Foundations of Machine Learning

Dr. Sriparna Saha

Associate Professor

Department of Computer Science and Engineering

Indian Institute of Technology Patna

Bihar, India

Email: sriparna@iitp.ac.in

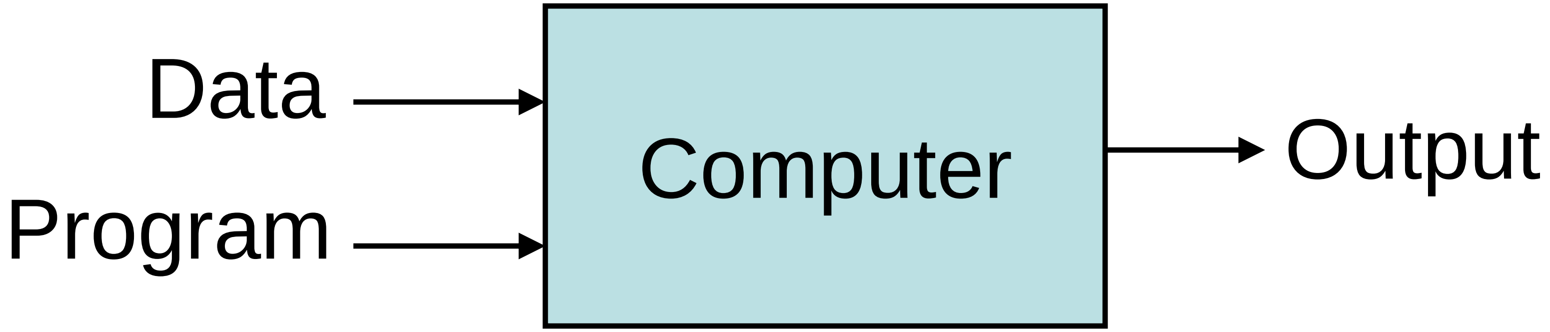
A Few Quotes

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- Machine learning is the hot new thing” (John Hennessy, President, Stanford)
- “Web rankings today are mostly a matter of machine learning” (Prabhakar Raghavan, Dir. Research, Yahoo)
- “Machine learning is going to result in a real revolution” (Greg Papadopoulos, CTO, Sun)
- “Machine learning is today’s discontinuity” (Jerry Yang, CEO, Yahoo)

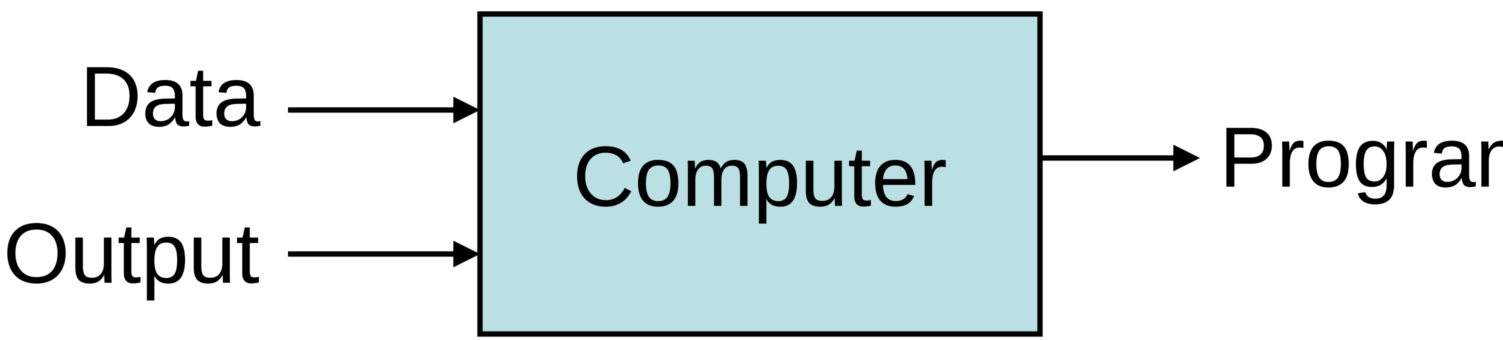
So What Is Machine Learning?

- Automating automation
- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!

Traditional Programming



Machine Learning



Magic?

No, more like gardening

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging
- [Your favorite area]

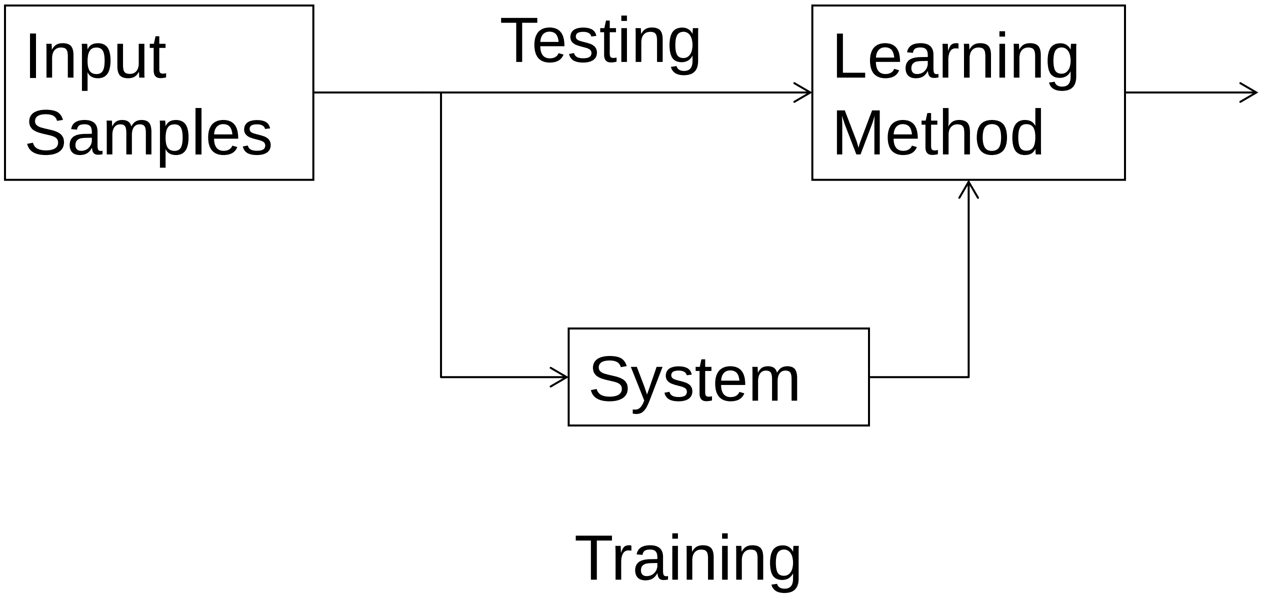
Types of Learning

- **Supervised (inductive) learning**
  - Training data includes desired outputs
- **Unsupervised learning**
  - Training data does not include desired outputs
- **Semi-supervised learning**
  - Training data includes a few desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

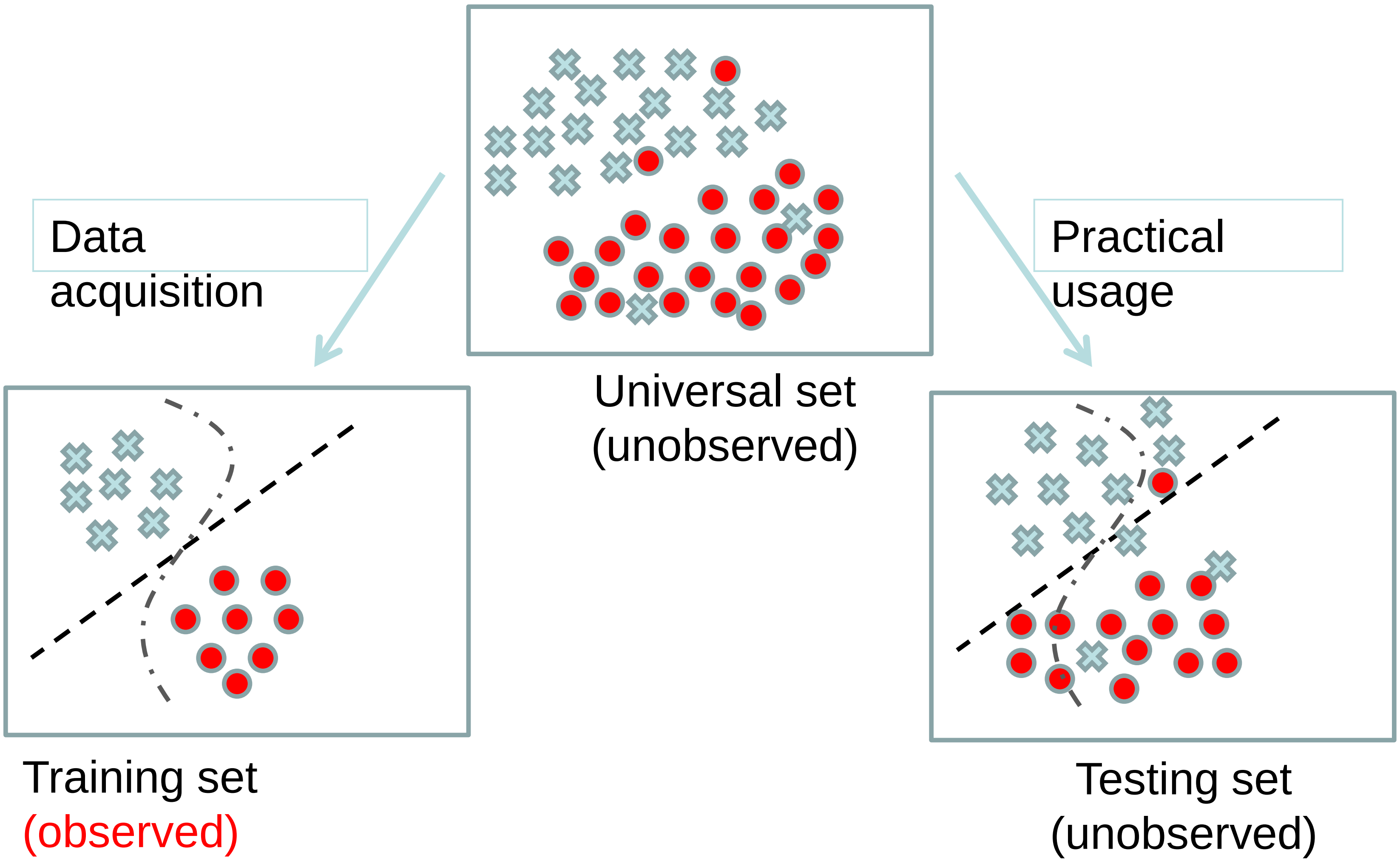
Inductive Learning

- **Given** examples of a function *(X, F(X))*
- **Predict** function *F(X)* for new examples *X*
  - Discrete *F(X)*: Classification
  - Continuous *F(X)*: Regression
  - *F(X)* = Probability(*X*): Probability estimation

Learning system model



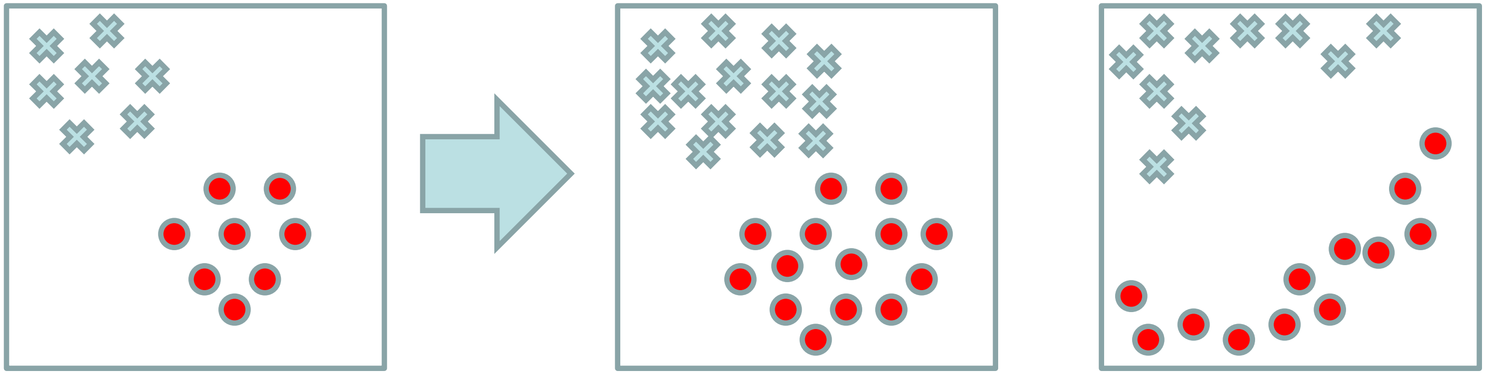
Training and testing



Training and testing

- Training is the process of making the system able to learn.

- No free lunch rule:
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias



- Some Applications:

From
Jim Elder
829 Loop Street, Apt 300
Allentown, New York 14707

Nov 10, 1999

To
Dr. Bob Grant
602 Queensberry Parkway
Omar, West Virginia 25638

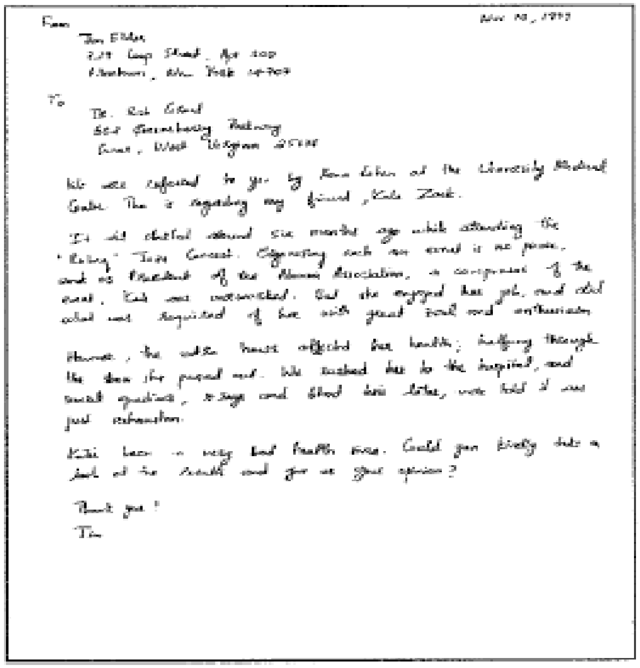
We were referred to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.

It all started around six months ago while attending the "Rubeq" Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm.

However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, x-rays and blood tests later, were told it was just exhaustion.

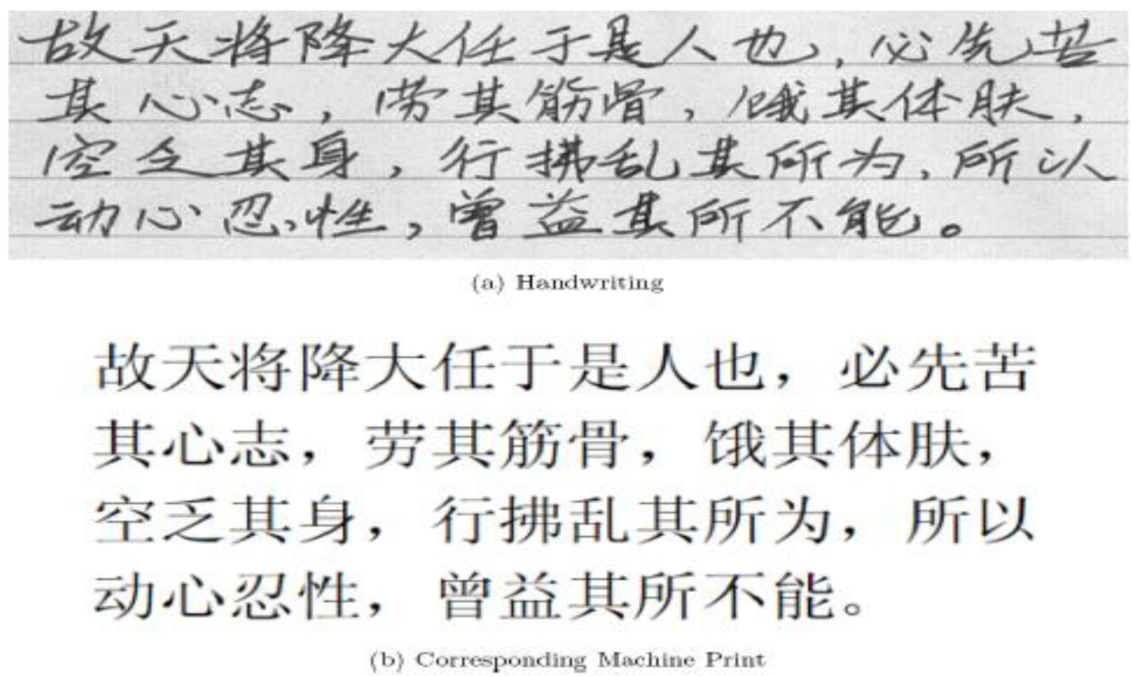
Kate's been in very bad health since. Could you kindly take a look at the results and give us your opinion?

Thank you!
Jim



English handwriting recognition.

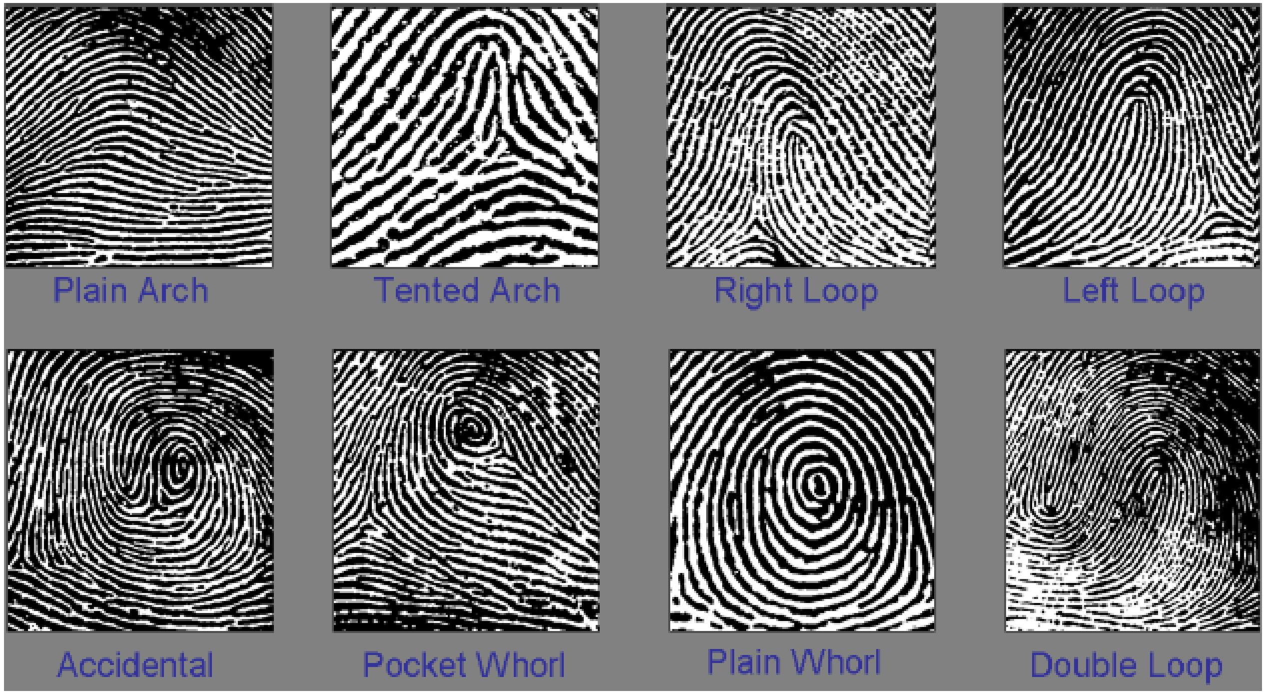
- Some Applications:



Chinese handwriting recognition.

Applications of ML?

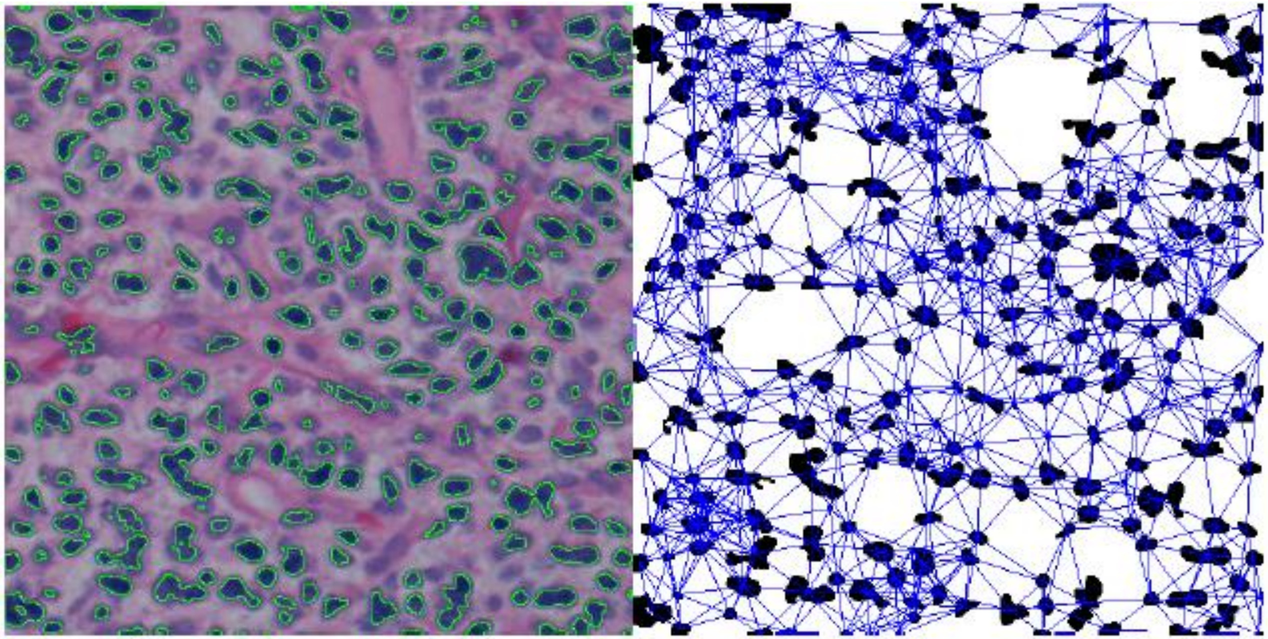
- Some Applications:



Fingerprint recognition.

Applications of ML

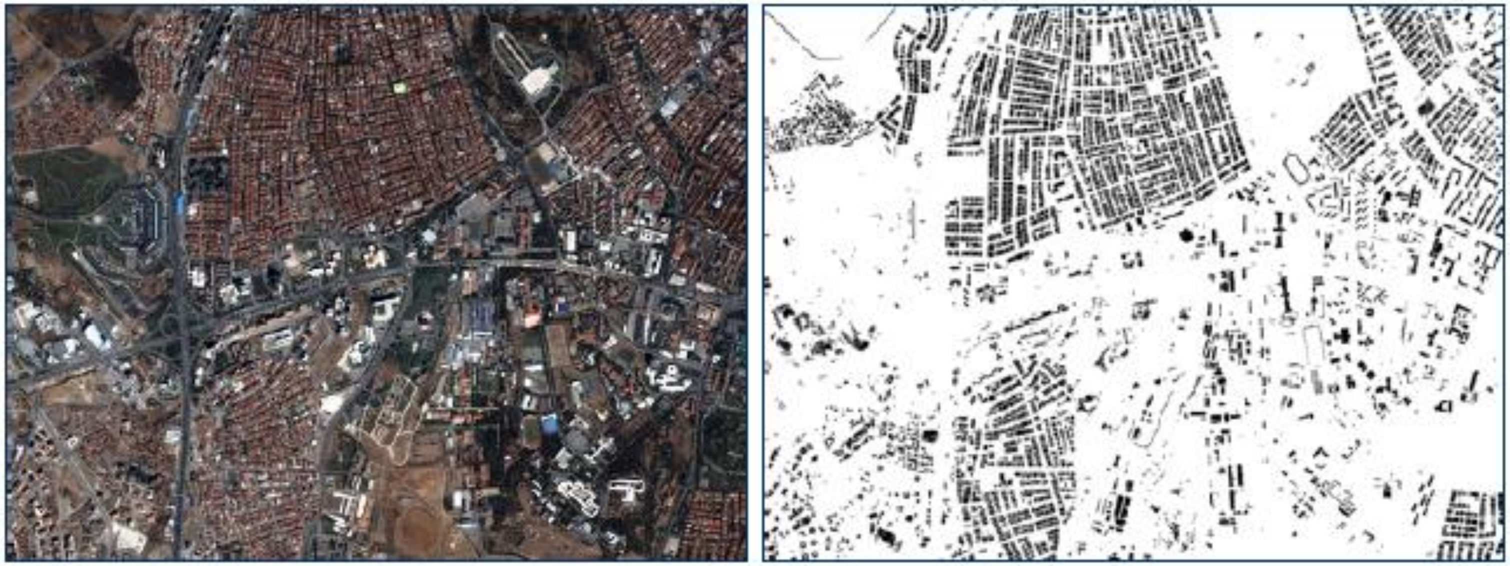
- Some Applications:



Cancer detection and grading using microscopic tissue data.

Applications of ML

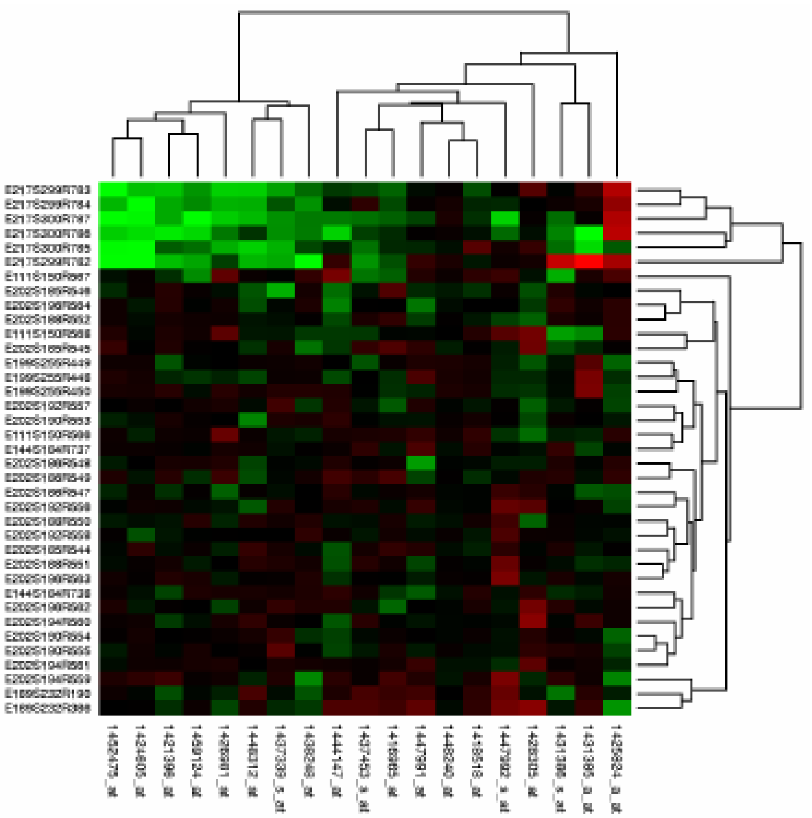
- Some Applications:



Building and building group recognition using satellite data.

Applications of ML

- Some Applications:



Clustering of microarray data.

Performance

- There are several factors affecting the performance:
  - **Types of training** provided
  - The form and extent of any initial **background knowledge**
  - The **type of feedback** provided
  - The **learning algorithms** used

- Two important factors:
  - Modeling
  - Optimization

Algorithms

- The success of machine learning system also depends on the algorithms.

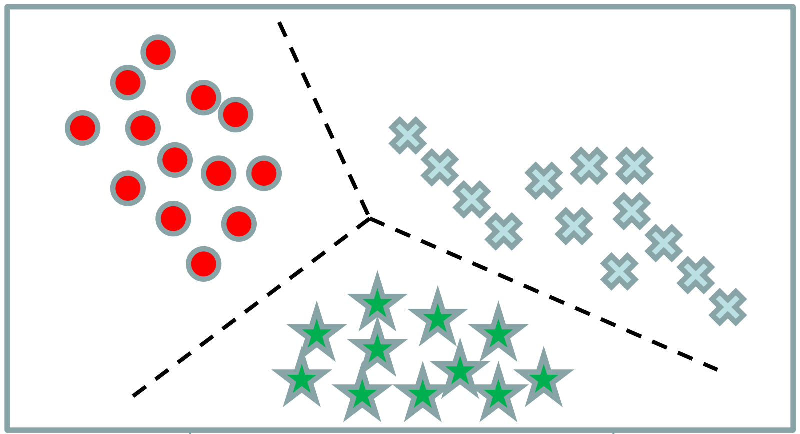
- The algorithms control the search to find and build the knowledge structures.

- The learning algorithms should extract useful information from training examples.

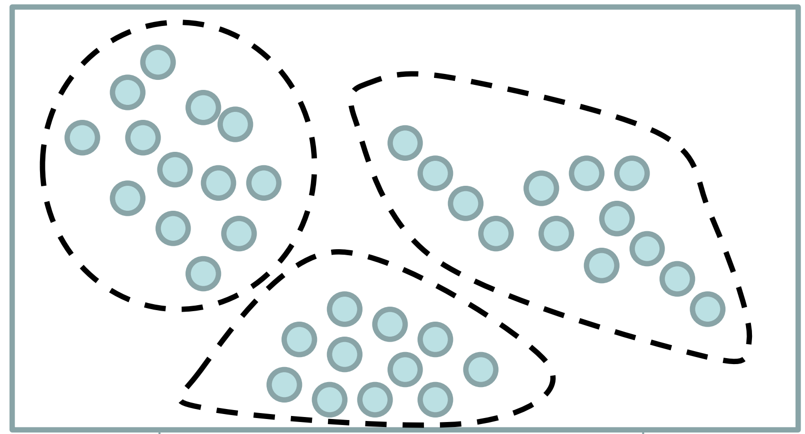
Algorithms

- **Supervised learning** ($\{x_n \in R^d, y_n \in R\}_{n=1}^N$)
  - Prediction
  - Classification (discrete labels), Regression (real values)
- **Unsupervised learning** ($\{x_n \in R^d\}_{n=1}^N$)
  - Clustering
  - Probability distribution estimation
  - Finding association (in features)
  - Dimension reduction
- **Semi-supervised learning**
- **Reinforcement learning**
  - Decision making (robot, chess machine)

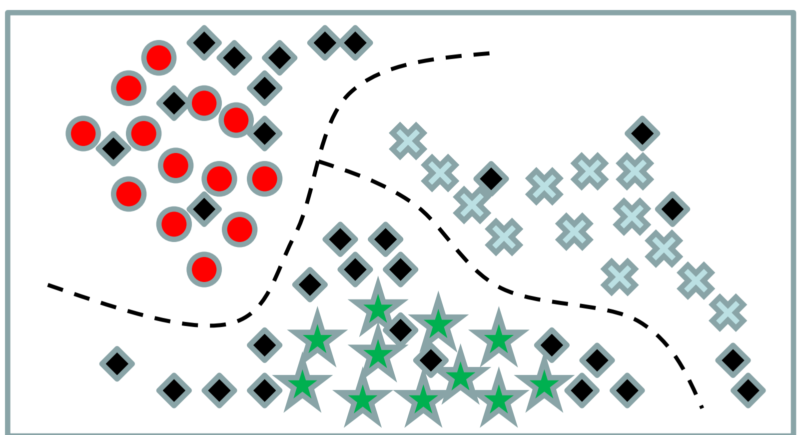
Algorithms



Supervised learning



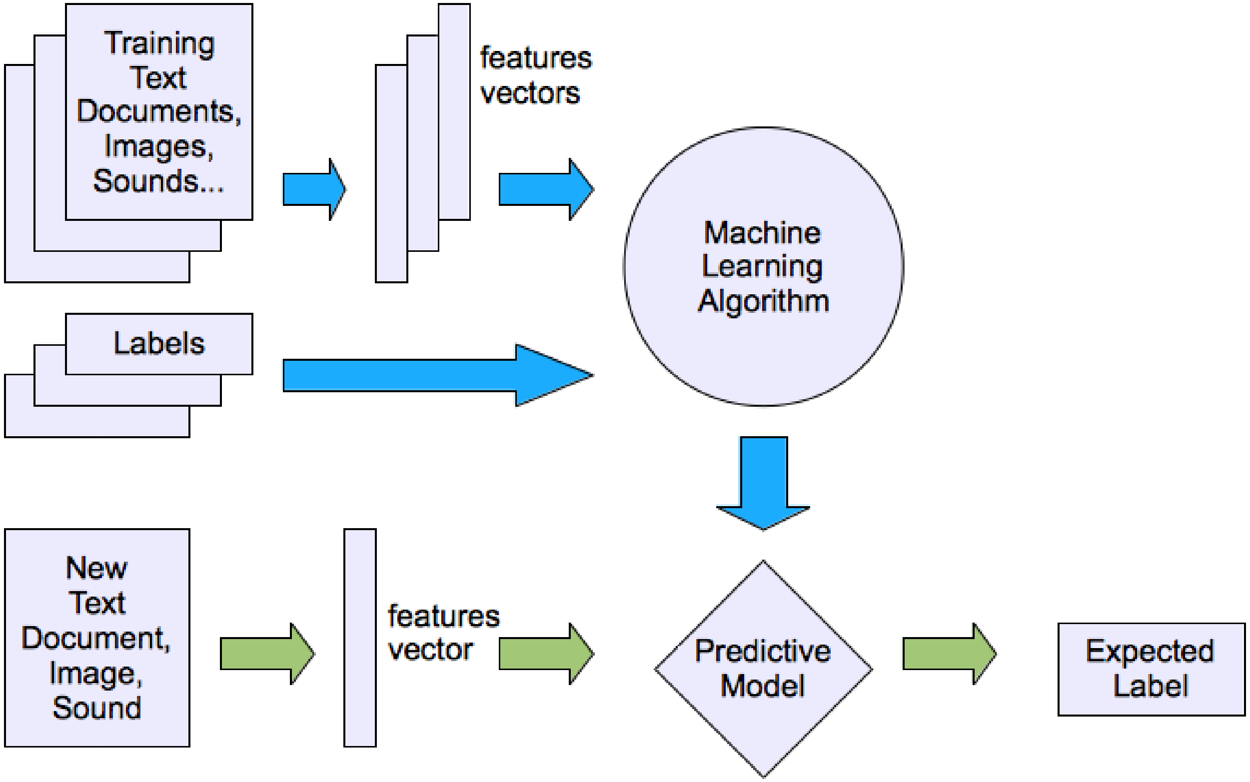
Unsupervised learning



Semi-supervised learning

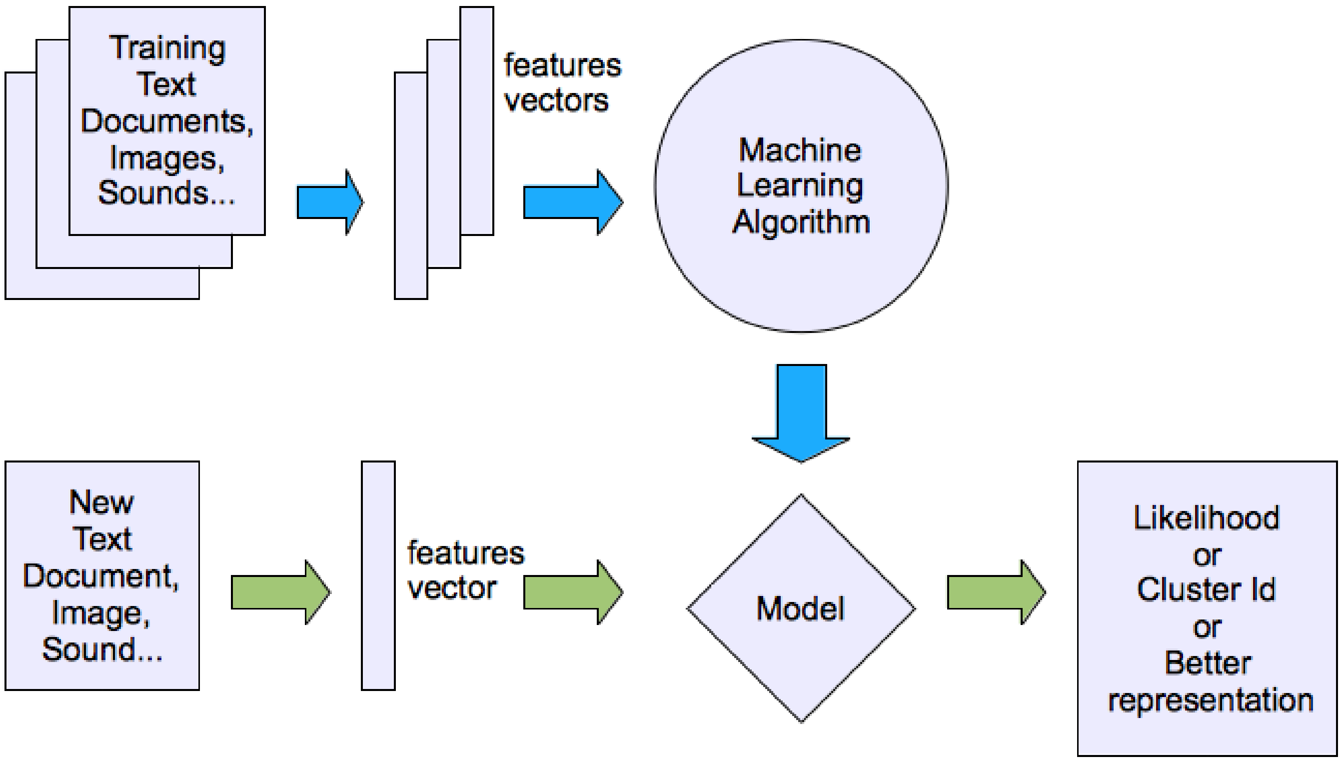
Machine learning structure

- Supervised learning



Machine learning structure

- Unsupervised learning

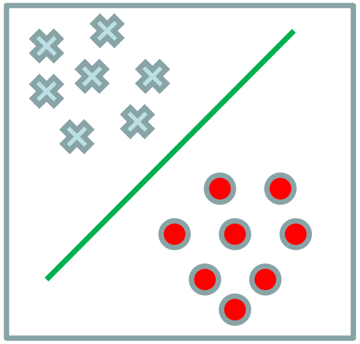


Learning techniques

- Supervised learning categories and techniques
  - **Linear classifier** (numerical functions)
  - **Parametric** (Probabilistic functions)
    - Naïve Bayes, Gaussian discriminant analysis (GDA), Hidden Markov models (HMM), Probabilistic graphical models
  - **Non-parametric** (Instance-based functions)
    - *K*-nearest neighbors, Kernel regression, Kernel density estimation, Local regression
  - **Non-metric** (Symbolic functions)
    - Classification and regression tree (CART), decision tree
  - **Aggregation**
    - Bagging (bootstrap + aggregation), Adaboost, Random forest

Learning techniques

- Linear classifier



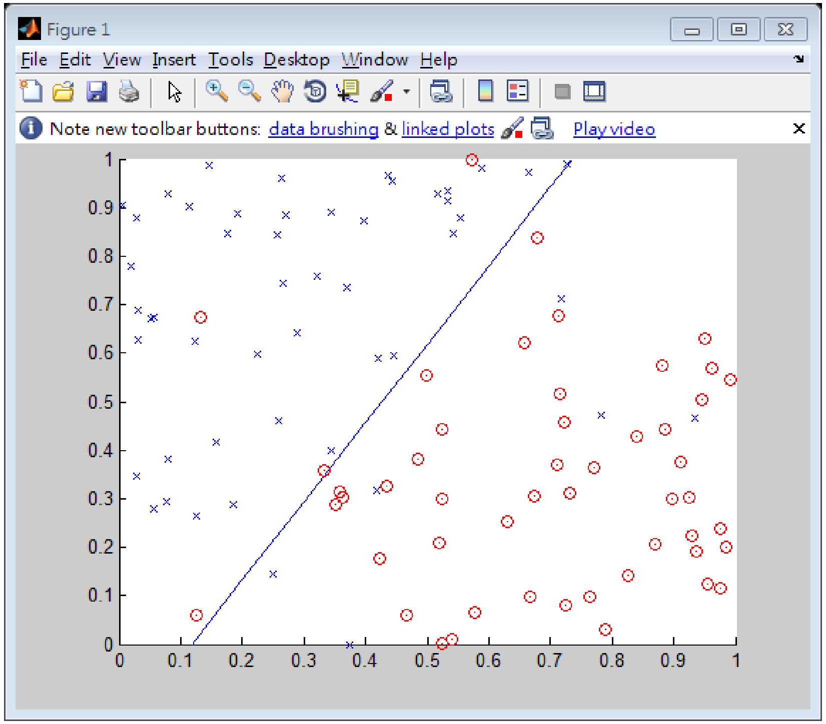
$$g(x_n) = sign(w^T x_n)$$

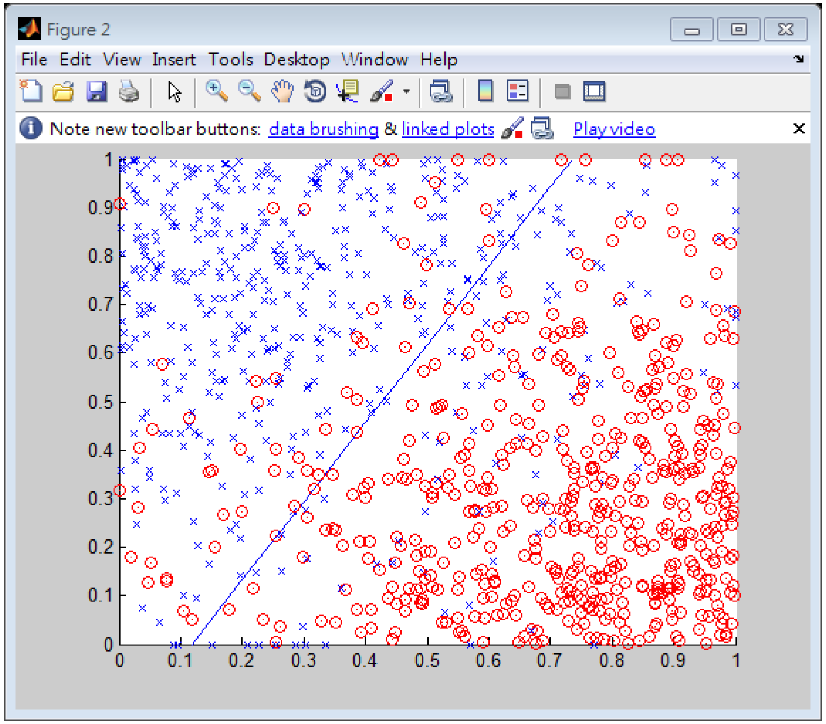
, where *w* is an *d*-dim vector (learned)

- Techniques:
  - Perceptron
  - Logistic regression
  - Support vector machine (SVM)
  - Ada-line
  - Multi-layer perceptron (MLP)

Learning techniques

Using **perceptron learning algorithm**(PLA)





Training

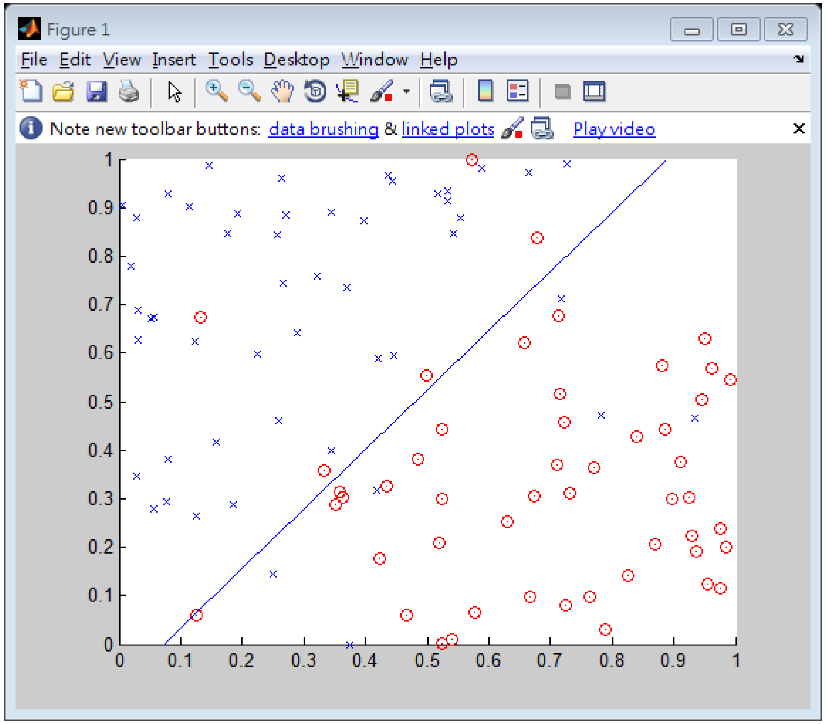
Error rate:
0.10

Testing

Error rate:
0.156

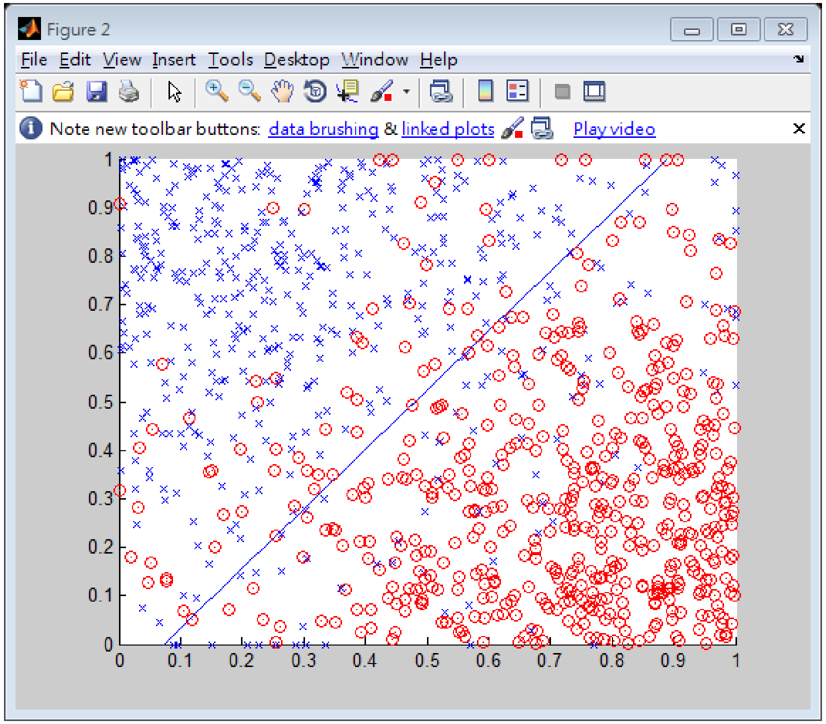
Learning techniques

Using **logistic regression**



Training

Error rate:
0.11

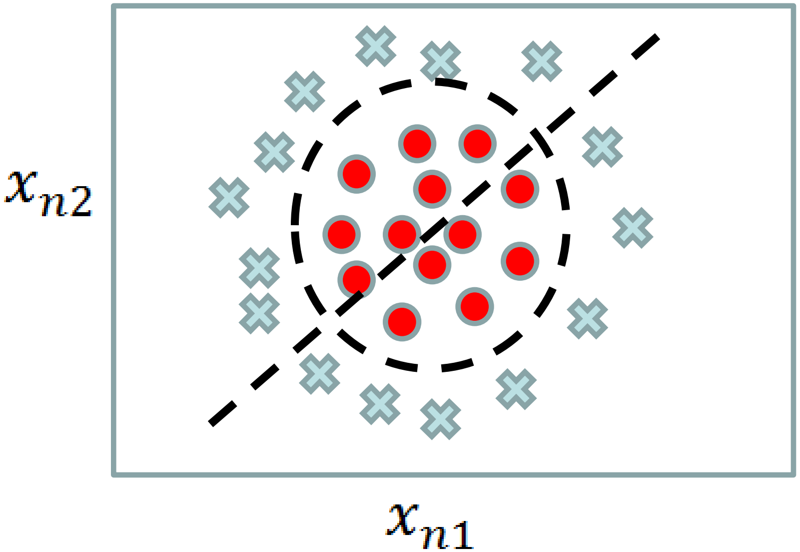


Testing

Error rate:
0.145

Learning techniques

- Non-linear case



$$x_n = [x_{n1}, x_{n2}]$$

$$x_n = [x_{n1}, x_{n2}, x_{n1} * x_{n2}, {x_{n1}}^2, {x_{n2}}^2]$$

$$g(x_n) = sign(w^T x_n)$$

- Support vector machine (SVM):
  - Linear to nonlinear: **Feature transform** and **kernel function**

Learning techniques

- Unsupervised learning categories and techniques
  - **Clustering**
    - K-means clustering
    - Spectral clustering
  - **Density Estimation**
    - Gaussian mixture model (GMM)
    - Graphical models
  - **Dimensionality reduction**
    - Principal component analysis (PCA)
    - Factor analysis

Why "Learn"?

- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- There is no need to "learn" to calculate payroll
- Learning is used when:
  - Human expertise does not exist (navigating on Mars),
  - Humans are unable to explain their expertise (speech recognition)
  - Solution changes in time (routing on a computer network)
  - Solution needs to be adapted to particular cases (user biometrics)

What We Talk About When We Talk About“Learning”

- Learning general models from a data of particular examples
- Data is cheap and abundant (data warehouses, data marts); knowledge is expensive and scarce.
- Example in retail: Customer transactions to consumer behavior:

  People who bought “Da Vinci Code” also bought “The Five People You Meet in Heaven” (www.amazon.com)
- Build a model that is *a good and useful approximation* to the data.

Growth of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment
  - It turns out to be difficult to extract knowledge from human experts→*failure of expert systems in the 1980's.*

Applications

- Association Analysis
- Supervised Learning
  - Classification
  - Regression/Prediction
- Unsupervised Learning
- Reinforcement Learning

Learning Associations

- Basket analysis:

  $P(Y \mid X)$ probability that somebody who buys X also buys Y where X and Y are products/services.

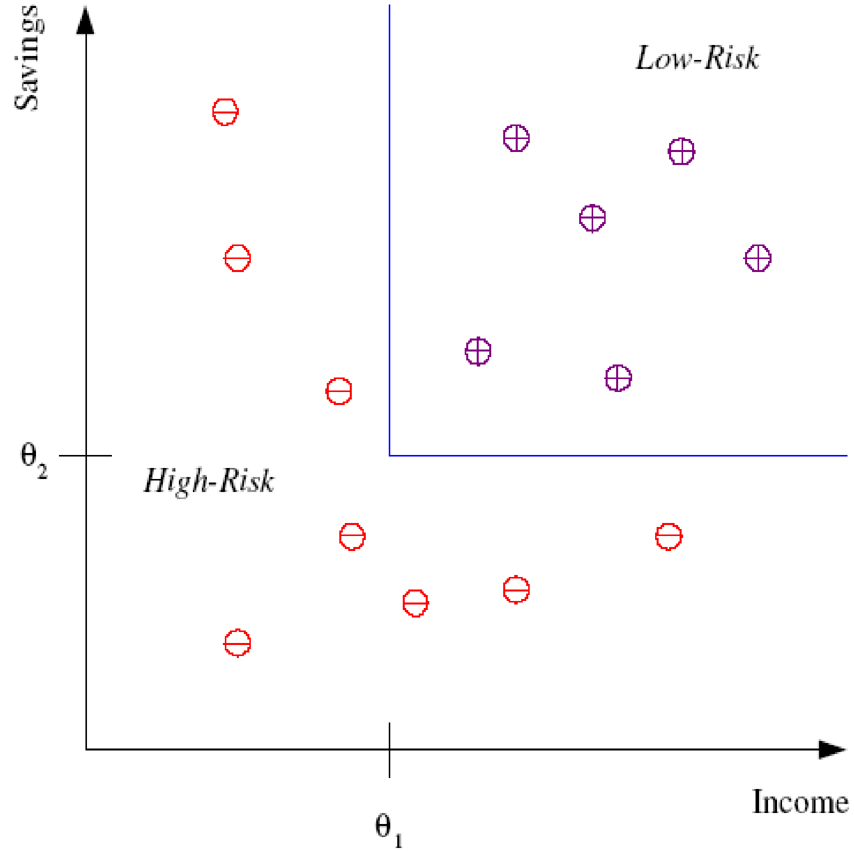
  Example: P (chips | beer) = 0.7

Market-Basket transactions

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* > θ_1 AND *savings* > θ_2

THEN low-risk ELSE high-risk

Model

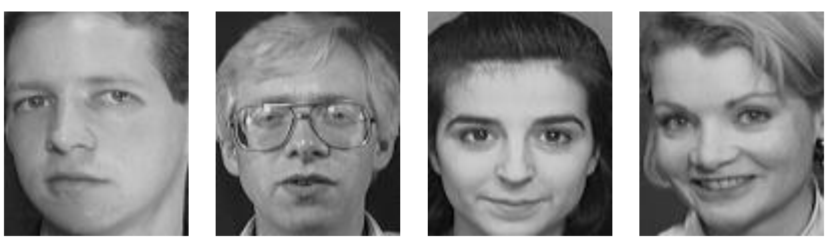
Classification: Applications

- Aka Pattern recognition
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- Web Advertizing: Predict if a user clicks on an ad on the Internet.

Face Recognition

Training examples of a person

Test images



AT&T Laboratories, Cambridge UK
http://www.uk.research.att.com/facedatabase.html

Prediction: Regression

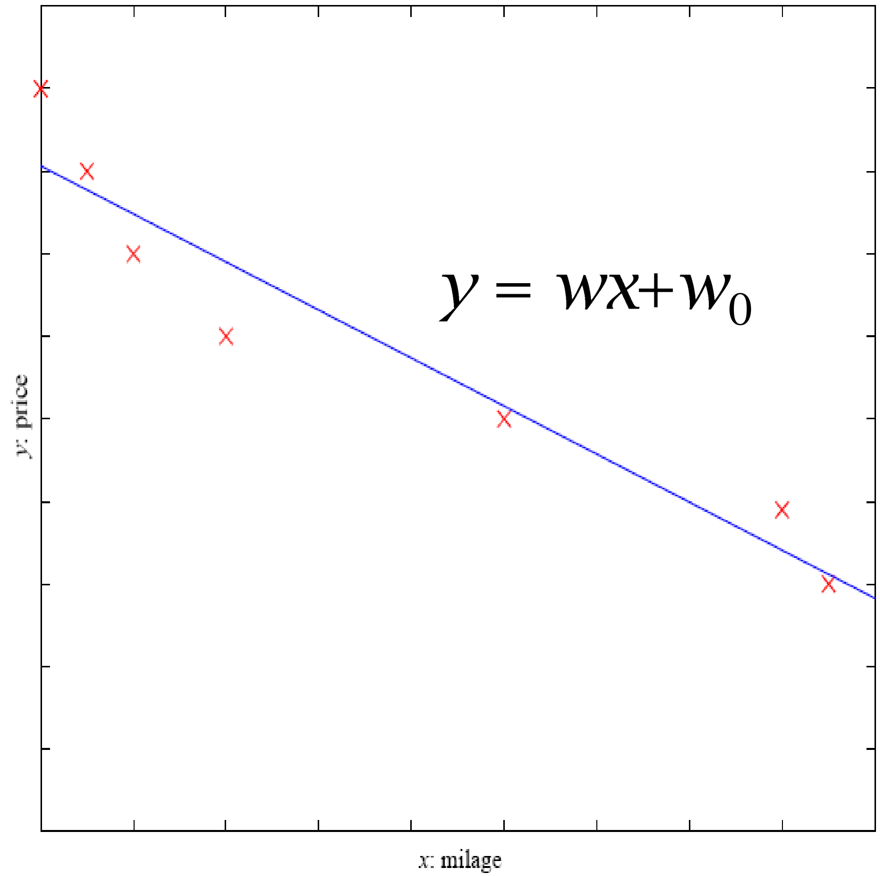
- Example: Price of a used car
- *x* : car attributes

  y : price

  $$y = g(x \mid \theta)$$

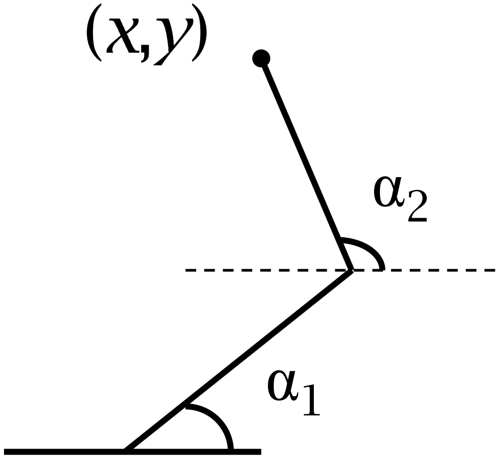
  g ( ) model,

  θ parameters



Regression Applications

- Navigating a car: Angle of the steering wheel (CMU NavLab)
- Kinematics of a robot arm



$\alpha_1 = g_1(x,y)$

$\alpha_2 = g_2(x,y)$

Supervised Learning: Uses

Example: decision trees tools that create rules

- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: The rule is easy to understand
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Unsupervised Learning

- Learning “what normally happens”
- No output
- Clustering: Grouping similar instances
- Other applications: Summarization, Association Analysis
- Example applications
  - Customer segmentation in CRM
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

Reinforcement Learning

- Topics:
  - Policies: what actions should an agent take in a particular situation
  - Utility estimation: how good is a state ($\rightarrow$used by policy)
- No supervised output but delayed reward
- Credit assignment problem (what was responsible for the outcome)
- Applications:
  - Game playing
  - Robot in a maze
  - Multiple agents, partial observability, …

Learning
An example application

- An emergency room in a hospital measures 17 variables (e.g., blood pressure, age, etc) of newly admitted patients.
- A decision is needed: whether to put a new patient in an intensive-care unit.
- Due to the high cost of ICU, those patients who may survive less than a month are given higher priority.
- Problem: to predict high-risk patients and discriminate them from low-risk patients.

Another application

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - Marital status
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: to decide whether an application should be approved, or to classify applications into two categories, approved and not approved.

Machine learning and our focus

- Like human learning from past experiences.
- A computer does not have "experiences".
- A computer system learns from data, which represent some "past experiences" of an application domain.
- Our focus: learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk.
- The task is commonly called: Supervised learning, classification, or inductive learning.

The data and the goal

- Data: A set of data records (also called examples, instances or cases) described by
  - k attributes: A_1, A_2, … A_k.
  - a class: Each example is labelled with a pre-defined class.
- Goal: To learn a classification model from the data that can be used to predict the classes of new (future, or test) cases/instances.

An example: data (loan application)

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

An example: the learning task

- Learn a classification model from the data
- Use the model to classify future loan applications into
  - Yes (approved) and
  - No (not approved)
- What is the class for following

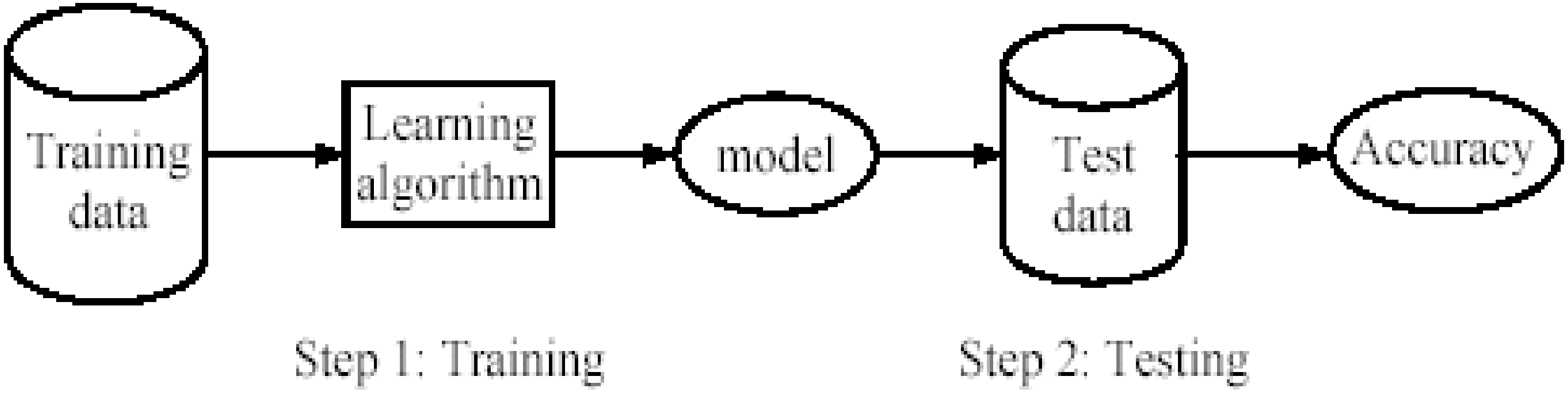
Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

Supervised learning process: two steps

Learning (training): Learn a model using the training data

Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifica tions}}{\text{Total number of test cases}},$$



What do we mean by learning?

- Given
  - a data set *D*,
  - a task *T*, and
  - a performance measure *M*,

  a computer system is said to **learn** from *D* to perform the task *T* if after learning the system's performance on *T* improves as measured by *M*.
- In other words, the learned model helps the system to perform *T* better as compared to no learning.

An example

- Data: Loan application data
- Task: Predict whether a loan should be approved or not.
- Performance measure: accuracy.

No learning: classify all future applications (test data) to the majority class (i.e., Yes):

Accuracy = 9/15 = 60%.

- We can do better than 60% with learning.

Fundamental assumption of learning

Assumption: The distribution of training examples is identical to the distribution of test examples (including future unseen examples).

- In practice, this assumption is often violated to certain degree.
- Strong violations will clearly result in poor classification accuracy.
- To achieve good accuracy on the test data, training examples must be sufficiently representative of the test data.

Evaluating classification methods

- **Predictive accuracy**

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}}$$

- Efficiency
  - time to construct the model
  - time to use the model
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability:
  - understandable and insight provided by the model
- Compactness of the model: size of the tree, or the number of rules.

Evaluation methods

- **Holdout set**: The available data set *D* is divided into two disjoint subsets,
  - the *training set* D_{train} (for learning a model)
  - the *test set* D_{test} (for testing the model)
- **Important:** training set should not be used in testing and the test set should not be used in learning.
  - Unseen test set provides a unbiased estimate of accuracy.
- The test set is also called the holdout set. (the examples in the original data set *D* are all labeled with classes.)
- This method is mainly used when the data set *D* is large.

Evaluation methods (cont…)

- **n-fold cross-validation**: The available data is partitioned into *n* equal-size disjoint subsets.
- Use each subset as the test set and combine the rest *n*-1 subsets as the training set to learn a classifier.
- The procedure is run *n* times, which give *n* accuracies.
- The final estimated accuracy of learning is the average of the *n* accuracies.
- 10-fold and 5-fold cross-validations are commonly used.
- This method is used when the available data is not large.

Evaluation methods (cont…)

- **Leave-one-out cross-validation**: This method is used when the data set is very small.
- It is a special case of cross-validation
- Each fold of the cross validation has only a single test example and all the rest of the data is used in training.
- If the original data has *m* examples, this is *m*-fold cross-validation

Evaluation methods (cont…)

- **Validation set**: the available data is divided into three subsets,
  - a training set,
  - a validation set and
  - a test set.
- A validation set is used frequently for estimating parameters in learning algorithms.
- In such cases, the values that give the best accuracy on the validation set are used as the final parameter values.
- Cross-validation can be used for parameter estimating as well.

Classification measures

- Accuracy is only one measure (error = 1-accuracy).
- **Accuracy is not suitable in some applications**.
- In text mining, we may only be interested in the documents of a particular topic, which are only a small portion of a big document collection.
- In classification involving skewed or highly imbalanced data, e.g., network intrusion and financial fraud detections, we are interested only in the minority class.
  - High accuracy does not mean any intrusion is detected.
  - E.g., 1% intrusion. Achieve 99% accuracy by doing nothing.
- The class of interest is commonly called the **positive class**, and the rest **negative classes.**

Precision and **recall** measures

- Used in information retrieval and text classification.
- We use a confusion matrix to introduce

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

where

TP: the number of correct classifications of the positive examples (**true positive**),
FN: the number of incorrect classifications of positive examples (**false negative**),
FP: the number of incorrect classifications of negative examples (**false positive**), and
TN: the number of correct classifications of negative examples (**true negative**).

Precision and **recall** measures (cont…)

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

Precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

An example

	Classified Positive	Classified Negative
Actual Positive	1	99
Actual Negative	0	1000

- This confusion matrix gives
  - precision p = 100% and
  - recall r = 1%

    because we only classified one positive example correctly and no negative examples wrongly.
- Note: precision and recall only measure classification on the positive class.

F_1-value (also called F_1-score)

- It is hard to compare two classifiers using two measures. F_1 score combines precision and recall into one measure

$$F_1 = \frac{2pr}{p+r}$$

F_1-score is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

- The harmonic mean of two numbers tends to be closer to the smaller of the two.
- For F_1-value to be large, both *p* and *r* much be large.

Resources: Datasets

- UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html
- UCI KDD Archive: http://kdd.ics.uci.edu/summary.data.application.html
- Statlib: http://lib.stat.cmu.edu/
- Delve: http://www.cs.utoronto.ca/~delve/

Resources: Journals

- Journal of Machine Learning Research
  www.jmlr.org
- Machine Learning
- IEEE Transactions on Neural Networks
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Annals of Statistics
- Journal of the American Statistical Association
- ...

Resources: Conferences

- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Neural Information Processing Systems (NIPS)
- Computational Learning
- International Joint Conference on Artificial Intelligence (IJCAI)
- ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)
- IEEE Int. Conf. on Data Mining (ICDM)

References and acknowledgemnent

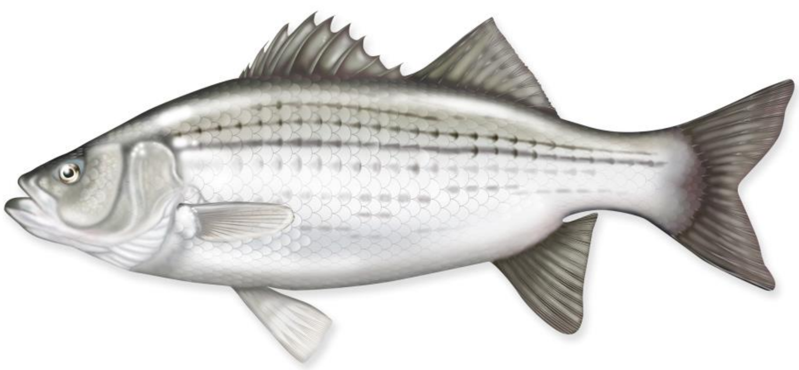
- Srihari, S.N., Covindaraju, Pattern recognition, Chapman &Hall, London, 1034-1041, 1993,
- Sergios Theodoridis, Konstantinos Koutroumbas , pattern recognition , Pattern Recognition ,Elsevier(USA)) ,1982
- R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, New York: John Wiley, 2001
- W.L.Chao, J.J.Ding, “Integrated Machine Learning Algorithms for Human Age Estimation”, NTU, 2011.
- Semi-supervised Learning, Avrim Blum.

An Example

- Suppose that:
  - A fish packing plant wants to automate the process of sorting incoming fish on a conveyor belt according to species,
  - There are two species:
    - Sea bass,
    - Salmon.

An Example

- How to **distinguish** one specie from the other ? (length, width, weight, number and shape of fins, tail shape,etc.)



An Example

- Suppose somebody at the fish plant say us that:
  - Sea bass is generally longer than a salmon
- Then our **models** for the fish:
  - **Sea bass** have some typical length, and this is greater than that for **salmon**.

An Example

- Then length becomes a **feature**,
- We might attempt to classify the fish by seeing whether or not the **length** of a fish exceeds some **critical value (threshold value) /*.**

An Example

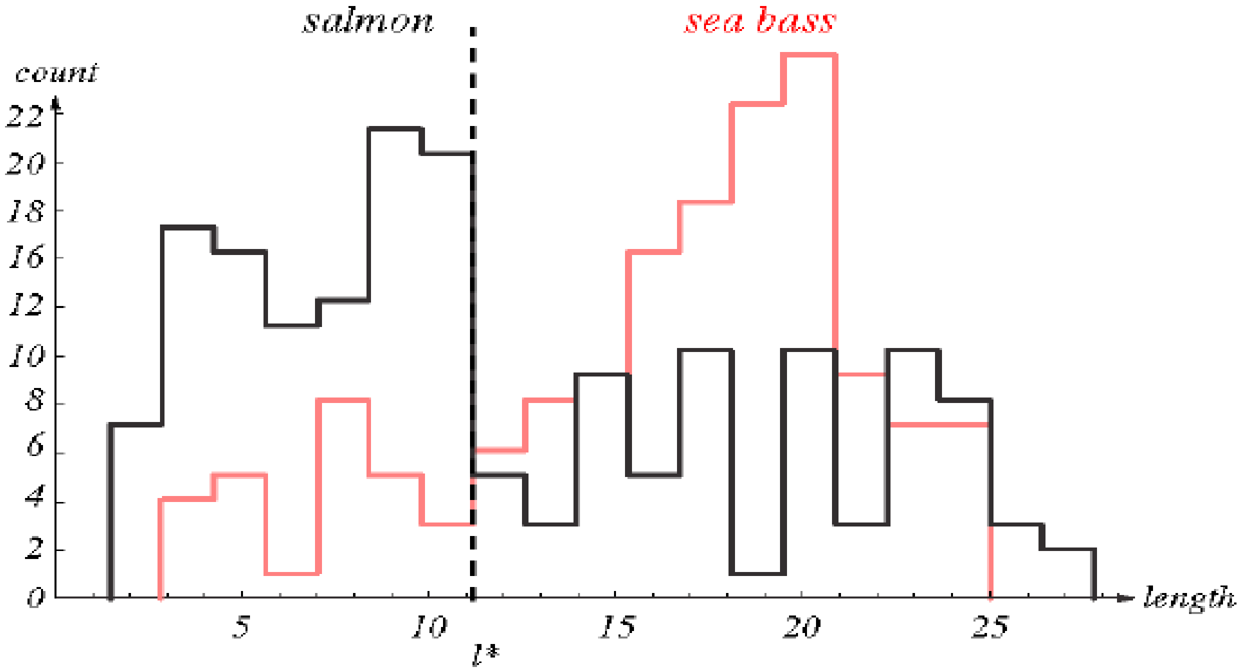
- How to decide on the **critical value (threshold value) ?**

An Example

- How to decide on the **critical value (threshold value) ?**
  - We could obtain some training samples of different types of fish,
  - make length measurements,
  - Inspect the results.

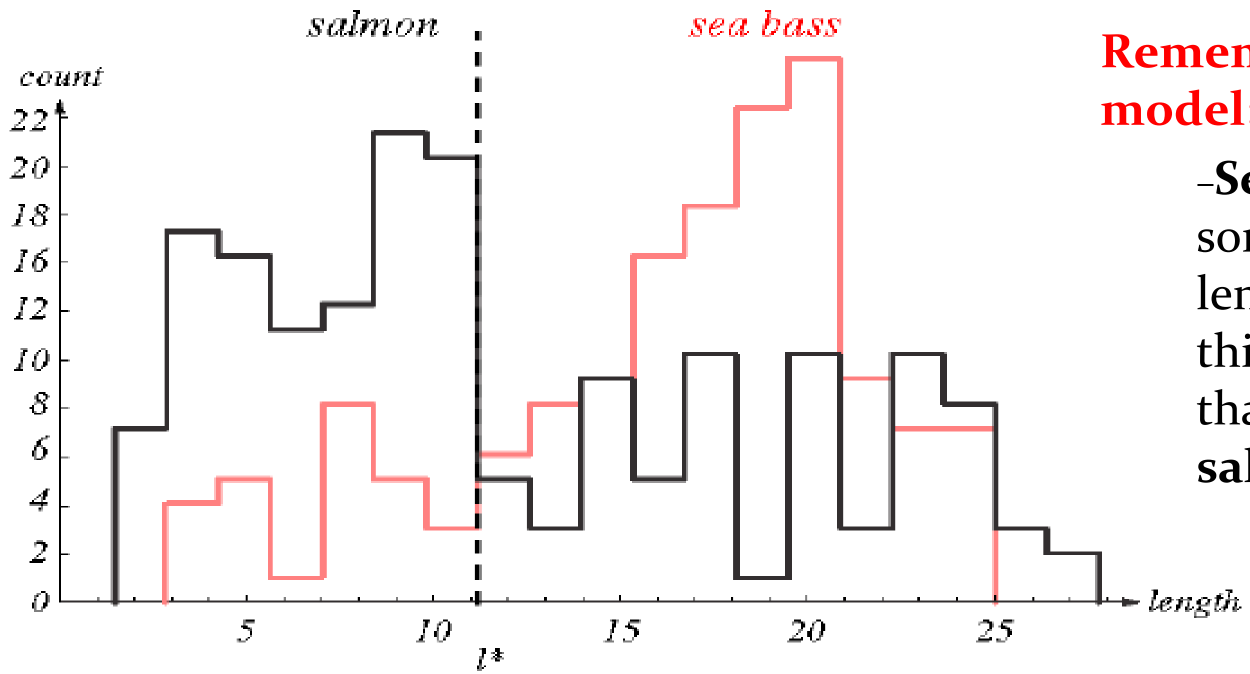
An Example

- Measurement results on the training sample related to two species.



An Example

- Can we **reliably seperate** sea bass from salmon by using length as a feature ?

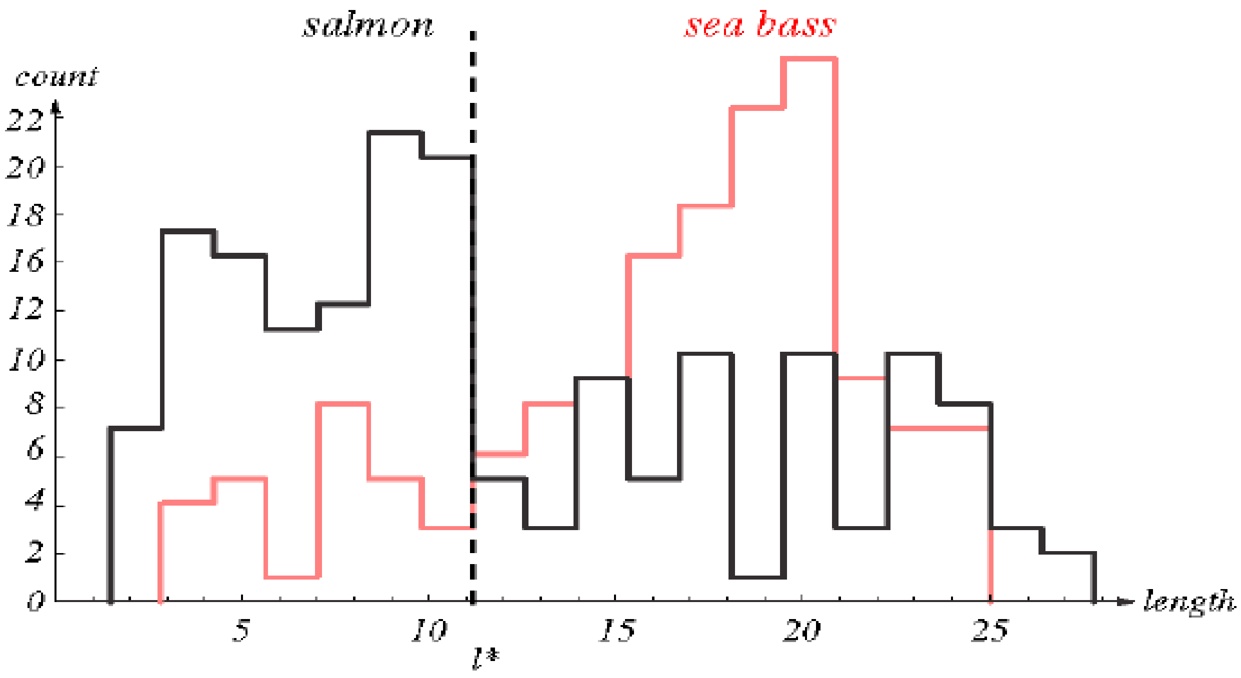


Remember our model:

–**Sea bass** have some typical length, and this is greater than that for **salmon**.

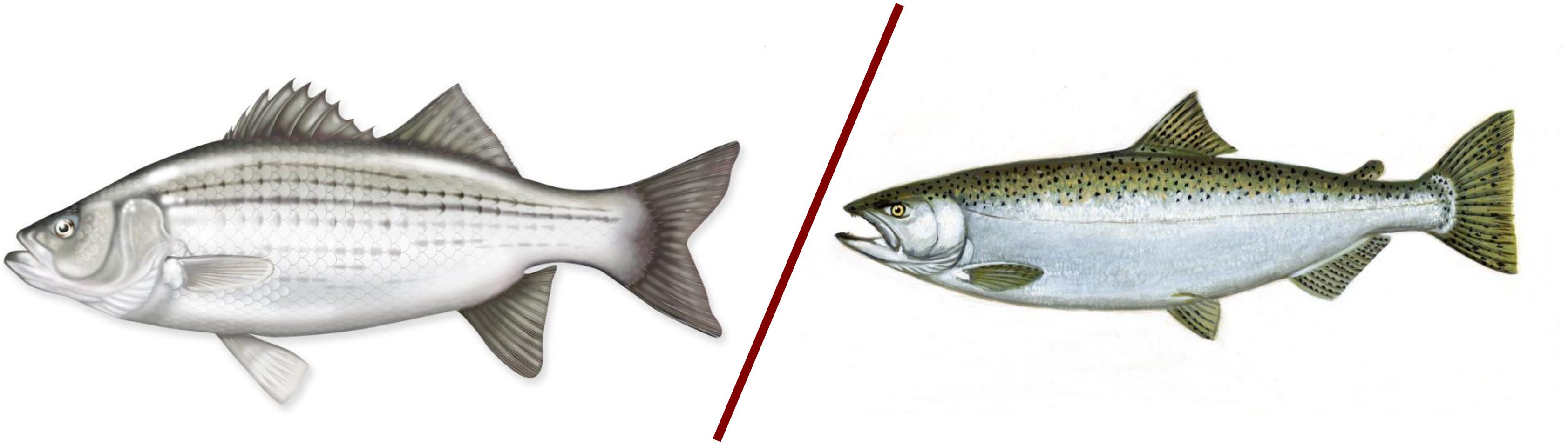
An Example

- From histogram we can see that single criteria is quite poor.



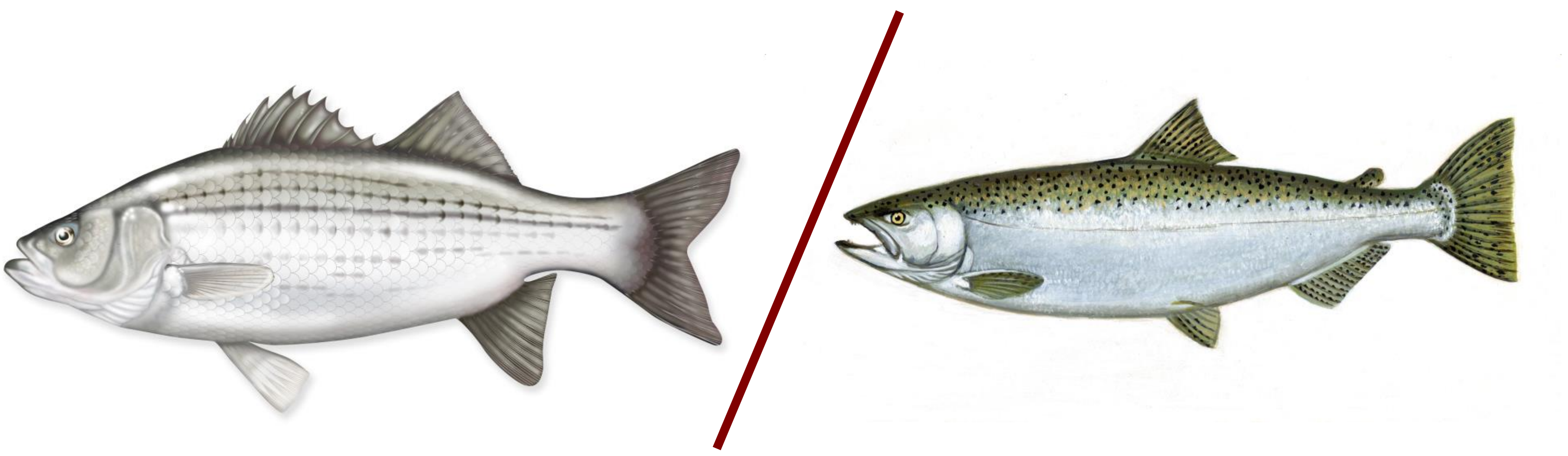
An Example

- It is obvious that length is not a good feature.
- **What we can do to seperate sea bass from salmon?**



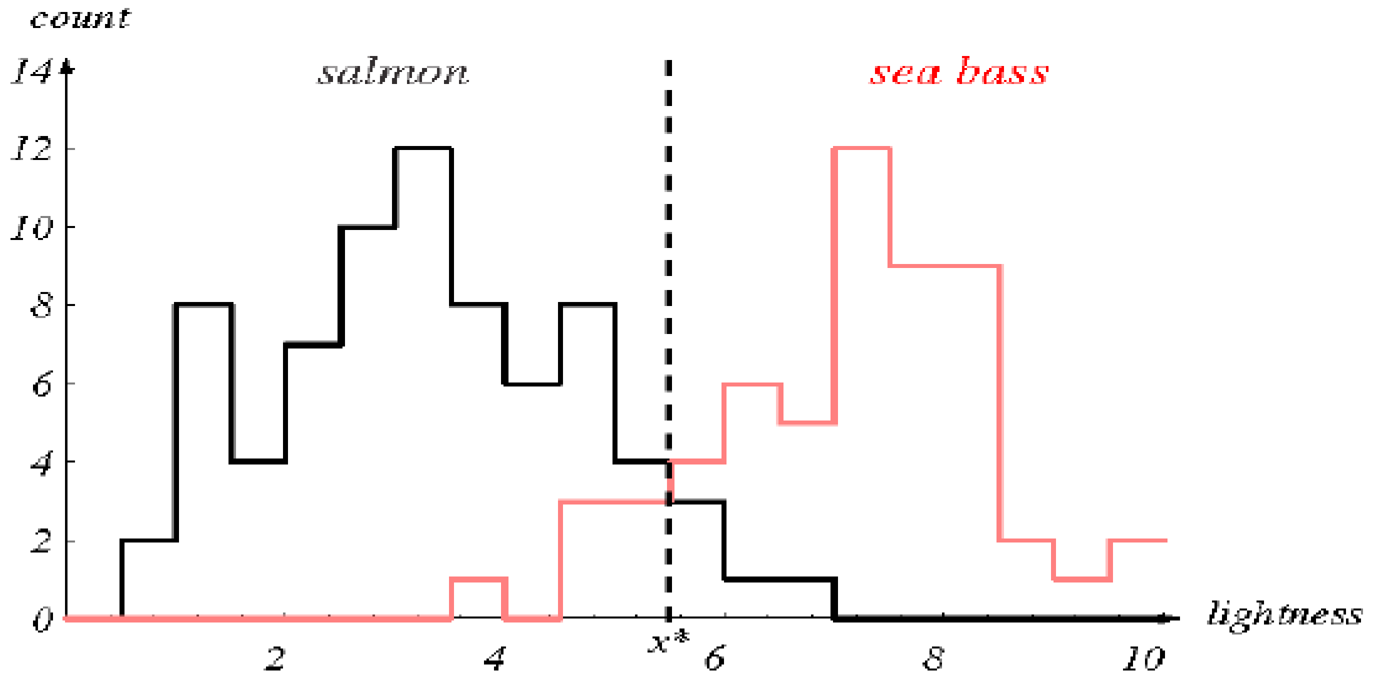
An Example

- **What we can do to seperate sea bass from salmon?**
- **Try another feature:**
  - **average lightness of the fish scales.**



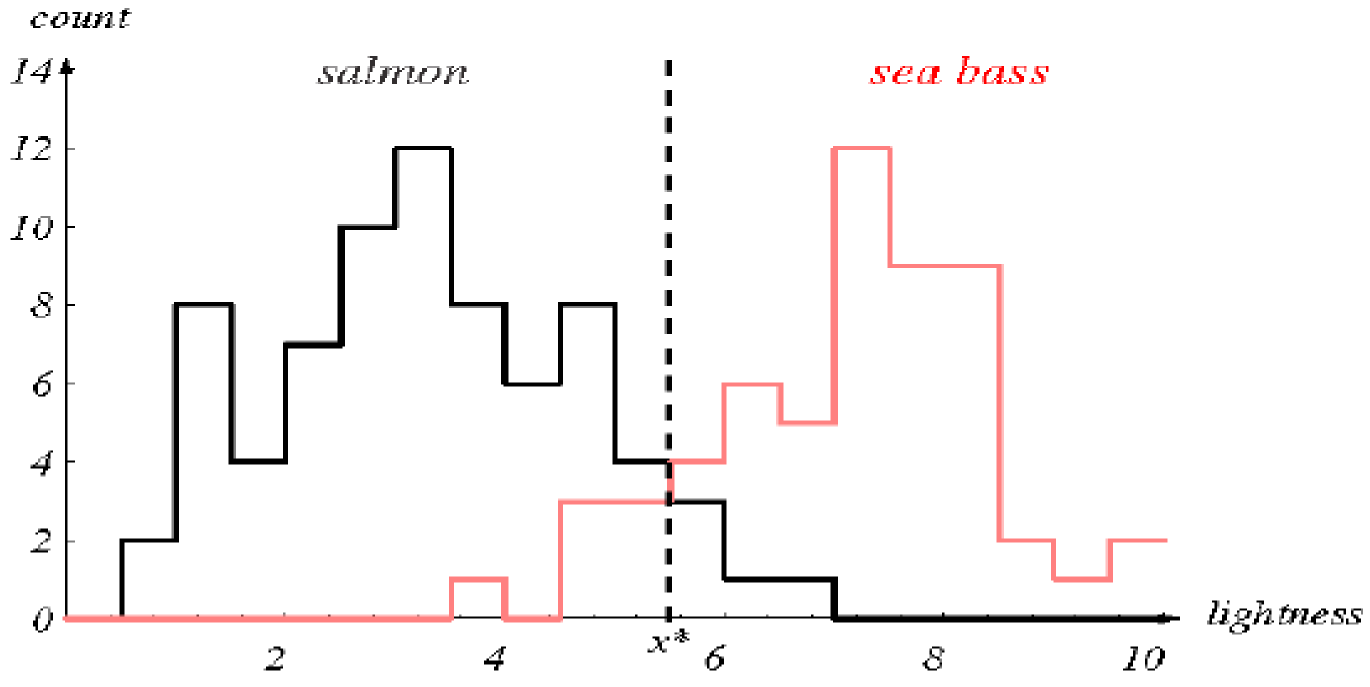
An Example

- Can we **reliably seperate** sea bass from salmon by using lightness as a feature ?



An Example

- Lighness is better than length as a feature but again there are some problems.



An Example

- Suppose we also know that:
  - Sea bass are typically wider than salmon.
- We can use more than one feature for our decision:
  - Lightness (x_1) and width (x_2)

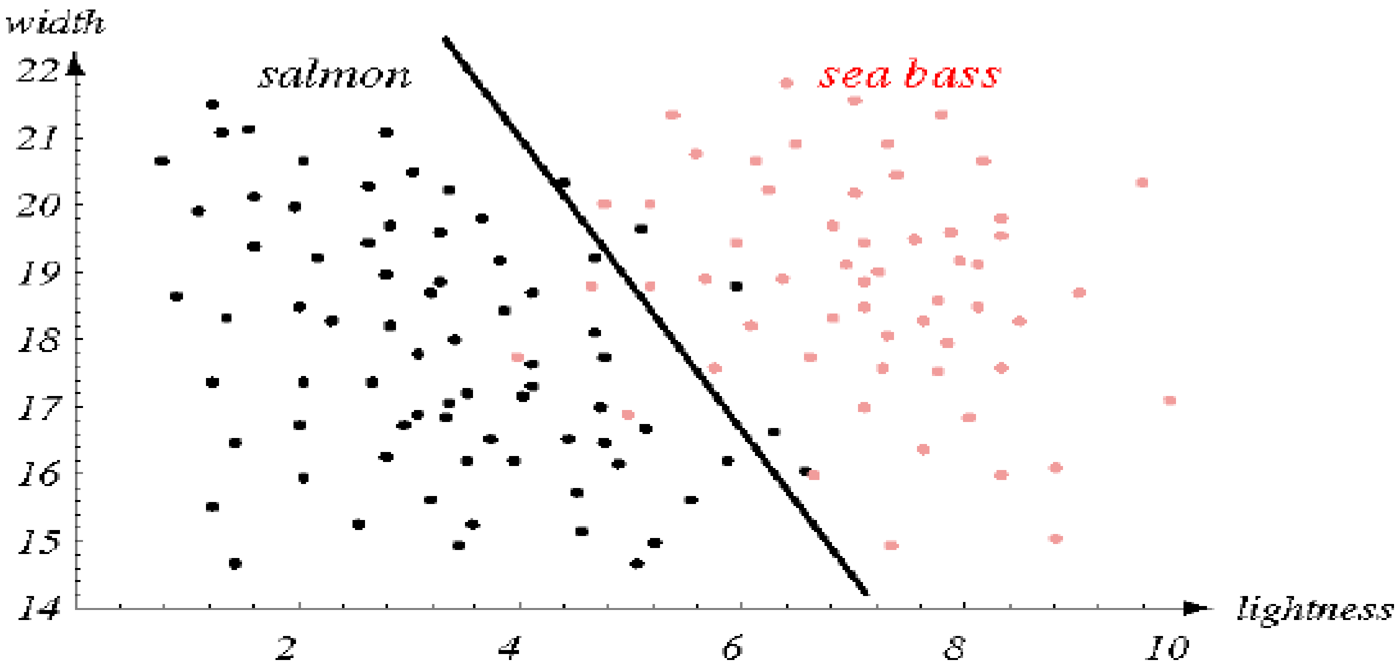
An Example

- Each fish is now a point in two dimension.
  - Lightness (x_1) and width (x_2)

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

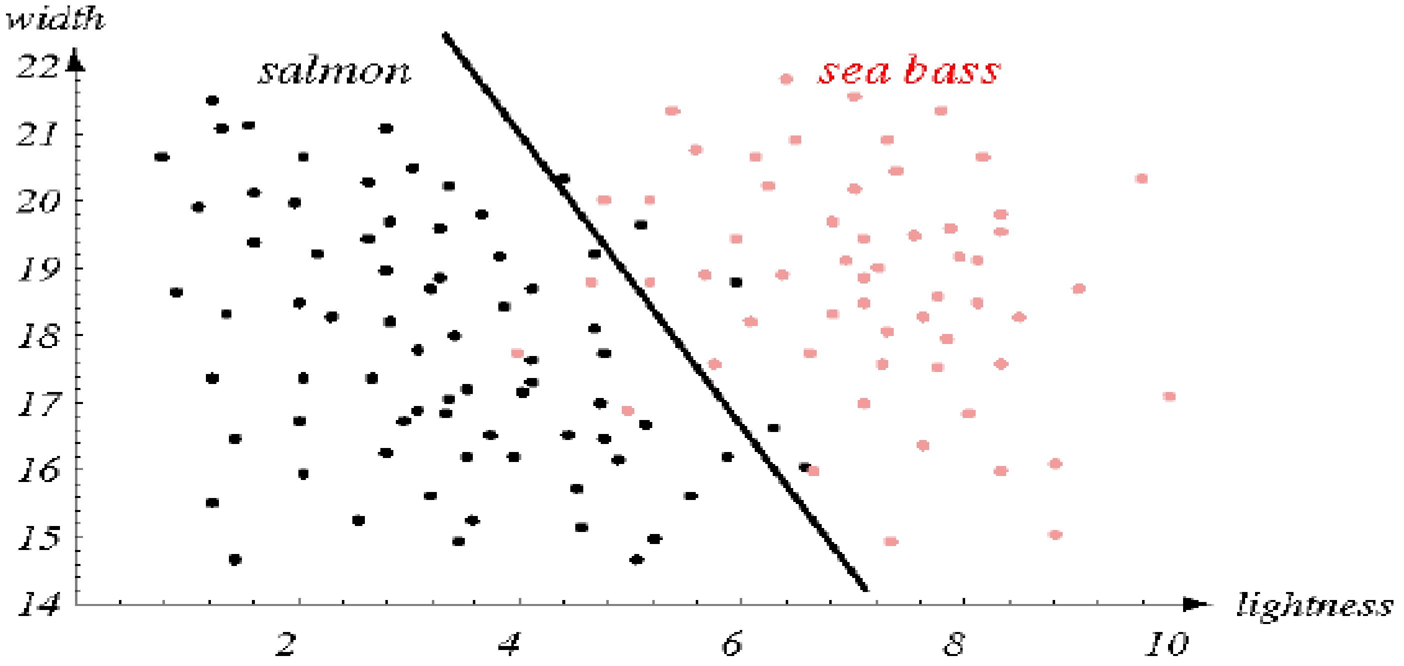
An Example

- Each fish is now a point in two dimension.
  - Lightness (x_1) and width (x_2)



An Example

- Each fish is now a point in two dimension.
  - Lightness (x_1) and width (x_2)



Cost of error

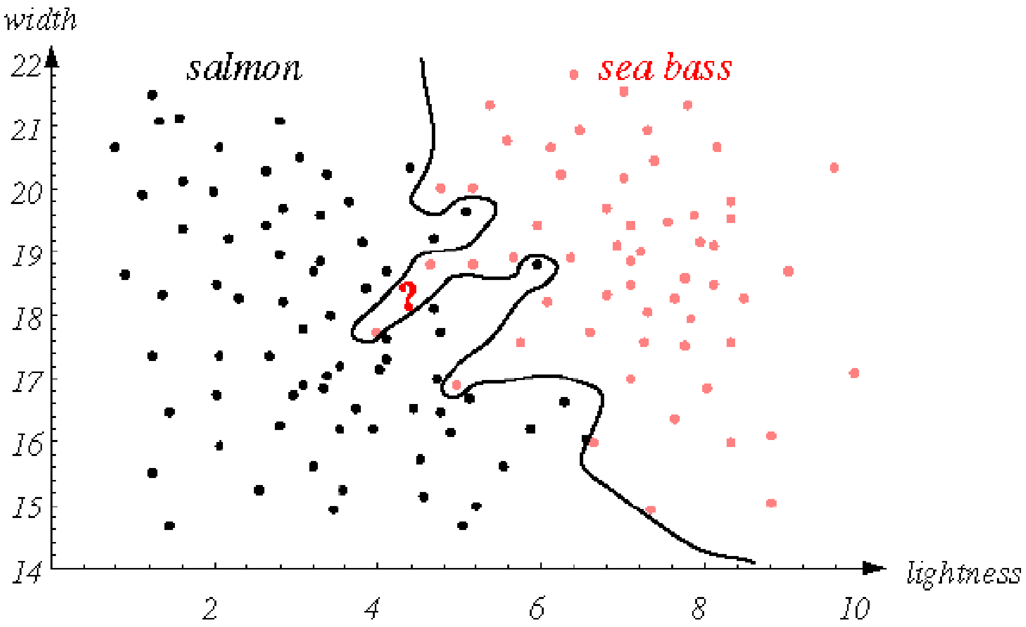
- **Cost of different errors** must be considered when making decisions,
- We try to make a decision rule so as to **minimize** such a cost,
- This is the central task of **decision theory**.

Cost of error

- For example, if the fish packing company knows that:
  - Customers who buy salmon will **object** if they see sea bass in their cans.
  - Customers who buy sea bass will **not be unhappy** if they occasionally see some expensive salmon in their cans.

Decision boundaries

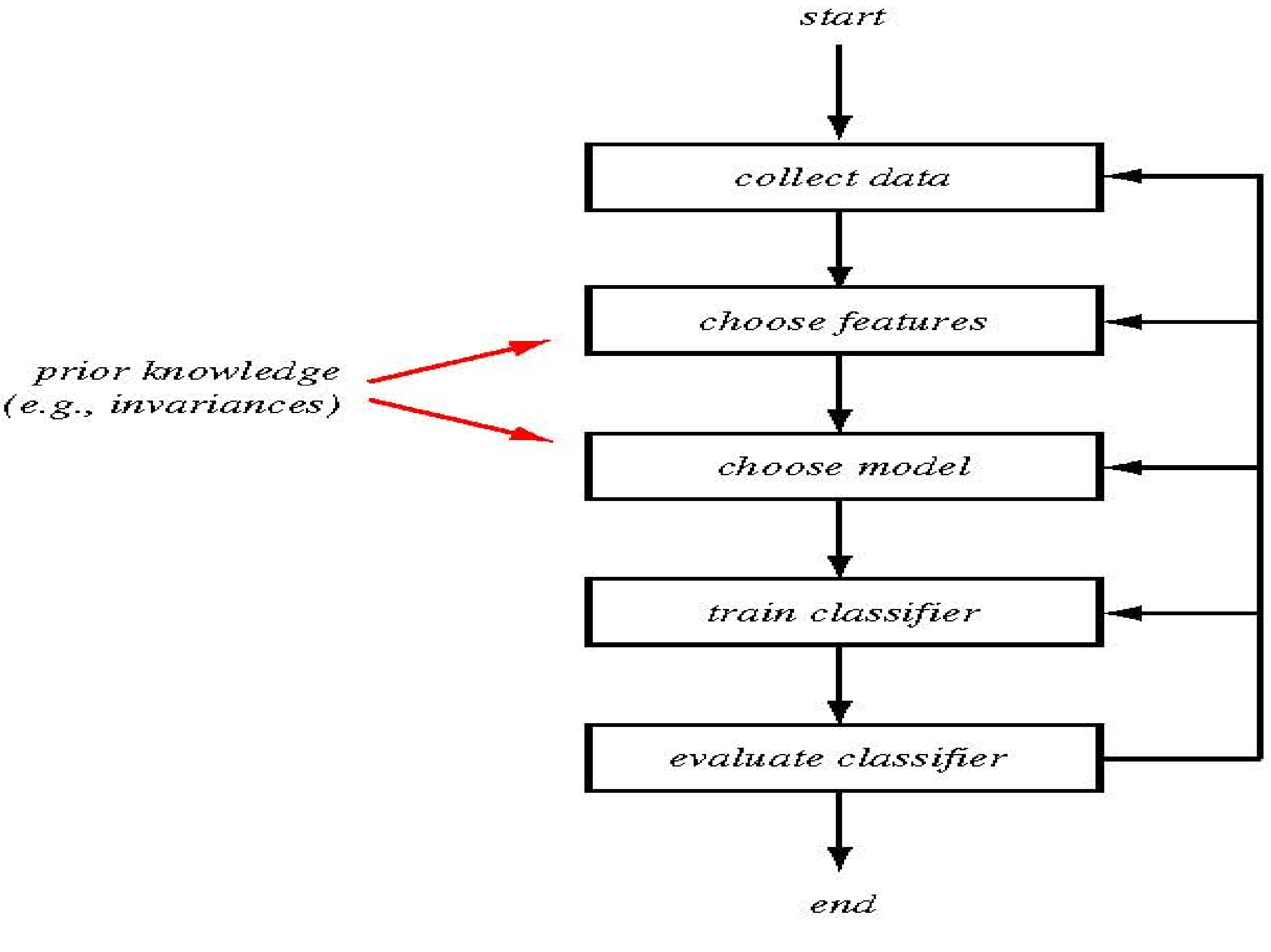
- We can perform better if we use more complex decision boundaries.



Decision boundaries

- There is a trade of between complexity of the decision rules and their performances to unknown samples.
- **Generalization:** The ability of the classifier to produce correct results on *novel* patterns.
- Simplify the decision boundary!

The design cycle



The design cycle

- Collect data:
  - Collect train and test data
- Choose features:
  - Domain dependence and prior information,
  - Computational cost and feasibility,
  - Discriminative features,
  - Invariant features with respect to translation, rotation and scale,
  - Robust features with respect to occlusion, distortion, deformation, and variations in environment.

- Feature Selection & Extraction
- Selection v extraction
- How many and which subset of features to use in constructing the decision boundary?
- Some features may be redundant
- Curse of dimensionality—Error rate may in fact increase with too many features in the case of small number of training samples

Unsupervised Learning

- Definition of Unsupervised Learning:

  Learning useful structure *without* labeled classes, optimization criterion, feedback signal, or any other information beyond the raw data

Unsupervised Learning

- Examples:
  - Find natural groupings of Xs (X=human languages, stocks, gene sequences, animal species,…)→
    Prelude to discovery of underlying properties
  - Summarize the news for the past month→
    Cluster first, then report centroids.
  - Sequence extrapolation: E.g. Predict cancer incidence next decade; predict rise in antibiotic-resistant bacteria
- Methods
  - Clustering (n-link, k-means, GAC,…)
  - Taxonomy creation (hierarchical clustering)
  - Novelty detection ("meaningful"outliers)
  - Trend detection (extrapolation from multivariate partial derivatives)

Similarity Measures in Data Analysis

- General Assumptions
  - Each data item is a tuple (vector)
  - Values of tuple are nominal, ordinal or numerical
  - Similarity = $(\text{Distance})^{-1}$
- Pure Numerical Tuples
  - $\text{Sim}(d_i,d_j) = \Sigma d_{i,k} d_{j,k}$
  - $\text{sim}\ (d_i,d_j) = \cos(d_i d_j)$
  - …and many more (slide after next)

Similarity Measures in Data Analysis

- For Ordinal Values
  - E.g. "small," "medium," "large," "X-large"
  - Convert to numerical assuming constant Δ…on a normalized [0,1] scale, where: max(v)=1, min(v)=0, others interpolate
  - E.g. "small"=0, "medium"=0.33, etc.
  - Then, use numerical similarity measures
  - Or, use similarity matrix (see next slide)

Similarity Measures (cont.)

- For Nominal Values
  - E.g. "Boston", "LA", "Pittsburgh", or "male", "female", or "diffuse", "globular", "spiral", "pinwheel"
  - Binary rule: If $d_{i,k}=d_{j,k}$, then sim=1, else 0
  - Use underlying sematic property: E.g. $Sim(Boston, LA)=\alpha dist(Boston, LA)^{-1}$, or $Sim(Boston, LA)=\alpha(|size(Boston) – size(LA)|)^{-1}$
  - Use similarity Matrix

Similarity Matrix

	tiny	little	small	medium	large	huge
tiny	1.0	0.8	0.7	0.5	0.2	0.0
little	1.0	0.9	0.7	0.3	0.1	
small			1.0	0.7	0.3	0.2
medium				1.0	0.5	0.3
large					1.0	0.8
huge						1.0

- – Diagonal must be 1.0
- – Monotonicity property must hold
- – Triangle inequality must hold
- – Transitive property need *not* hold

References

- Srihari, S.N., Covindaraju, Pattern recognition, Chapman &Hall, London, 1034-1041, 1993,
- Sergios Theodoridis, Konstantinos Koutroumbas , pattern recognition , Pattern Recognition ,Elsevier(USA)) ,1982
- R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, New York: John Wiley, 2001,