y_1
x_{12}	x_{22}	.	.	.	x_{p2}	y_2
.	.				.	.
.	.				.	.
.	.				.	.
x_{1n}	x_{2n}	.	.	.	x_{pn}	y_n

which shows the outcome (response) y **for a set of investigated objects with features** $x_1, \ldots, x_p$

Prediction by learning from data implies that

$$\hat{y} = \hat{f}(x_1, x_2, \ldots, x_p)$$

that can be used to foresee the outcome for new objects (with known or observed features)

Major Quantitative Prediction Models

- **Linear or nonlinear regression models with i.i.d. error terms**

$$y_t = f(x_{1t}, x_{2t}, \ldots, x_{pt}) + \varepsilon_t$$

- **Time series regression models with stochastic noise**

$$y_t = f(x_{1t}, x_{2t}, \ldots, x_{pt}) + N_t$$

- **Transfer function models**

$$y_t = f(x_{1t}, x_{1,t-1}, \ldots, x_{2t}, x_{2,t-1}, \ldots, x_{pt}, x_{p,t-1}, \ldots) + N_t$$

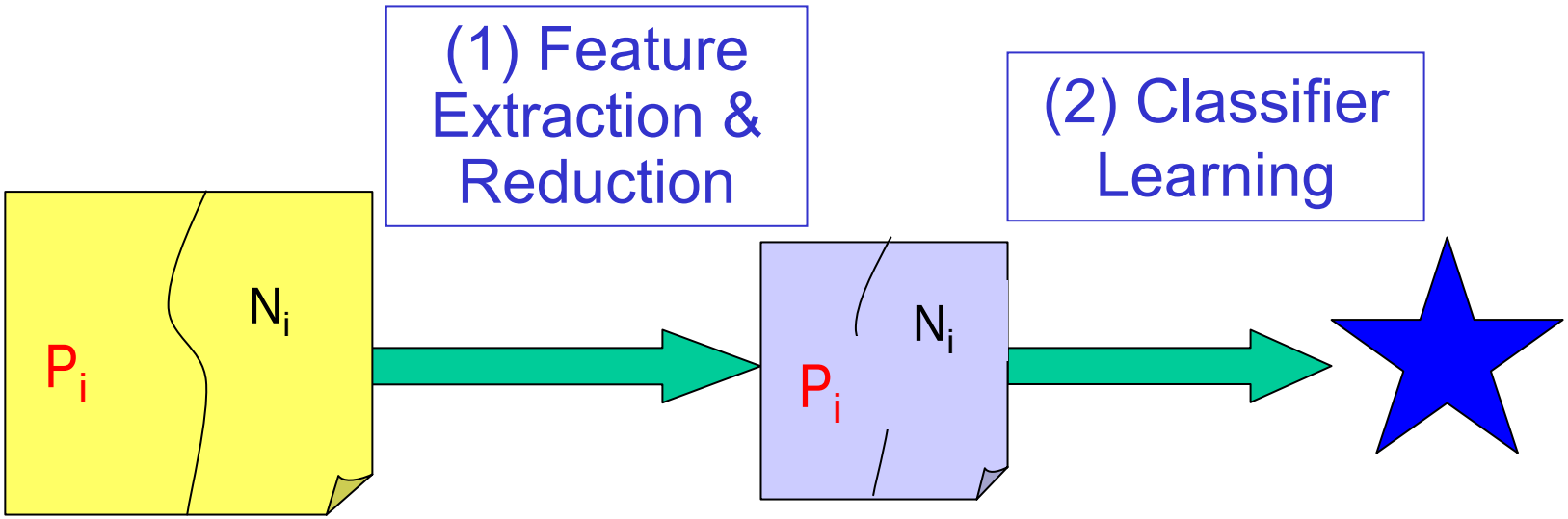
Supervised & Unsupervised Learning

- Supervised learning
  - A teacher provides a category label or cost for each pattern in the training set (i.e., ground truth based on experts' knowledge)
  - Can we trust such ground truth? Inconsistency?
- Unsupervised learning
  - The system forms clusters or "natural groupings" of the input patterns
- Semi-supervised learning
  - Use both labeled and un-labeled patterns to reduce the labeling cost

Algorithms for Classification and Regression

- Linear perceptron, least squares
- LDA/FDA (linear/Fisher discriminant analysis): simple linear cuts, kernel non-linear generalizations
- SVM (support vector machine): optimal, wide margin linear cuts, kernel non-linear generalizations
- Decision trees: logical rules
- KNN (*k*-nearest N\neighbors): simple non-parametric
- Artificial neural networks (ANN): general non-linear models, adaptation ability, “artificial brain”

Topic Categorization: Training Classifiers

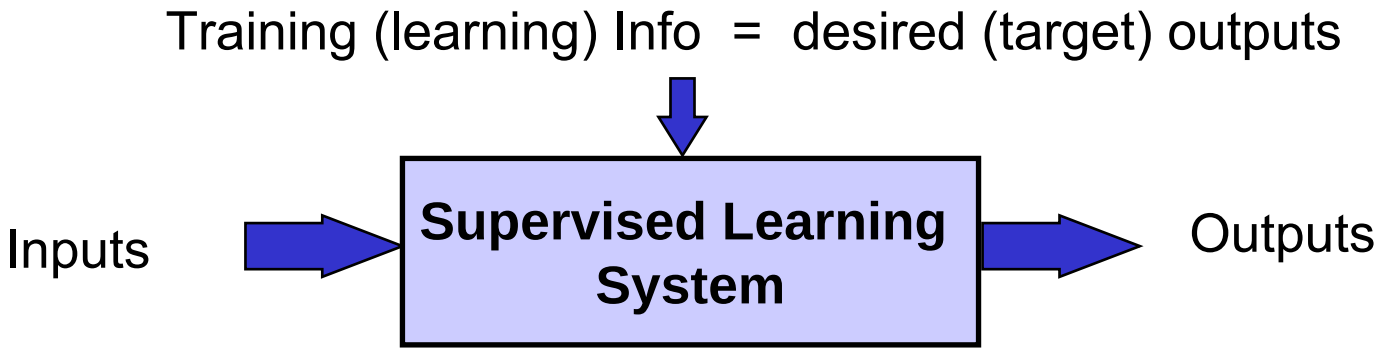


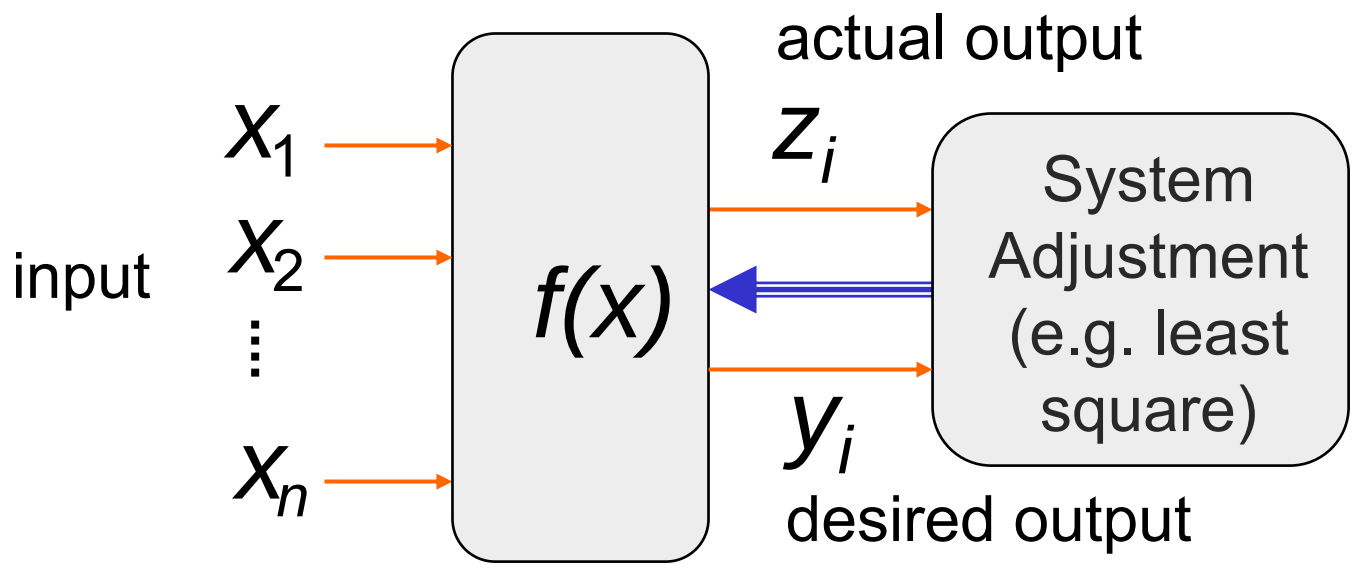
Training set for each category C_i , i= 1,…,m. (Positive + Negative : supervision information)

Documents in new feature space

Classifier T_i for category C_i

Supervised Learning





Three Phases in Supervised Learning

- Collecting labeled training data
  - Examples with label assignment
  - Labels depend on application needs
- Learning classification models
  - Appropriate model and representation of the problem need to be selected in terms of attributes, distance (similarity) measure and classifier type
  - Adaptive parameters in the model need to optimized to provide correct classification of training examples (e.g. minimizing the number of misclassified training vectors)
- Validation
  - Cross-validation, independent control sets and other measure of "real" accuracy and generalization should be used to assess the success of the model *(finding trade off between accuracy and generalization is not trivial)*

Data Correlations and Fingerprints

- Instead of deciphering models generating the data, which is often hard to do, one may simply try to find correlations between inputs and outputs (knowledge-ignorant). If measurements on certain attributes correlate with the underlying physical processes governing the data, one can use such attributes as indicators of some “hidden” states and to make predictions for new cases

- Considering for example the levels of the low density lipoprotein (LDL, “bad” cholesterol) and high density lipoprotein (HDL, “good” cholesterol) particles in the blood as indicators (*fingerprints*) of the atherosclerosis, can we use them to predict subjects’ behaviors?

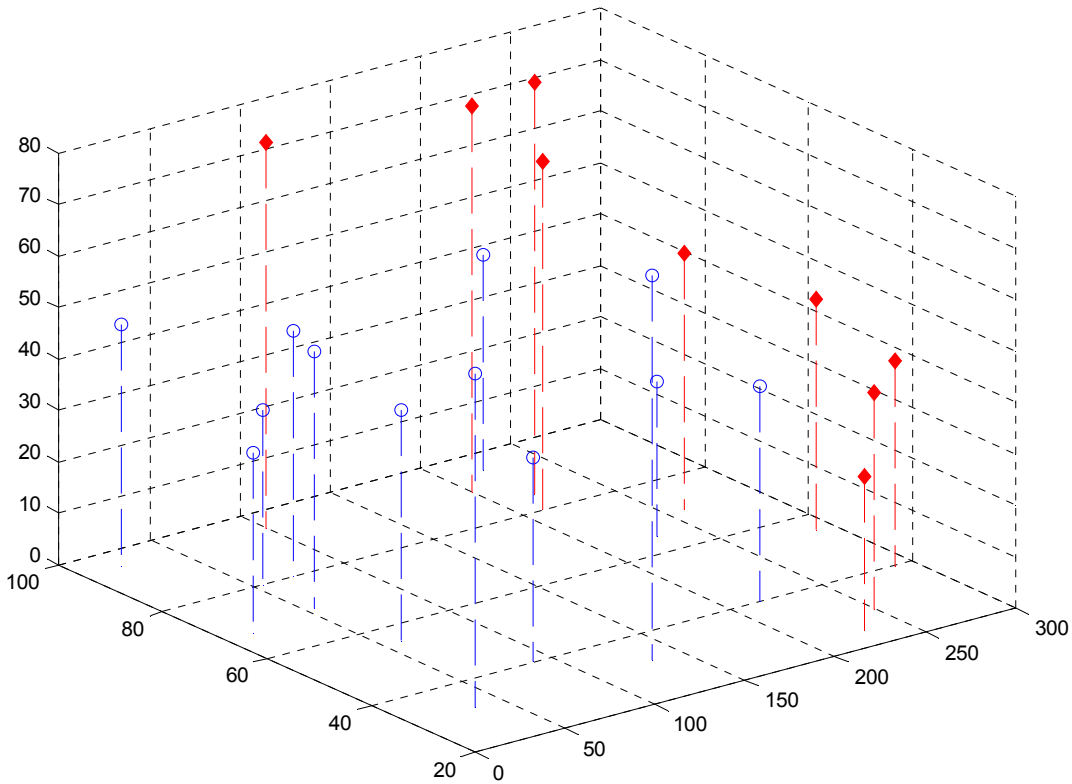
Training Set: LDL Example

A set of objects (here patients) $\mathbf{x}_i$, $i=1, \ldots, N$ is given. For each patient a set of features (attributes and the corresponding measurements on these attributes) are given too. Finally, for each patient we are given the class C_k , $k=1, \ldots, K$, he/she belongs to.

Age	LDL	HDL	Sex	Class
41	230	60	F	healthy (0)
32	120	50	M	stroke within 5 years (1)
45	90	70	M	heart attack within 5 years (1)

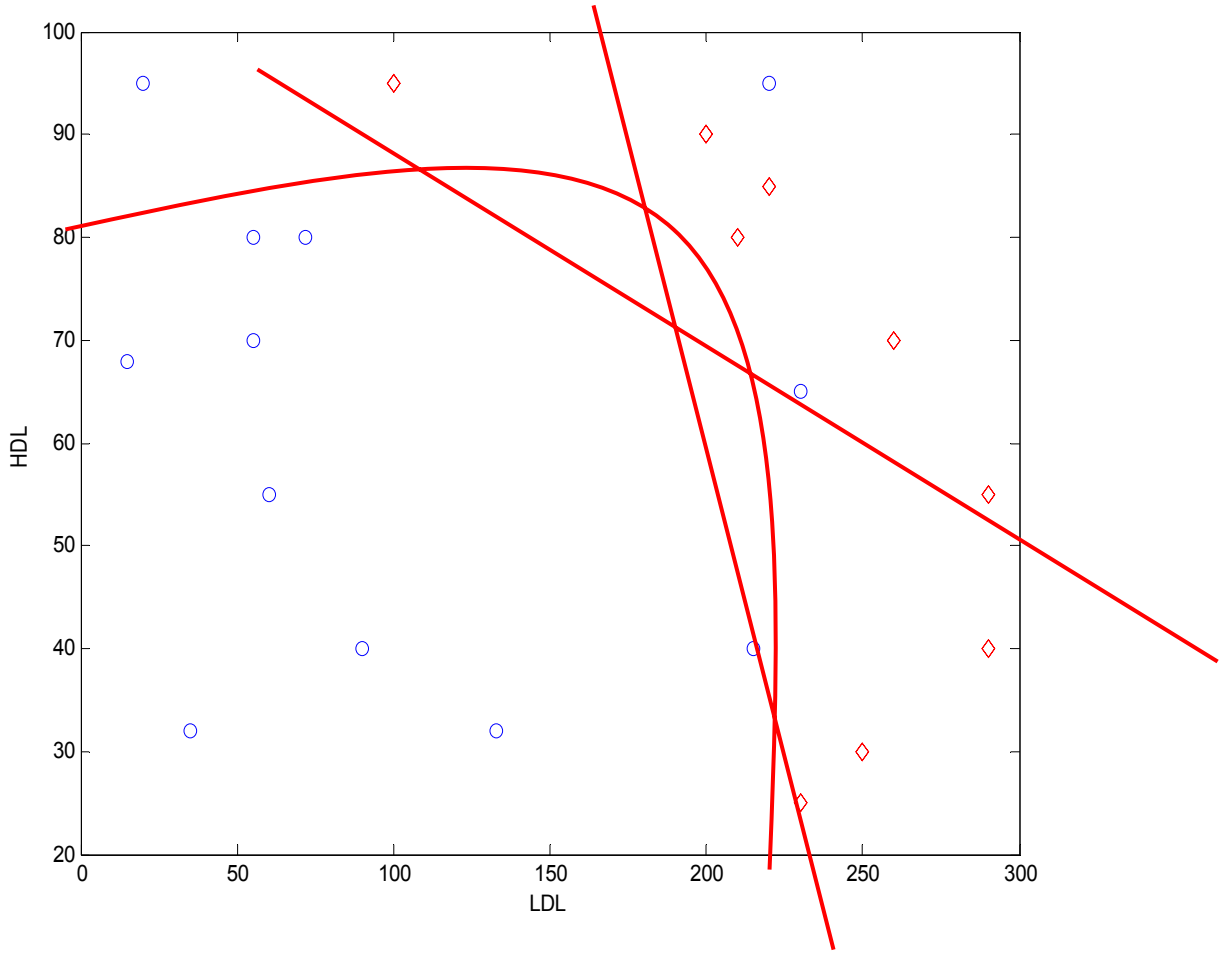
$$\{ \boldsymbol{x}_i , C_k \} \qquad i=1, \ldots, N$$

Graphical Illustration: LDL Example

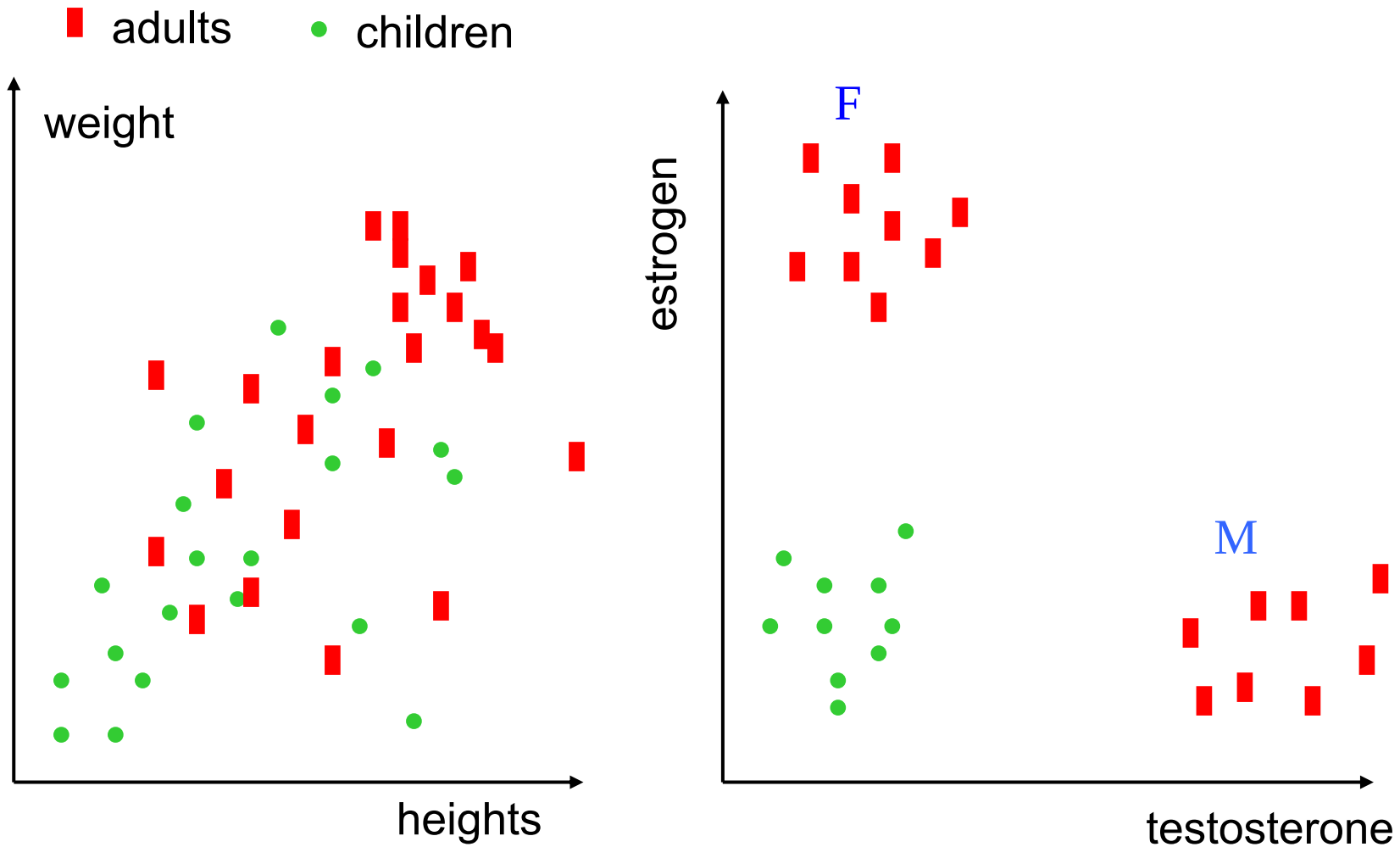


Blue; heart attack or stroke within 5 years from the exam: red (simulated data); *x* – LDL; *y* - HDL; *z* – age (see study by Westendorp et. al., Arch Intern Med. 2003, 163(13):1549.

LDL Example: 2D Projection



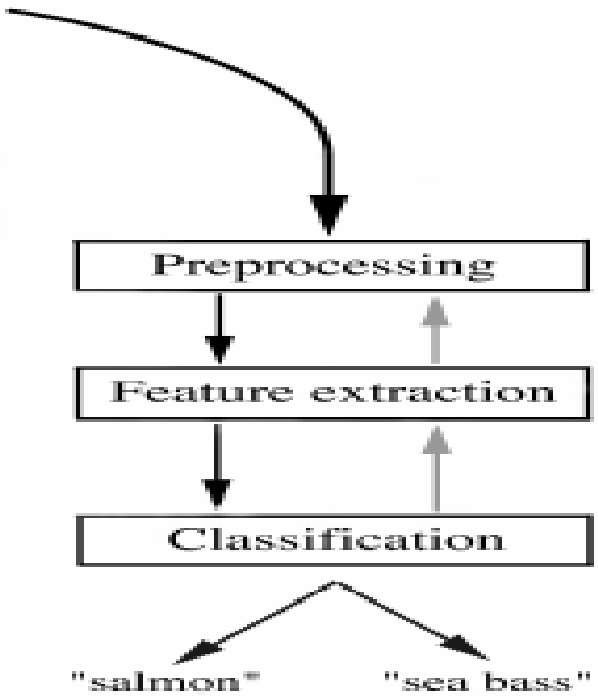
Feature Selection: Discriminative Features



An Example of Fish Classification

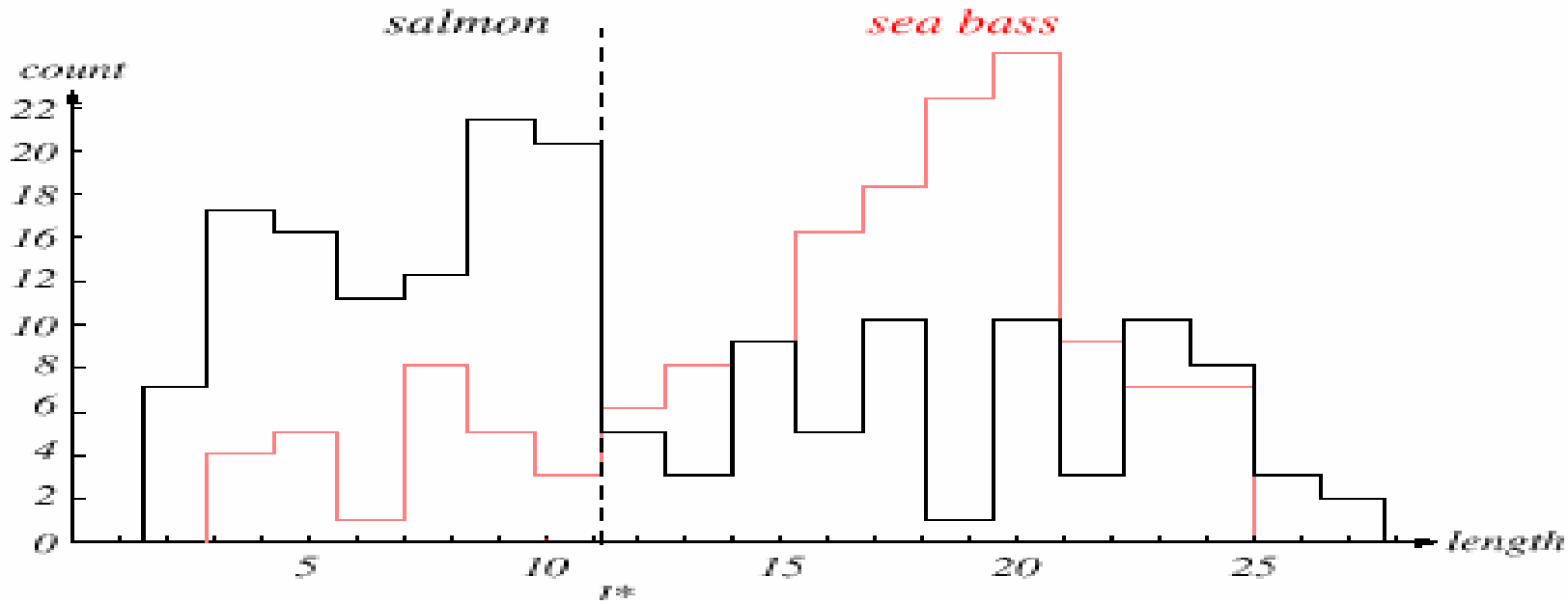
- Problem Analysis: sort incoming fish on a conveyor according to species (sea bass or salmon)
  - Set up a camera and take some sample images to extract features (using optical sensing)
    - Length
    - Lightness
    - Width
    - Number and shape of fins
    - Position of the mouth, etc…
    - This is the set of all suggested features to explore for use in our classifier!

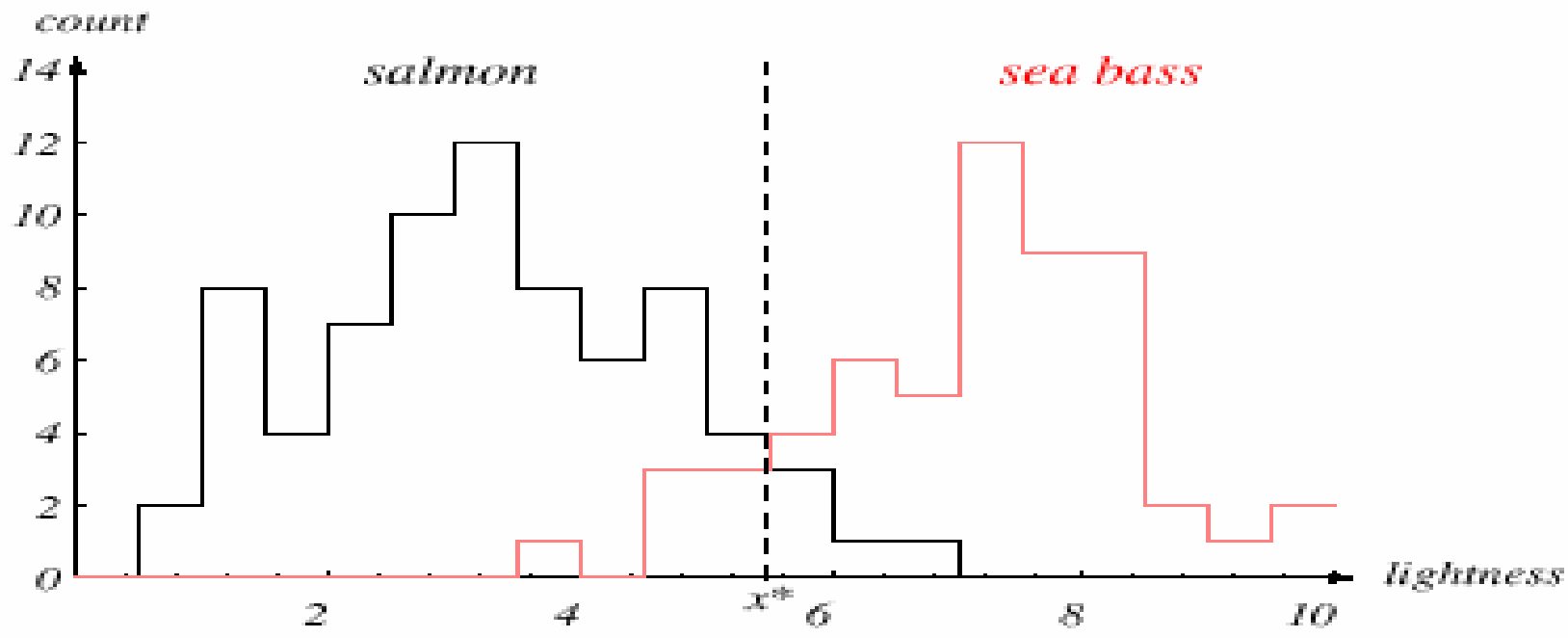
System Implementation



Processing Steps

- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
  - To extract one fish for the next step
- Feature extraction
  - Measuring certain features of the fish to be classified
  - Is one of the most critical steps in the pattern recognition system design
- Classification
  - Select the length of the fish as a possible feature for discrimination

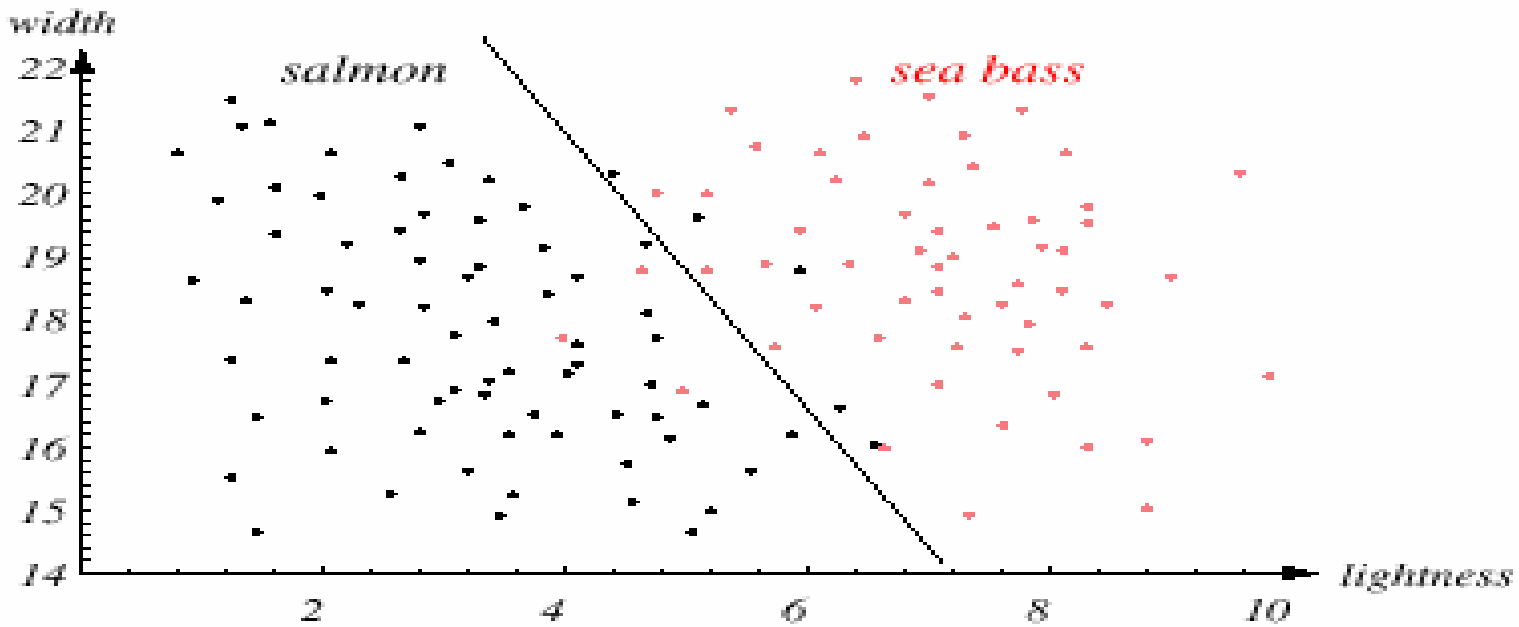

salmon
sea bass
count
22
20
18
16
12
10
8
6
4
2
0
5
10
15
20
25
length
x*


count
14
12
10
8
6
4
2
0
salmon
sea bass
2
4
x*
6
8
10
lightness

Features and Scatter Plot

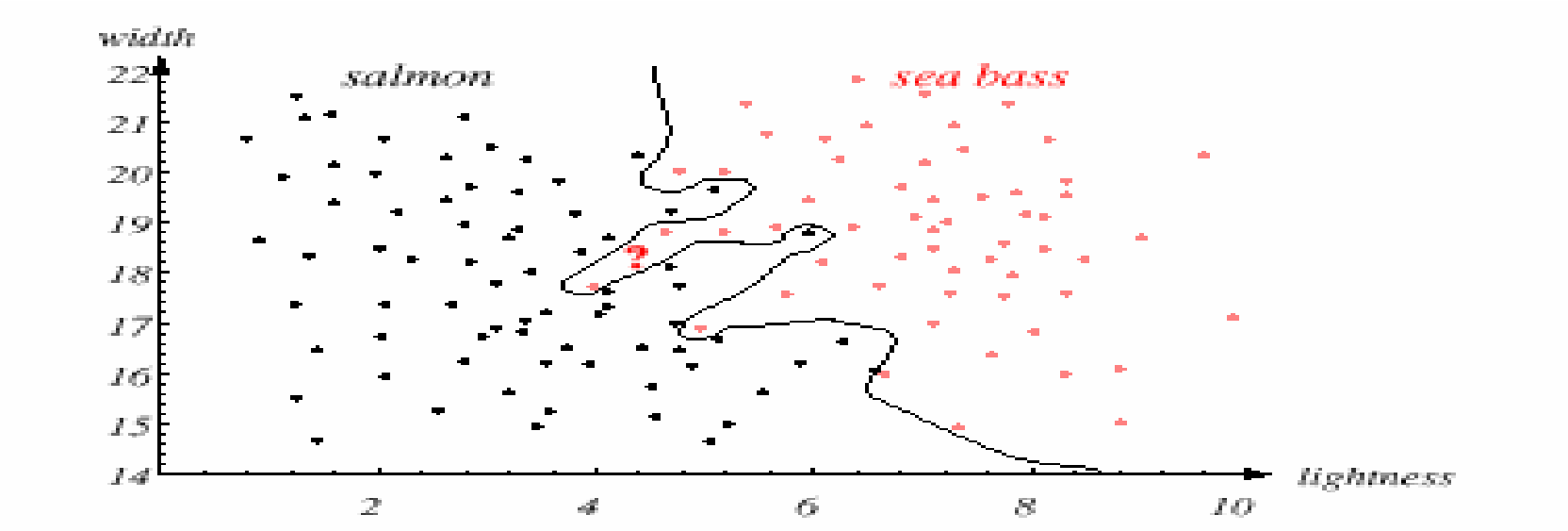
Fish → $x^T = [x_1, x_2]$

Lightness **Width**



Alternative Features and Models

- May add other features that are not correlated with the ones we already have: but not to reduce the performance by adding such “noisy features”
- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following:



Similarity Measures

Consider a set of N objects represented as vectors in a certain p-dimansional feature space (e.g. measurments on p attributes, such as expression levels for p genes) $\boldsymbol{x}_i=[x_{i1}, x_{i2}, \ldots, x_{im}, \ldots, x_{ip}]^T$. For quantitative atributes (variables), a commonly used measure of dissimilarity is the *squared distance*:

$$d_2(\boldsymbol{x}_i, \boldsymbol{x}_j)=\sum_m (x_{im}-x_{jm})^2 \quad .$$

In case of gene expression data, the correlation coefficient is often used to measure similarity between expression profiles across genes or samples:

$$cc(\boldsymbol{x}_i, \boldsymbol{x}_j)=\frac{\sum_m (x_{im}-\langle x_i\rangle)(x_{jm}-\langle x_j\rangle)}{\sqrt{\sum_m (x_{im}-\langle x_i\rangle)^2 \sum_m (x_{jm}-\langle x_j\rangle)^2}} \quad ,$$

where $\langle x_i\rangle=\sum_m x_{im}/p$.

Performance Analysis

- Need mechanisms and metrics to compare the effectiveness of various algorithms
- False Rejection: *the fraction of false positives*
- False Alarm: *the fraction of false negatives*
- Recall: *the fraction of relevant cases retrieved*
- Precision: *the fraction of retrieved cases that were relevant*
- Error Rate: *the fraction of wrong classification*
- F-Value*: combination of recall and precision*
- Usually, increased performance in one metric results in decreased performance in the other

Local Performance Measures

- Local Performance Measures for Category C_i

Category C_i		Manual Labels	
		C+	C-
Classifier Judgments	C+	TP_i	FP_i
	C-	FN_i	TN_i

$$\mathrm{Pr}_i = \frac{TP_i}{TP_i + FP_i}$$

$$\mathrm{Re}_i = \frac{TP_i}{TP_i + FN_i}$$

$$F_{1i} = \frac{2\,\mathrm{Re}_i\,\mathrm{Pr}_i}{\mathrm{Re}_i + \mathrm{Pr}_i}$$

- *Precision*, *Recall* and *F1*

Global Performance Measures

- Global Performance Measures

Category set $C = \{C_1,\ldots,C_m\}$		Manual Labels	
		C+	C-
Classifier Judgments	C+	$TP = \sum_{i=1}^{m} \mathrm{TP}_i$	$FP = \sum_{i=1}^{m} \mathrm{FP}_i$
	C-	$FN = \sum_{i=1}^{m} \mathrm{FN}_i$	$TN = \sum_{i=1}^{m} \mathrm{TN}_i$

Summary Performance Measures

- Micro-averaging

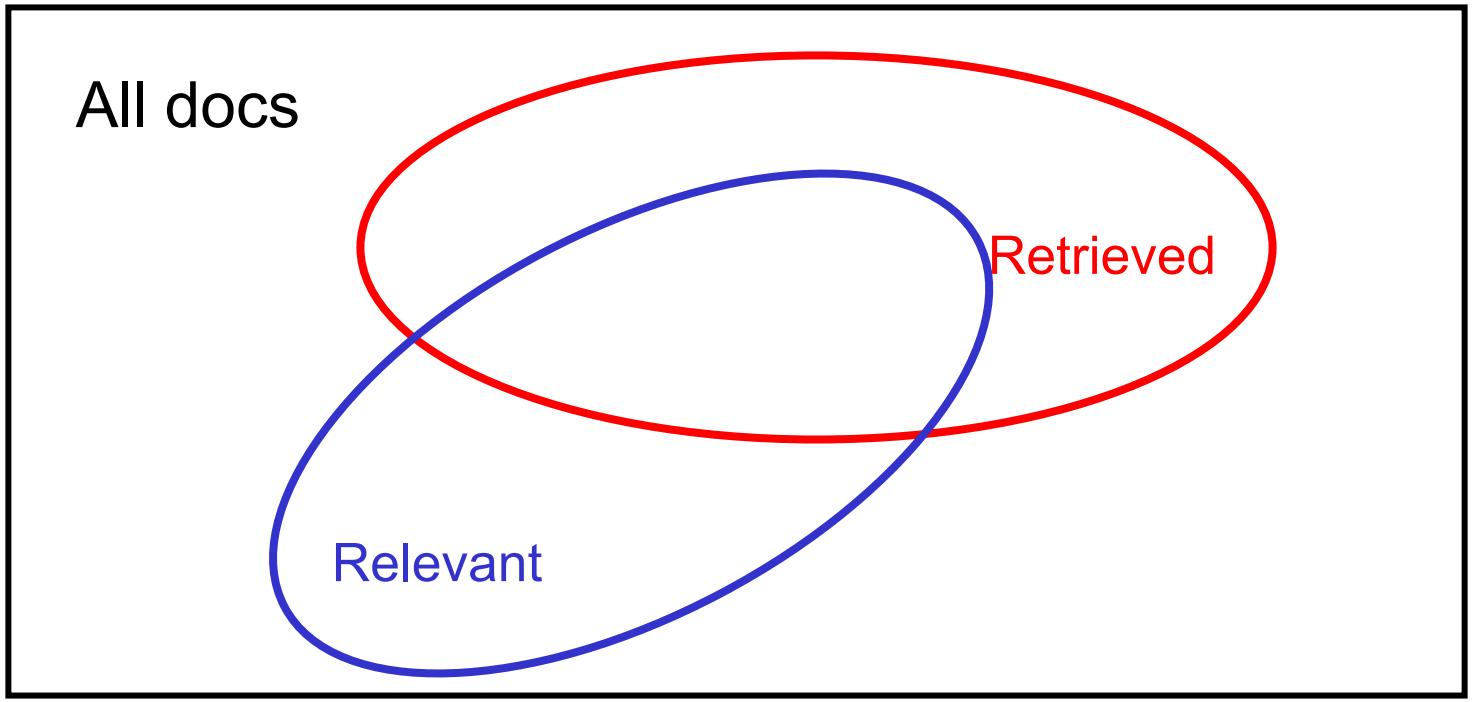
$$\Pr^{u} = \frac{TP}{TP + FP}, \quad \mathrm{Re}^{u} = \frac{TP}{TP + FN}, \quad F_1^{\mu} = \frac{2TP}{FP + FN + 2TP}.$$

- Macro-averaging

$$\Pr^{M} = \frac{\sum_{i=1}^{m} \Pr_i}{m}, \quad \mathrm{Re}^{M} = \frac{\sum_{i=1}^{m} \mathrm{Re}_i}{m},$$

$$F_1^{M} = \frac{2\,\mathrm{Re}^{M}\,\Pr^{M}}{\mathrm{Re}^{M} + \Pr^{M}} = \frac{2\sum_{i=1}^{m} \mathrm{Re}_i \sum_{i=1}^{m} \Pr_i}{m(\sum_{i=1}^{m} \Pr_i + \sum_{i=1}^{m} \mathrm{Re}_i)}.$$

Precision vs. Recall

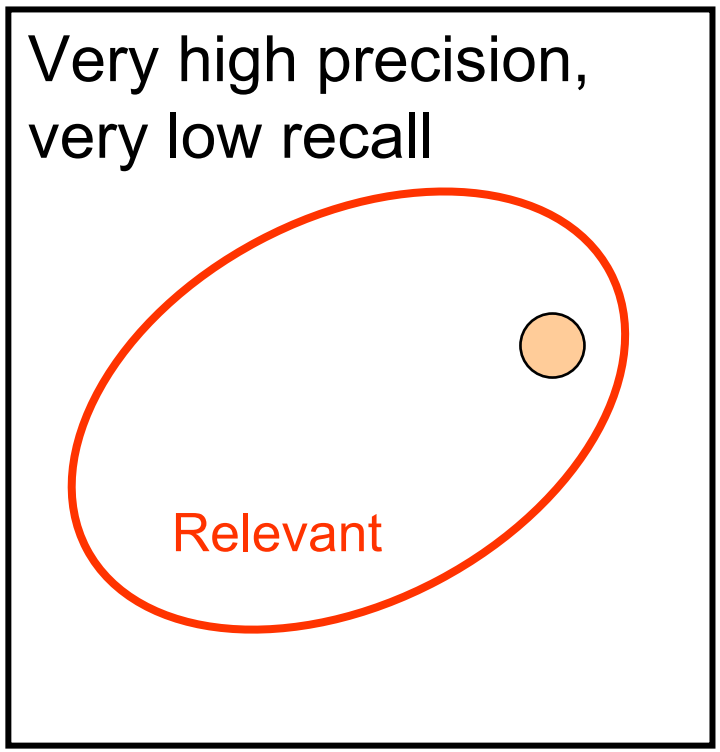


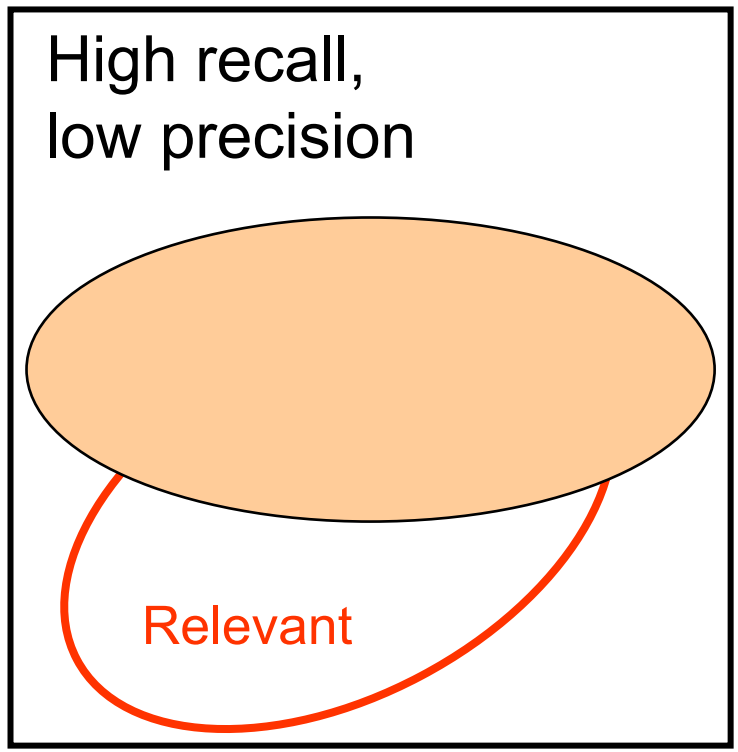
$$\text{Recall} = \frac{|\text{RelRetrieved}|}{|\text{Rel in Collection}|}$$

$$\text{Precision} = \frac{|\text{RelRetrieved}|}{|\text{Retrieved}|}$$

Why Precision and Recall?

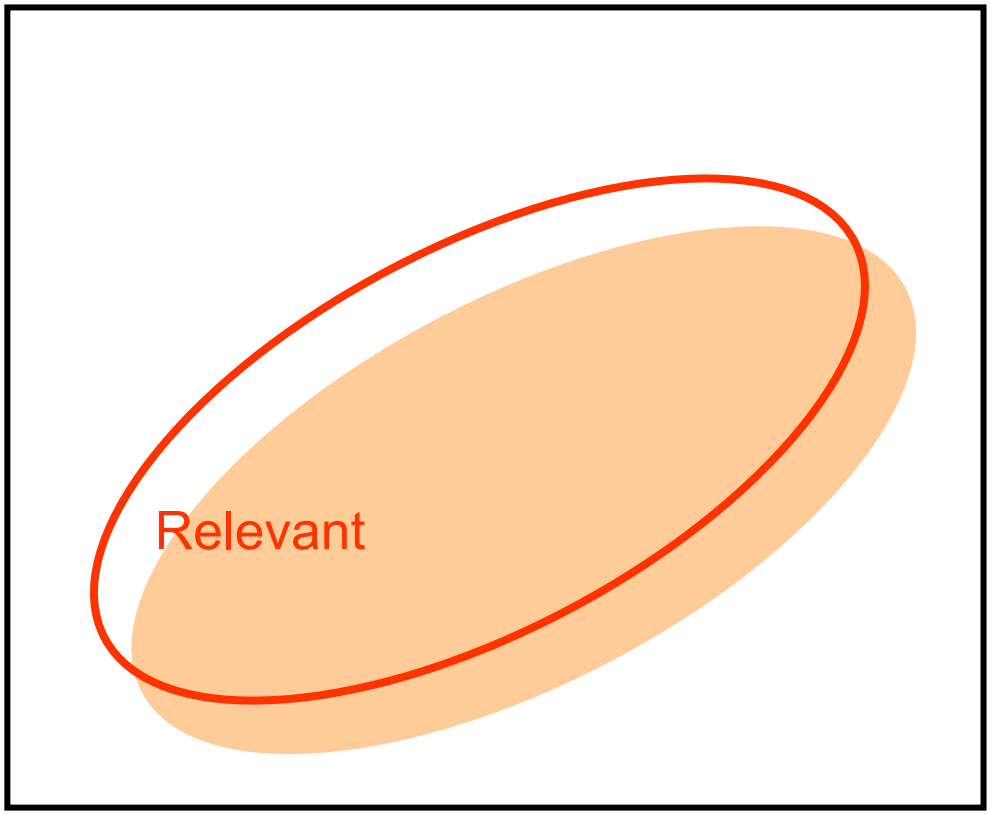
Get as much good stuff while at the same time getting as little junk as possible



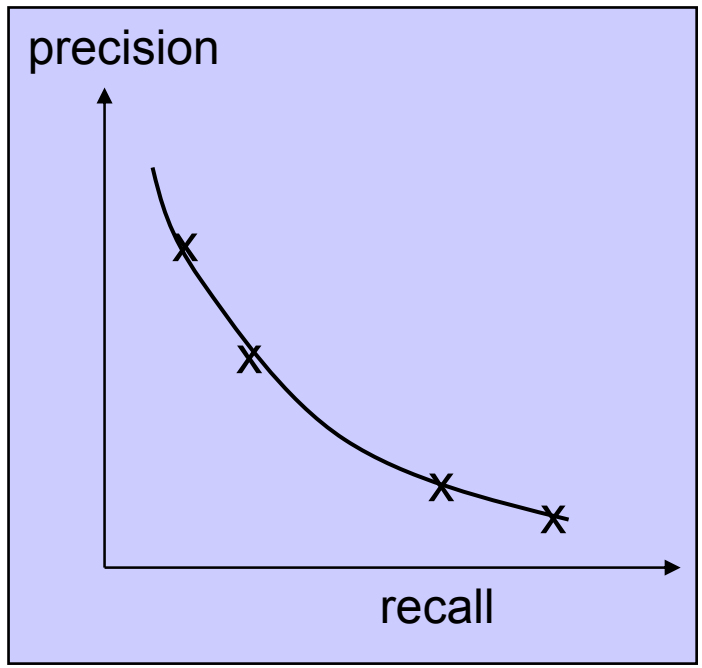


Retrieved vs. Relevant Documents

High precision, high recall (at last!)



Conflicting goals: similar to ROC



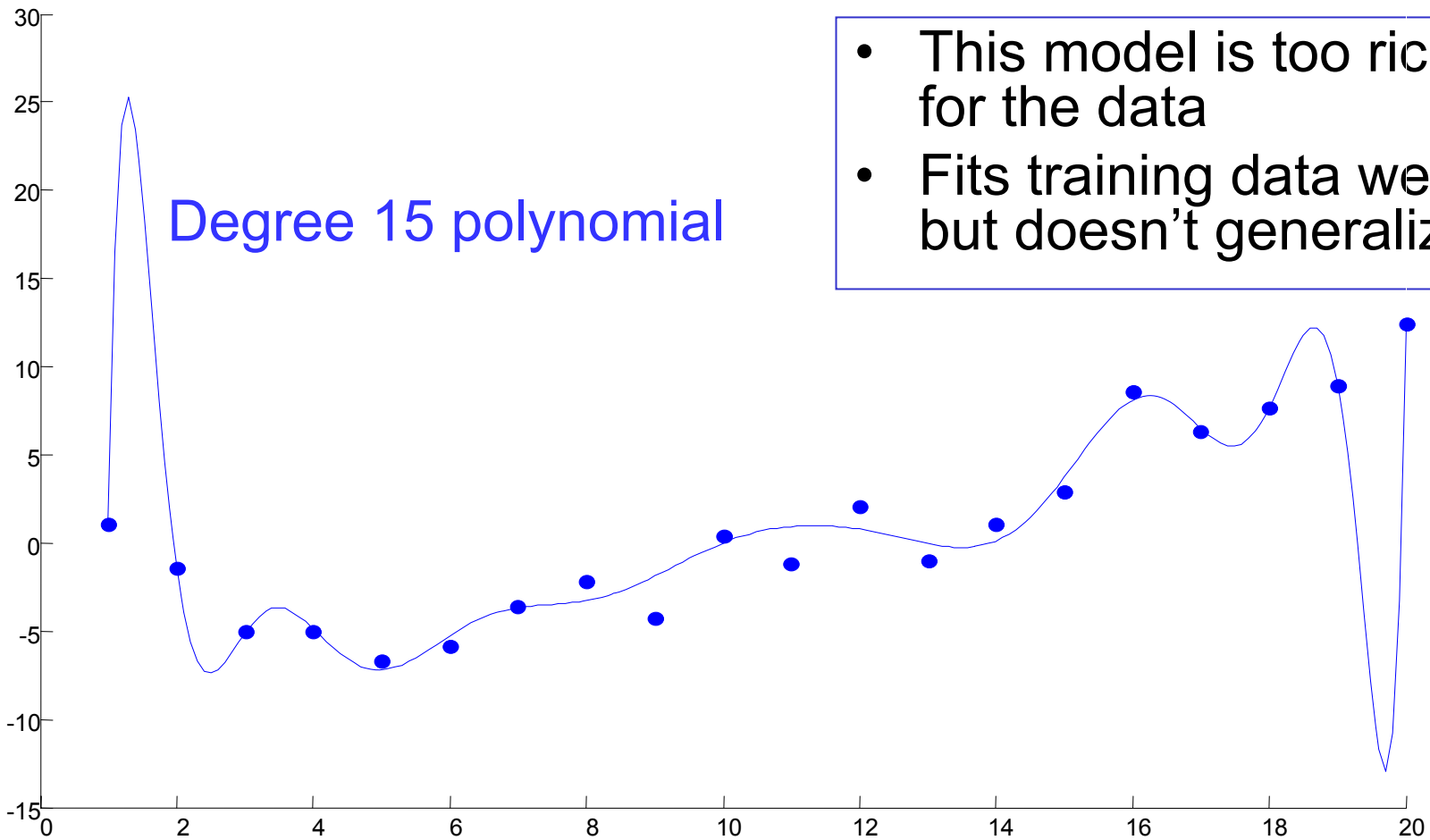
Learning and Data Selection

- Divide the data into two sets
  - Training: use the training data to fit all potential models
  - Test (Evaluation): use the test data to evaluate and validate the trained models
- Data-rich environment (even better): divide the data into three sets
  - Training: use the training data to fit all potential models
  - Validation: use the validation data to select model
  - Test: use the test data to obtain an unbiased estimate of the selected models

Model Selection & Generalization

- Learning is an ill-posed problem; data is not sufficient to find a unique solution
- The need for inductive bias: assumptions about the hypothesized model, $\mathcal{H}$, from true C
- Generalization: How well a model performs on new data?
- Overfitting: $\mathcal{H}$ more complex than true C or f
- Underfitting: $\mathcal{H}$ less complex than true C or f

Overfitting Example 1

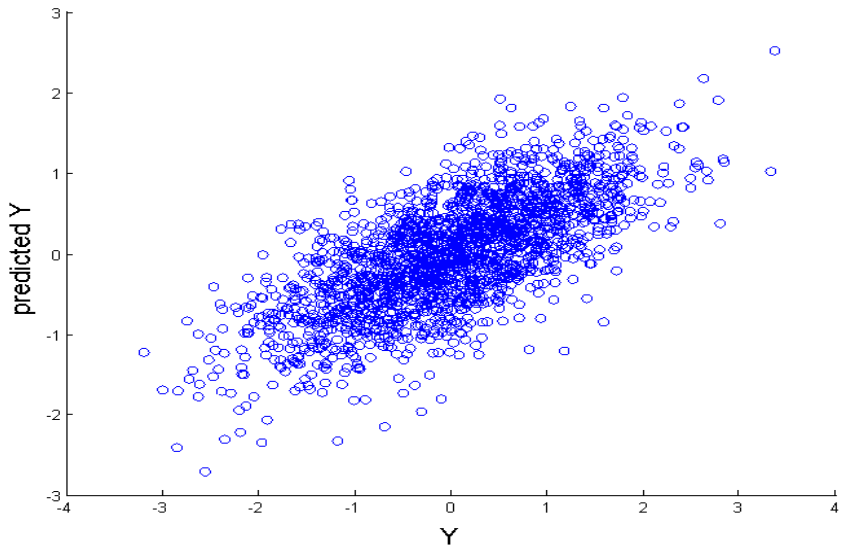


- This model is too rich for the data
- Fits training data well, but doesn't generalize

(thanks to Romain for the slide)

Overfitting Example 2

- Generate 2000 $\boldsymbol{x}_i \in \mathbb{R}^{1000}$, $\boldsymbol{x}_i \sim \mathcal{N}(0, I)$ i.i.d.
- Generate 2000 $y_i \in \mathbb{R}$, $y_i \sim \mathcal{N}(0, 1)$ i.i.d. *completely independent of the* $\boldsymbol{x}_i$*'s*
  - We shouldn't be able to predict y at *all* from $\boldsymbol{x}$
- Find $\hat{\boldsymbol{w}} = \operatorname{argmin}_{\boldsymbol{w}} L(\boldsymbol{w})$
- Use this to predict y_i for each $\boldsymbol{x}_i$ by $\hat{y}_i = \hat{\boldsymbol{w}}^\top \boldsymbol{x}_i$



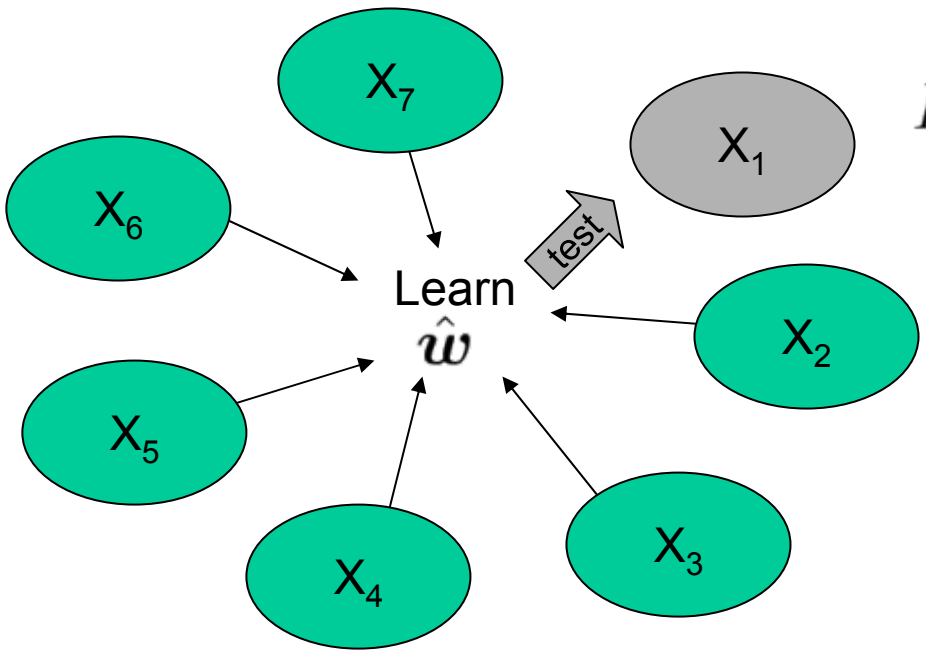
It really looks like we've found a relationship between $\boldsymbol{x}$ and y ! But no such relationship exists, so $\hat{\boldsymbol{w}}$ will do no better than random on new data.

Cross-Validation

- To estimate generalization error, we need data unseen during training. We split the data as
  - Training set (50%)
  - Validation set (25%)
  - Test (publication) set (25%)
- Resampling when there is few data

K-fold Cross Validation

- A technique for estimating test error
- Uses *all* of the data to validate
- Divide data into K groups $\{X_1, X_2, \ldots, X_K\}$.
- Use each group as a validation set, then average all validation errors



$$L_1 = \sum_{(\boldsymbol{x},y)\in X_1} (y - \hat{\boldsymbol{w}}^\top \boldsymbol{x})$$

$$CV(s) = \frac{1}{K}\sum_{i=1}^{K} L_i$$

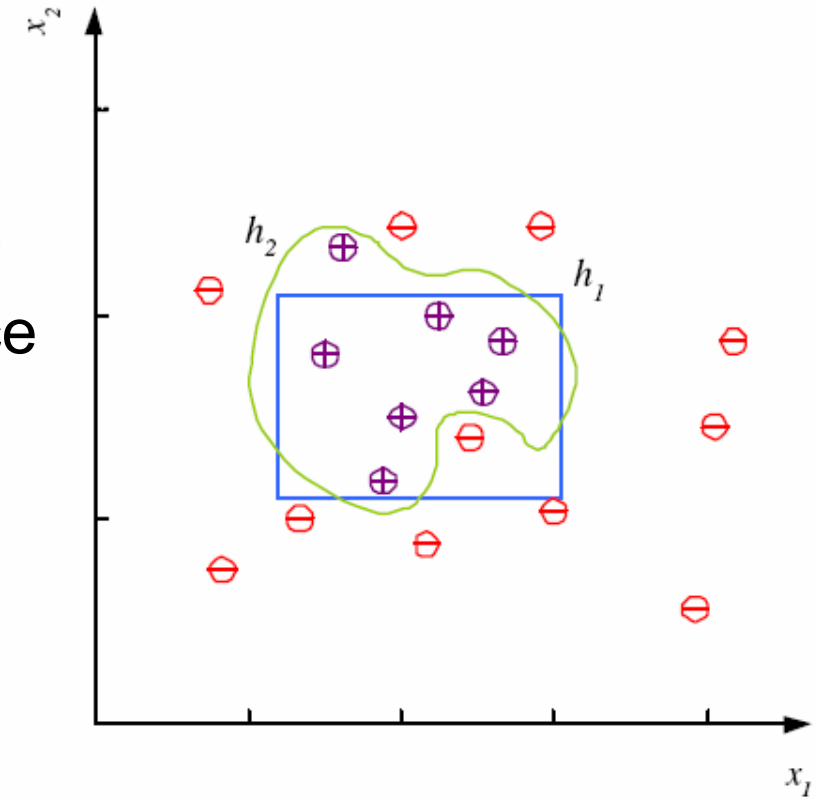
Triple Trade-Off

- There is a trade-off between three factors (Dietterich, 2003):
  - Complexity of $\mathcal{H}$: $c(\mathcal{H})$,
  - Training set size: N,
  - Generalization error, E, on new data
- As $N\uparrow$, $E\downarrow$
- As $c(\mathcal{H})\uparrow$, first $E\downarrow$ and then $E\uparrow$

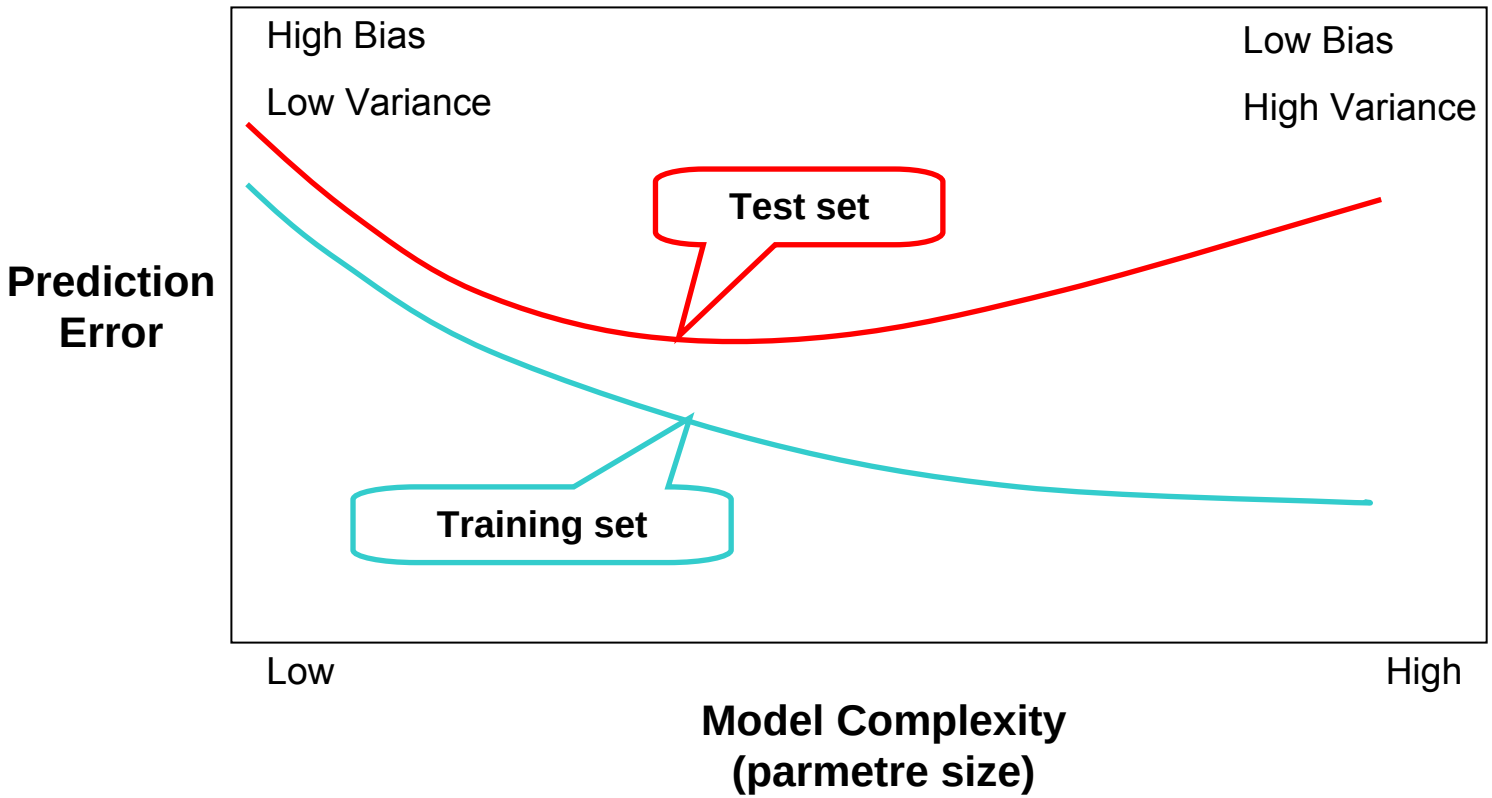
Noise and Model Complexity

Use the less complex model because

- Simpler to use :(lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance - Occam's razor)

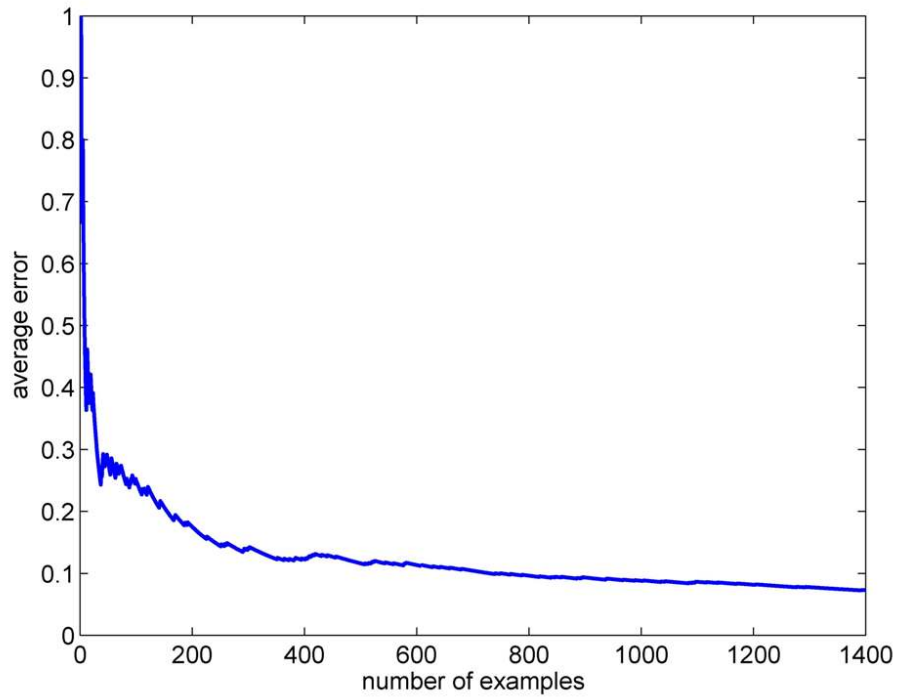


Bias, Variance and Generalization



Performance & Training Set Size

- Evaluate the performance of the algorithm by measuring average classification error (over the entire data set) as a function of the number of examples seen so far:



Statistical Decision Theory

• Given *P(X,W),* the joint distribution of the signal *X* and the pattern W; and a loss function, *l(W, d(X)),* of making a decision *d(X)* when the actual pattern is *W*, then the *optimal Bayes decision rule* implements:

$$d_0(X) = \arg\min_{d(X)} \sum_W l(W, d(X)) \bullet P(W \mid X)$$

• If *l(W, d(X))* is a 0-1 loss function, i. e. error count, we have the well-known *maximum a posteriori* (MAP) *decision rule (same answer as in channel decoding)*:

$$d_{01}(X) = \arg\max_W P(X \mid W) \bullet P(W)$$

Summary

- Today's Class
  - Supervised learning basics (Chapter 2)
  - Web: http://www.ece.gatech.edu/~chl/ECE7252.sp08
- Next Classes
  - Machine learning tools and web resources
- Exercises: make sure you know the topics discussed and how to do all the exercises suggested in Lecture 2
- Reading Assignments
  - HTF, Chapters 1 & 2
  - *HAL's Legacy*, Chapters 6, 7 & 8