Overview of Machine Learning: Algorithms and Applications

Nathalie Japkowicz
School of Information Technology and Engineering
University of Ottawa

Overview I

- Machine Learning and Data Mining include a large number of are general-purpose methods that have been applied with great success to a many domains, including mechanical diagnosis, satellite image screening, credit-card fraud detection, medical diagnosis, marketing, loan application screening, electric load forecasting, and so on.

Overview II

- These approaches rely on different technologies including:
  - Decision Trees, Neural Networks, Bayesian Learning
  - Instance-Based Learning (e.g., k- Nearest Neighbours)
  - Rule Induction
  - Clustering / Unsupervised Learning
  - Support Vector Machines, etc.
- The more advanced approaches combine the above techniques using sophisticated combination schemes such as:
  - Bagging, Boosting, Stacking
  - Random Forests, etc.

Overview III

- In addition to designing algorithms, researchers in the field have been concerned with issues surrounding these algorithms, such as:
  - Feature Selection / Feature Construction
  - Missing / Unreliable Attributes
  - Cost-Sensitive learning
  - Distributional Skewness (the Class Imbalance Problem)
  - Learning from massive Data Sets
  - Data Visualization
  - Evaluation in Machine learning
  - Incorporating Domain Knowledge, etc.

Purpose of the Lecture

- To present an overview of some of these techniques.
- To demonstrate, through a few examples, their applicability to a large range of security domains.
- To illustrate what research in Machine Learning tries to achieve and how it does so.

Preliminaries

Useful Reference – Demo

- For a quick introduction to the field and a quick assessment of its usefulness to you, you can download a free software toolbox that implements the major machine learning algorithms:

  WEKA

- http://www.cs.waikato.ac.nz/ml/weka/

- Accompanying text book:

  Data Mining: Practical Machine Learning Tools and Techniques, by Ian Witten and Eibe Frank, Morgan Kaufmann, 2005.

Organization of the Talk

- Definition of Machine Learning: Supervised/ Unsupervised Learning
- Why Machine Learning?
- A Taxonomy of Machine Learning Methods
- Two Instances of Machine Learning Algorithms: Decision Trees, Neural Networks
- Two instances of Combination Schemes: Bagging, Boosting
- Three Applications:
  - Event Characterization for Radioxenon Monitoring
  - Detection of Helicopter Gearbox failures
  - Discrimination between Earthquakes and Nuclear Explosions.
  - Network Security

Supervised Learning: Definition

Given a sequence of input/output pairs of the form *<xi, yi>*, where *xi* is a possible input, and *yi* is the output associated with *xi:*

Learn a function *f* such that:

- *f(xi)=yi* for all *i's*,
- *f* makes a good guess for the outputs of inputs that it has not previously seen.

[If *f* has only 2 possible outputs, *f* is called a *concept* and learning is called *concept-learning*.]

Supervised Learning: Example

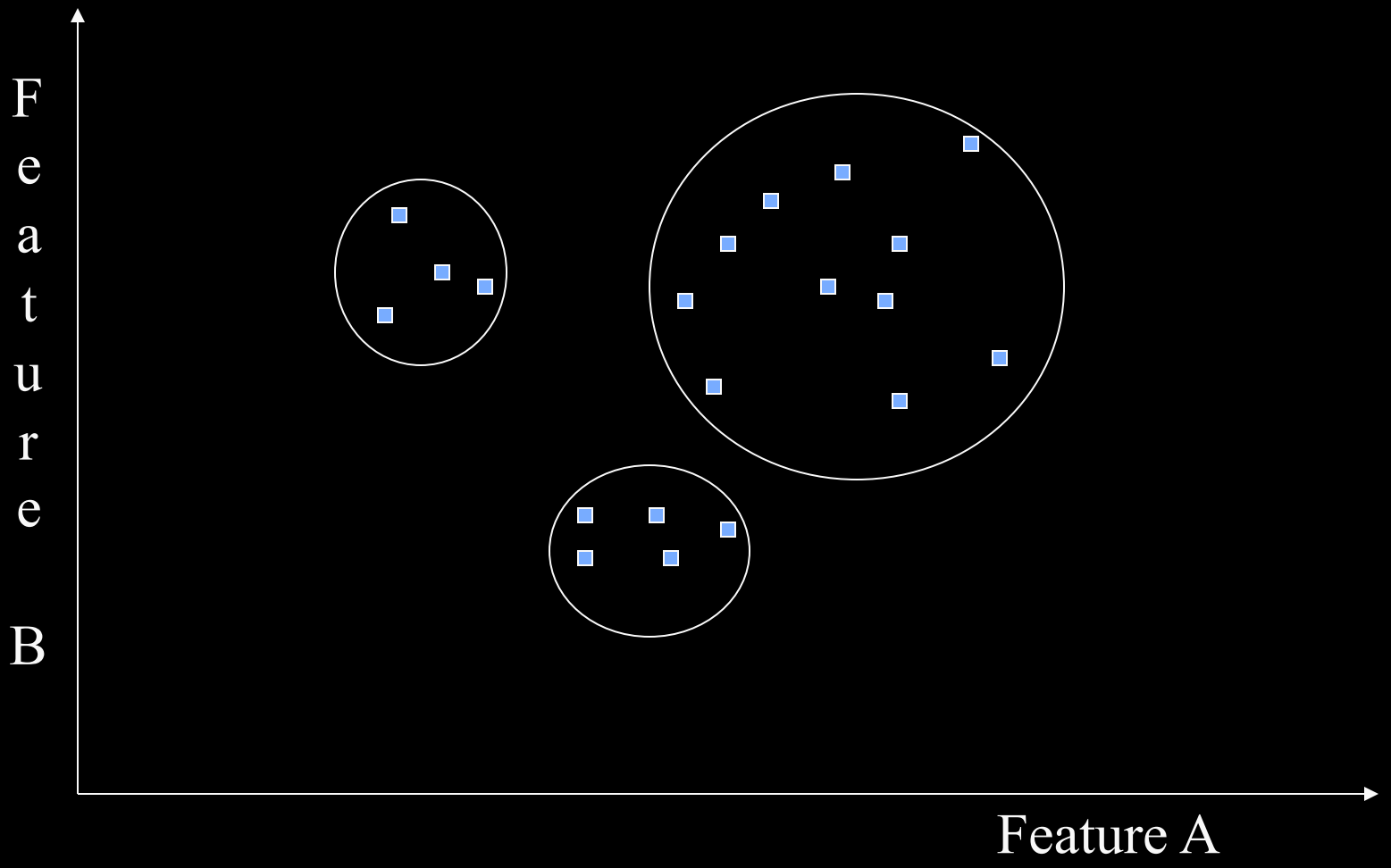
Patient	*Attributes*				*Class*
	Temperature	*Cough*	*Sore Throat*	*Sinus Pain*	
1	37	yes	no	no	no flu
2	39	no	yes	yes	flu
3	38.4	no	no	no	no flu
4	36.8	no	yes	no	no flu
5	38.5	yes	no	yes	flu
6	39.2	no	no	yes	flu

Goal: Learn how to predict whether a new patient with a given set of symptoms does or does not have the flu.

Unsupervised Learning: Definition

- While Supervised Learning considers the input/output pairs of the form *<xi, yi>*, Unsupervised Learning focuses on the input only: *xi.* It has no knowledge of the output, *yi.*
- Unsupervised Learning attempts to group together (or cluster) similar *xis.*
- Different similarity measures can be used as well as different strategies for building the clusters.

Unsupervised Learning: An Example Fitting a Mixture of Gaussians

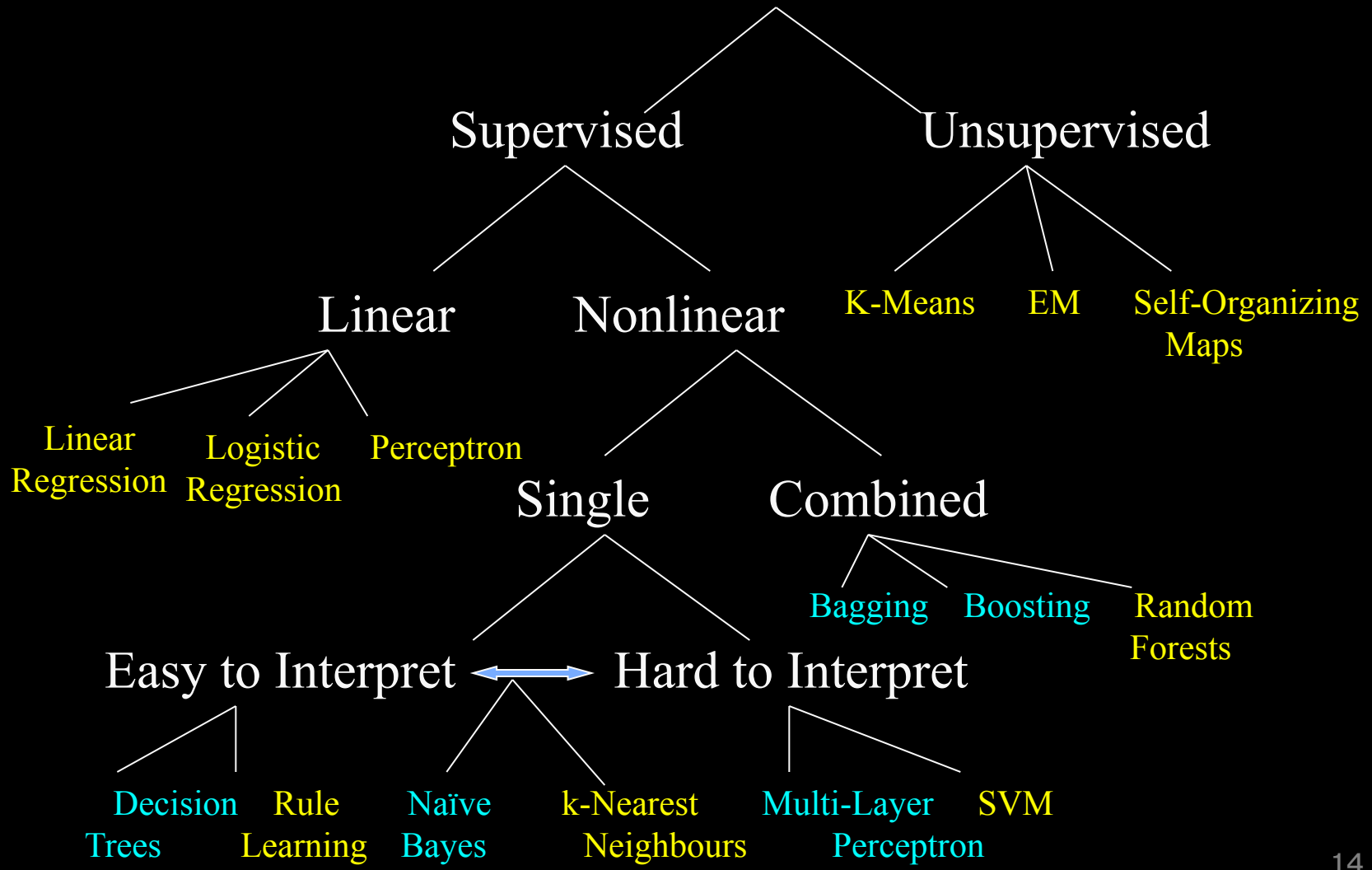


Why Machine Learning?

- Machine Learning Systems learn from data samples of solved cases.
- They do not require any expert knowledge, since they infer such knowledge directly from the data.
- They are useful in professional fields in which expertise is scarce and the codification of knowledge is limited.
- They are useful in domains where good tests and measurements are available, but methods of applying this information is insufficiently understood or systematized.
- They are useful in domains in which the information needs to be constantly updated, in order to maintain the system in routine use at high levels of performance

[From *Computer Systems that Learn*, Weiss & Kulikowski, Morgan Kaufmann, 1990]

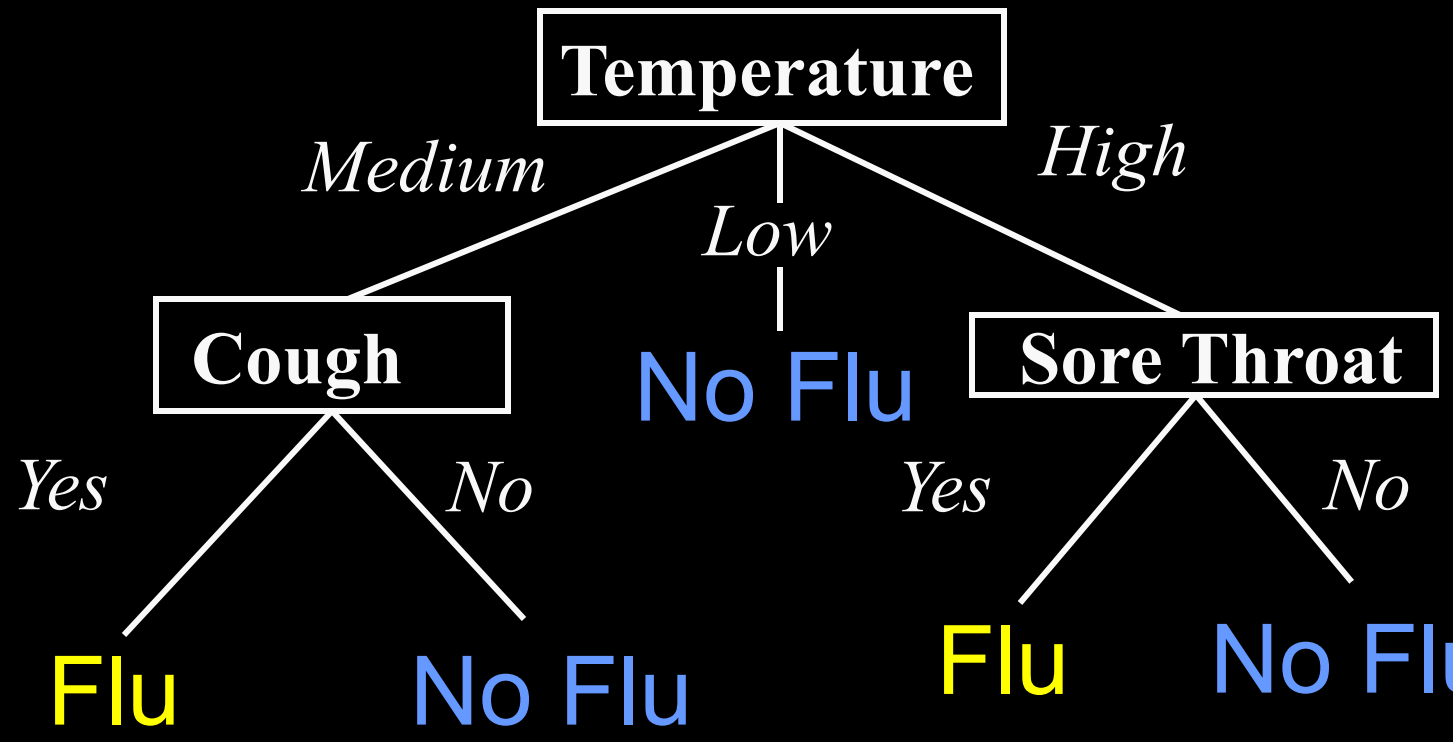
A Taxonomy of Machine Learning Techniques: Highlight on Important Approaches



Description of Two Common Classifiers:

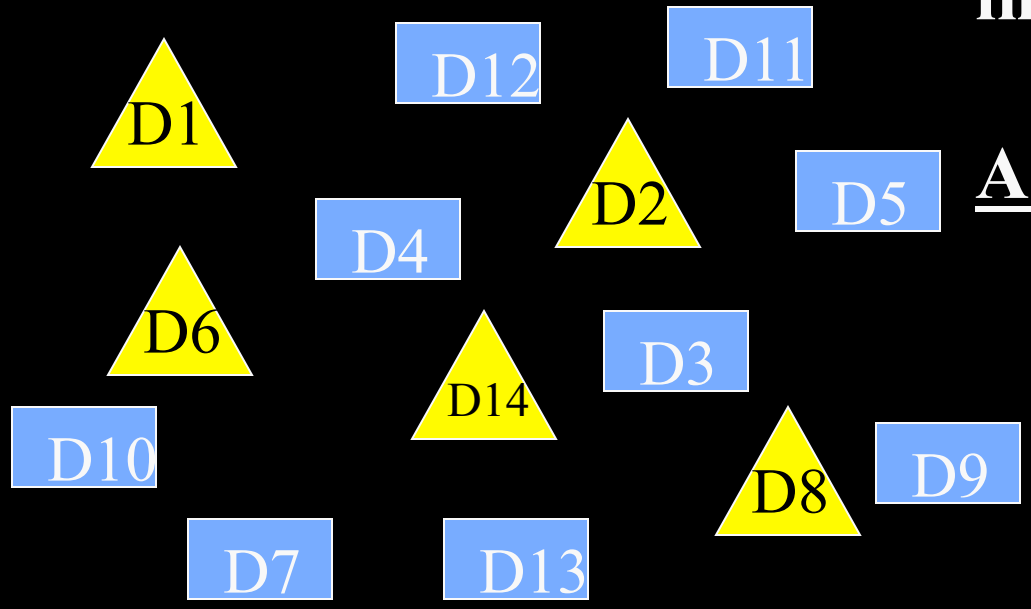
- Decision Trees
- Neural Networks

Decision Trees: A transparent Approach



A Decision Tree for the *Flu* Concept

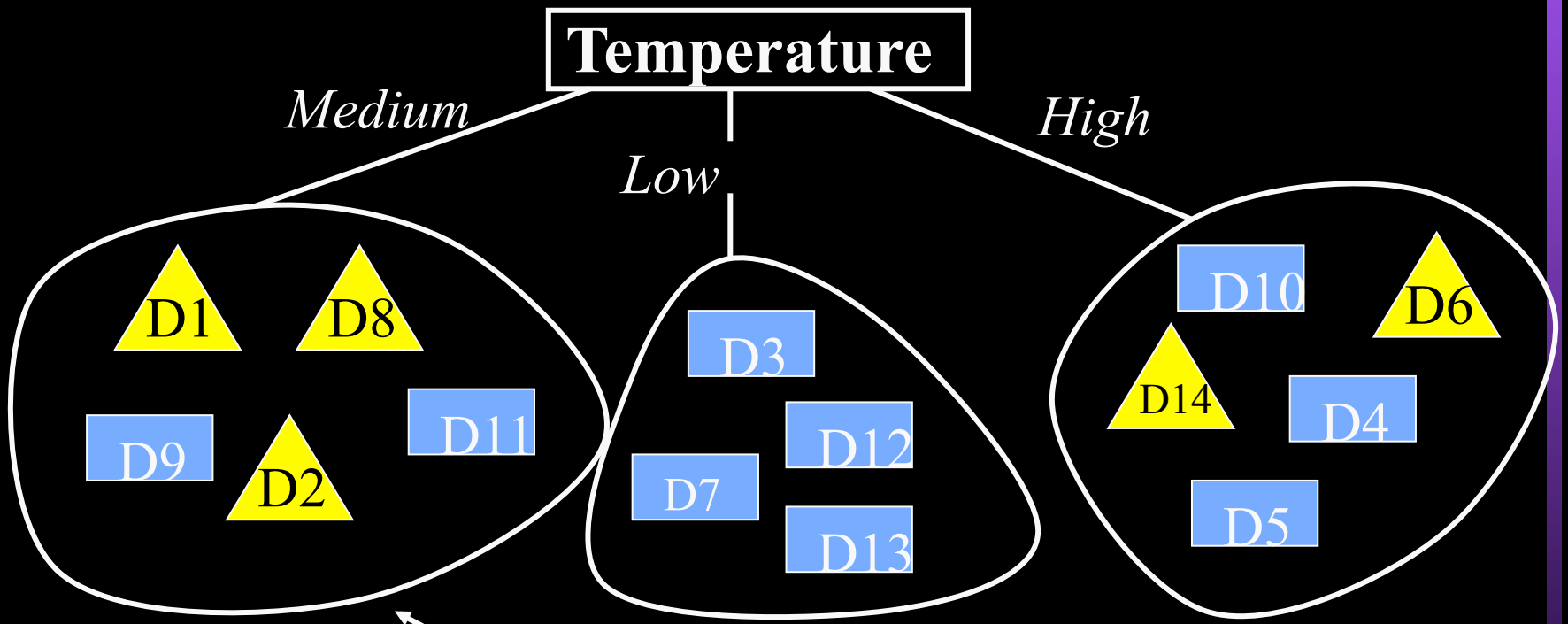
Construction of Decision Trees I



What is the most informative attribute?

<u>Assume:</u> Temperature

Construction of Decision Trees II



What are the most informative attributes?

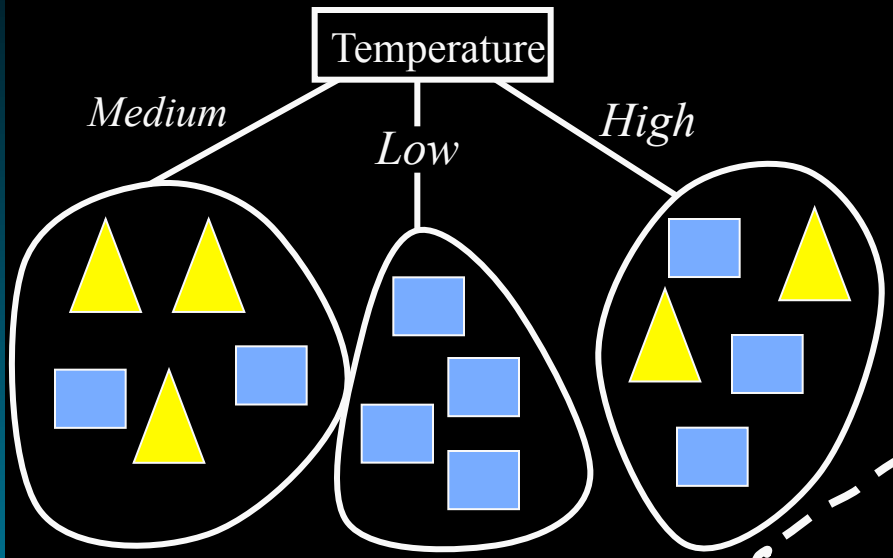
Assume:

Cough and **Sore Throat**

Construction of Decision Trees III

- The informativeness of an attribute is an information-theoretic measure that corresponds to the attribute that produces the purest children nodes.
- This is done by minimizing the measure of entropy in the trees that the attribute split generates.
- The entropy and information are linked in the following way: The more there is entropy in a set S, the more information is necessary in order to guess correctly an element of this set.
- Info[x,y] = Entropy[x,y] = - x/(x+y) log x/(x+y)
  - y/(x+y) log y/(x+y)

Construction of Decision Trees IV

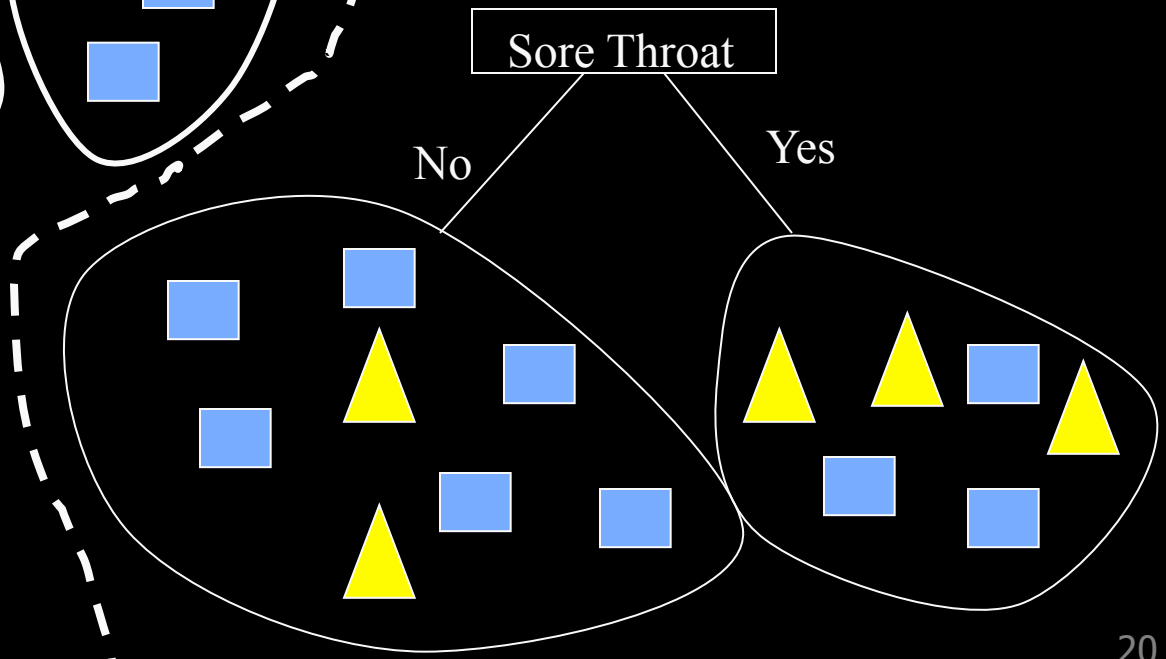


Info[2,3] = .971 bits
Info[4,0]= 0 bits
Info[3,2]= .971 bits

Avg Tree Info = 5/14 *.971
+ 4/14 * 0 + 5/14 * .971
= .693
Prior Info[9,5] = .940
➔ Gain = .940 - .693 = .247

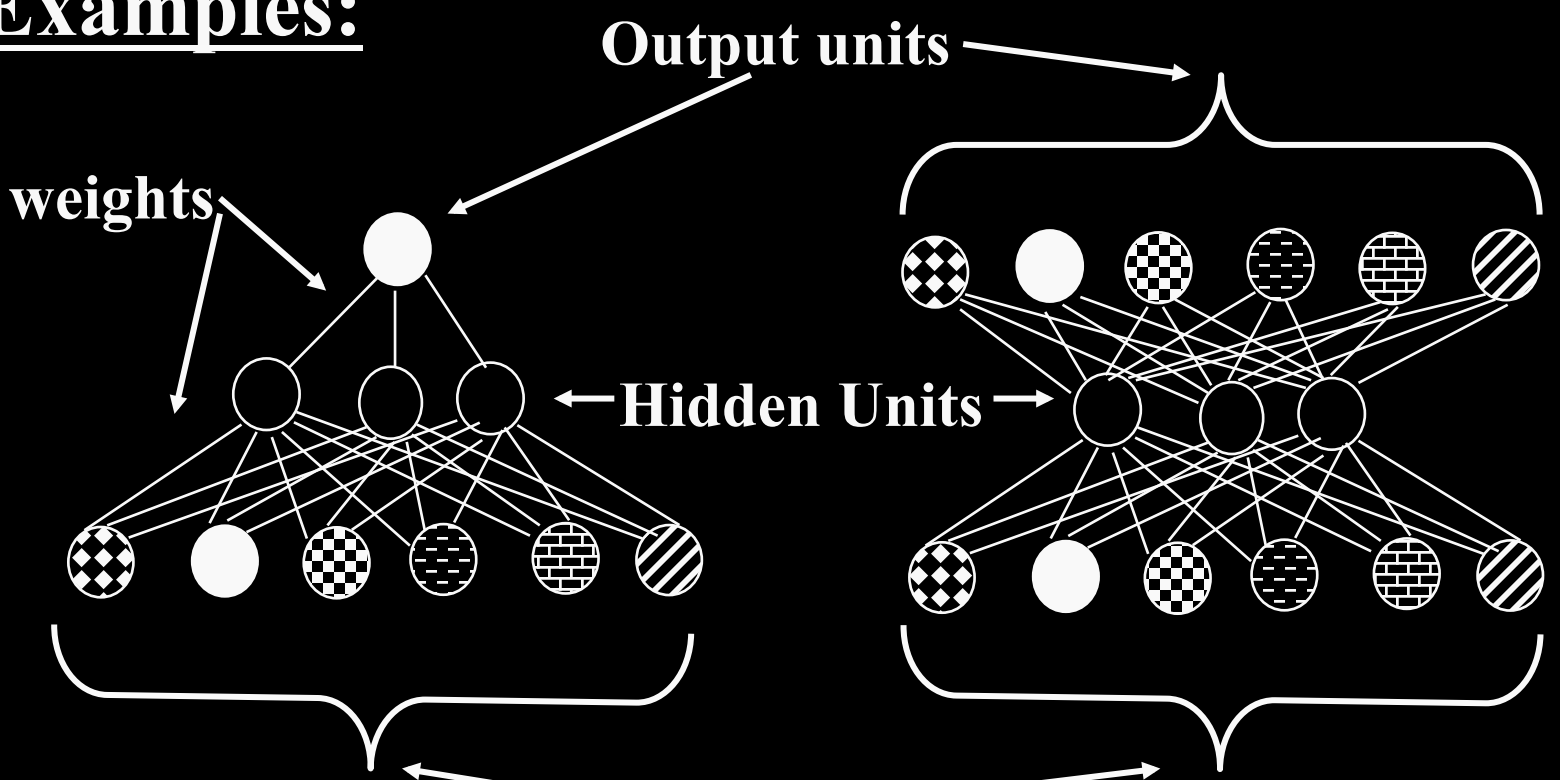
Info[2,6]= .811
Info[3,3]= 1

Avg Tree Info = 8/14 +.811
+ 6/14 * 1 = .892
Prior Info[9,5] = .940
➔ Gain = .940 - .892 = .048



Multi-Layer Perceptrons: An Opaque Approach

Examples:



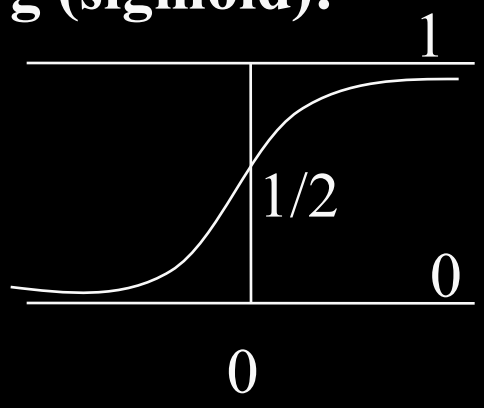
Representation in a Multi-Layer Perceptron

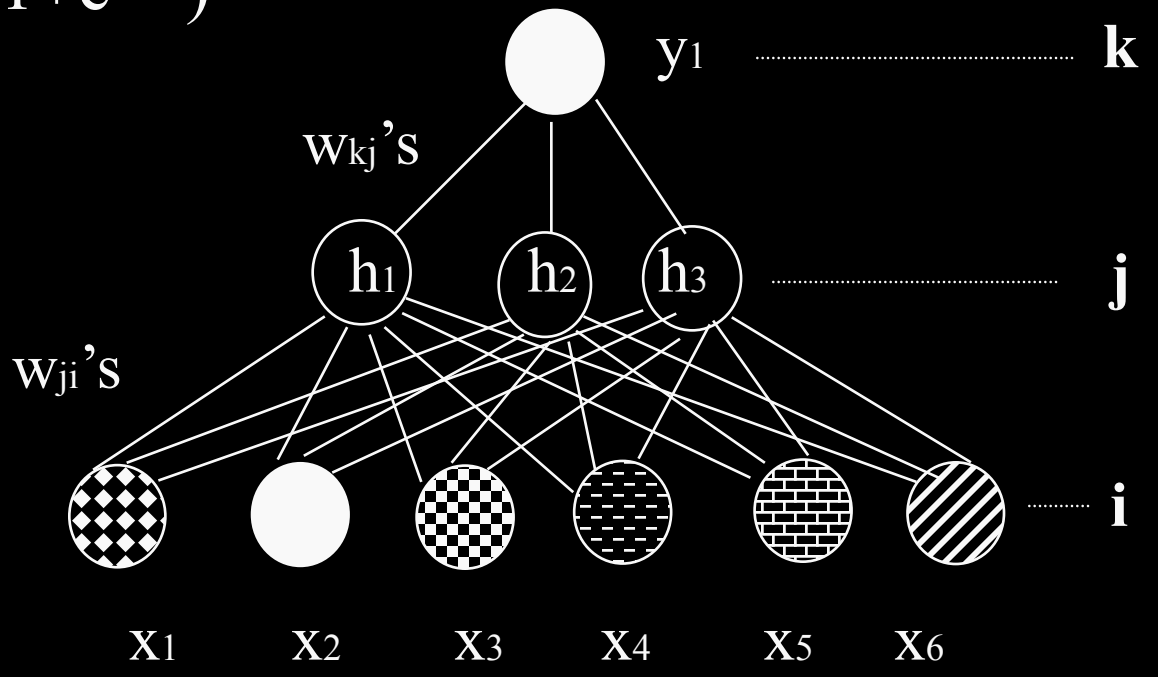
- $h_j = g(\Sigma_i w_{ji}.x_i)$
- $y_k = g(\Sigma_j w_{kj}.h_j)$

where $g(x) = 1/(1+e^{-x})$

Typically, y_1=1 for positive example and y_1=0 for negative example

g (sigmoid):





Learning in a Multi-Layer Perceptron I

- Learning consists of ***searching through the space of all possible matrices of weight values*** for a combination of weights that satisfies a database of positive and negative examples (multi-class as well as regression problems are possible).
- It is an ***optimization problem*** which tries to minimize the sum of square error:

$$E = 1/2 \sum_{n=1}^{N} \sum_{k=1}^{K} [y_k - f_k(x)]$$

where N is the total number of training examples and K, the total number of output units (useful for multiclass problems) and fk is the function implemented by the neural net

Learning in a Multi-Layer Perceptron II

- The optimization problem is solved by searching the space of possible solutions by gradient.
- This consists of taking small steps in the direction that minimize the gradient (or derivative) of the error of the function we are trying to learn.
- When the gradient is zero we have reached a local minimum that we hope is also the global minimum.

Description of Two classifier combination Schemes:

- Bagging
- Boosting

Combining Multiple Models

- The idea is the following: In order to make the outcome of automated classification more reliable, it may be a good idea to combine the decisions of several single classifiers through some sort of voting scheme
- Bagging and Boosting are the two most used combination schemes and they usually yield much improved results over the results of single classifiers
- One disadvantage of these multiple model combinations is that, as in the case of neural Networks, the learned model is hard, if not impossible, to interpret.

Bagging: Bootstrap Aggregating

Bagging Algorithm

model generation

Let n be the number of instances in the training data.
For each of t iterations:
  Sample n instances with replacement from training data.
  Apply the learning algorithm to the sample.
  Store the resulting model.

classification

For each of the t models:
  Predict class of instance using model.
Return class that has been predicted most often.

Figure 7.7 Algorithm for bagging.

Idea: perturb the composition of the data sets from which the classifier is trained. Learn a classifier from each different data set. Let these classifiers vote. ➔ This procedure reduces the portion of the performance error caused by variance in the training set.

Boosting

Boosting Algorithm

model generation

```
Assign equal weight to each training instance.
For each of t iterations:
  Apply learning algorithm to weighted dataset and store
     resulting model.
  Compute error e of model on weighted dataset and store error.
  If e equal to zero, or e greater or equal to 0.5:
     Terminate model generation.
  For each instance in dataset:
     If instance classified correctly by model:
        Multiply weight of instance by e / (1 - e).
  Normalize weight of all instances.
```

classification

```
Assign weight of zero to all classes.
For each of the t (or less) models:
  Add -log(e / (1 - e)) to weight of class predicted by model.
Return class with highest weight.
```

Figure 7.8 Algorithm for boosting.

Decrease the weight of the right answers ⇔ Increase the weight of errors

Idea: To build models that complement each other. A first classifier is built. Its errors are given a higher weight than its right answers so that the next classifier being built focuses on these errors, etc…

Machine Learning Applications

Application I: Event Characterization for Radioxenon Monitoring

- It has been shown that methods currently used for particulate monitoring to identify anomalous radionuclide observations, possibly indicative of a nuclear explosion, are not suitable for radioxenon monitoring.
- Unlike particulate monitoring, there is a ubiquitous radioxenon background.
- Distinguishing radioxenon of a nuclear explosion origin from routine anthropogenic radioxenon releases is problematic.
- The goal of these preliminary experiments is to verify whether machine learning techniques can be useful for such a task.

Application I: Event Characterization for Radioxenon Monitoring

- We were given three datasets from the Radiation Protection Bureau branch of Health Canada.
- The first data set describes the background concentration of Xenon isotopes, i.e., Xe-131m, Xe-133, Xe-133m, and Xe-135, in Ottawa, under normal conditions.
- The second data set describes the concentration levels of these Xenon isotopes that would be seen in Ottawa if a 90mBq/m3 explosion had taken place and 7 days had elapsed.
- The third data set describes the concentration levels of these Xenon isotopes that would be seen in Ottawa if a 1mBq/m3 explosion had taken place and 7 days had elapsed.

Application I: Event Characterization for Radioxenon Monitoring

- We applied the following classifiers to this problem:
  - Decision Trees (J48)
  - Multi-layer Perceptrons (MLP)
  - Naive Bayes (NB)
  - Support Vector Machine (SVM), and
  - Nearest Neighbours (kNN)

Application I: Event Characterization for Radioxenon Monitoring

Results in the 90 mBq/m3 case

Table 1: 90mBq/m3.

classifiers	Accuracy	TP	FP	Precision	Recall	F-measure	ROC	Classes
J48	**96.0884 %**	0.925	0.003	**0.996**	0.925	**0.959**	**0.992**	N
		0.997	0.075	0.93	**0.997**	**0.962**	**0.992**	A
NB	95.068 %	0.901	0	1	0.901	0.948	0.968	N
		1	0.099	0.91	1	0.953	0.968	A
MLP	89.6259 %	0.878	0.085	0.912	0.878	0.894	0.945	N
		0.915	0.122	0.882	0.915	0.898	0.945	A
SVM	87.415 %	**0.949**	0.201	0.825	**0.949**	0.883	0.874	N
		0.799	0.051	**0.94**	0.799	0.864	0.874	A
kNN	93.7075 %	0.922	0.048	0.951	0.922	0.936	0.989	N
		0.952	0.078	0.924	0.952	0.938	0.989	A

The problem is quite easy:
Simple classifiers: Decision Trees, Naïve Bayes and k-Nearest Neighbours obtain good results

Application I: Event Characterization for Radioxenon Monitoring

Results in the 1 mBq/m3 case

Table 2. 1 mBq/m3

classifiers	Accuracy	TP	FP	Precision	Recall	F-measure	ROC	Classes
J48	49.3197 %	**0.592**	0.605	0.494	**0.592**	**0.539**	0.492	N
		0.395	0.408	0.492	0.395	0.438	0.492	A
NB	49.3197 %	0.395	0.408	0.492	0.395	0.438	0.493	N
		0.592	0.605	0.494	0.592	0.539	0.493	A
MLP	**50.3401 %**	0.303	0.296	**0.506**	0.303	0.379	**0.501**	N
		0.704	0.697	**0.502**	**0.704**	0.586	**0.501**	A
SVM	49.6599 %	0.541	0.548	0.497	0.541	0.518	0.497	N
		0.452	0.459	0.496	0.452	0.473	0.497	A
kNN	40.4762 %	0.551	0.741	0.426	0.551	0.481	0.403	N
		0.259	0.449	0.365	0.259	0.303	0.403	A

The problem does not seem solvable:
Further research on Feature Selection techniques,
including our own method boosted accuracy only by 4%.

Application I: Event Characterization for Radioxenon Monitoring

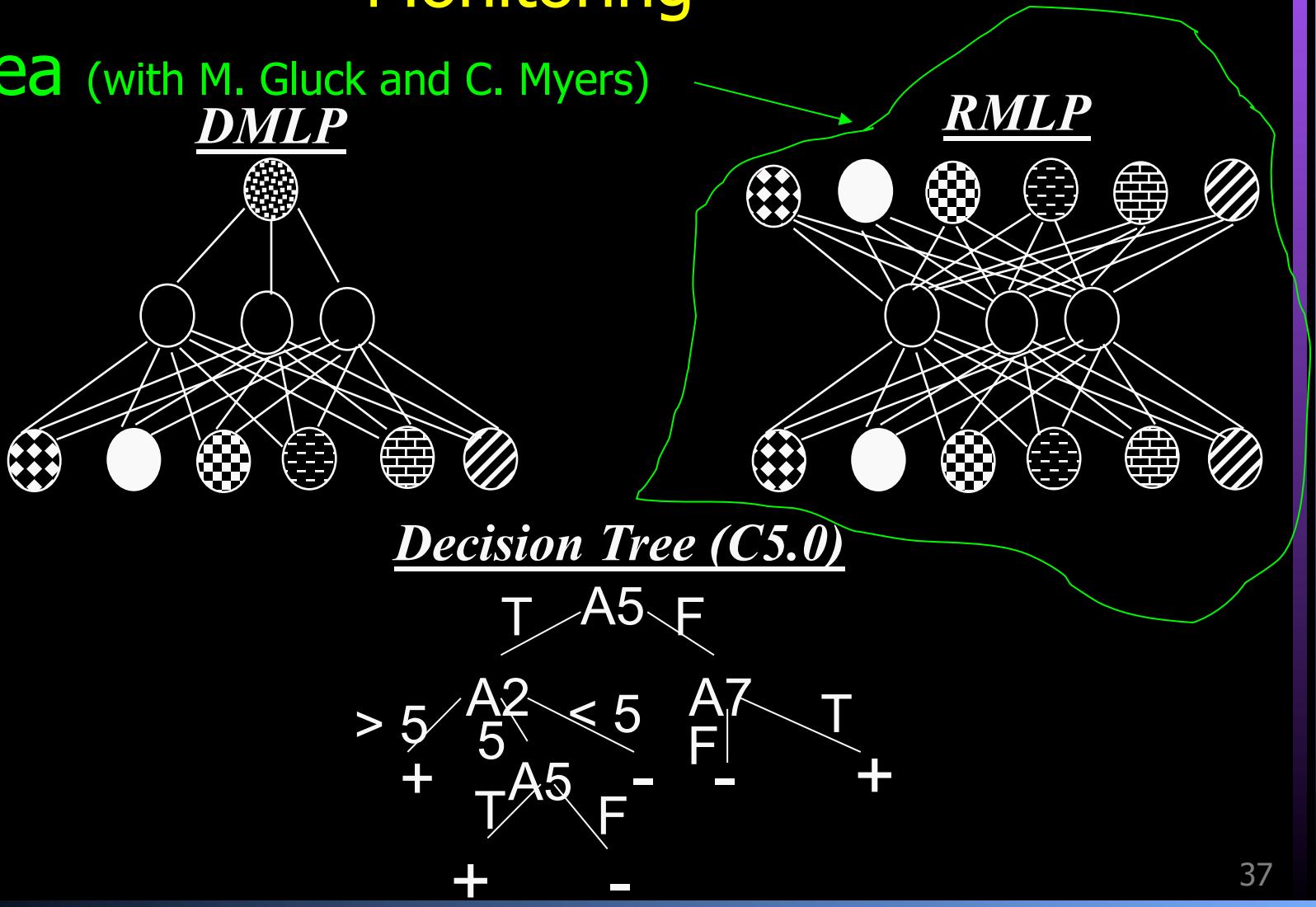
- Machine Learning is useful in this application as its techniques can help us simulate a realistic data set of radioxenon observations arising from nuclear explosions.
- Such a database would be composed of data of real routine anthropogenic radioxenon observations plus information known of radioxenon released from past nuclear weapons tests.
- The data thus generated would be used to determine the best path of research into event characterization methods for the Comprehensive Nuclear-Test-Ban Treaty Organization.

Application II: Helicopter Gearbox Monitoring

- **Practical Motivation:** Detect CH-46 Helicopter Gearboxes failures on flight, in order to land on time to avoid a crash.
- **Approach:** Discriminate between the vibration pattern of Healthy and Damaged CH-46 Helicopter Gearboxes, by learning how to recognize healthy gearboxes.
- The data was provided by the U.S. Navy. It had the particularity of containing many instances of healthy gearboxes and few instances of damaged ones. (The class imbalance problem) → We devised a single-class classification scheme.

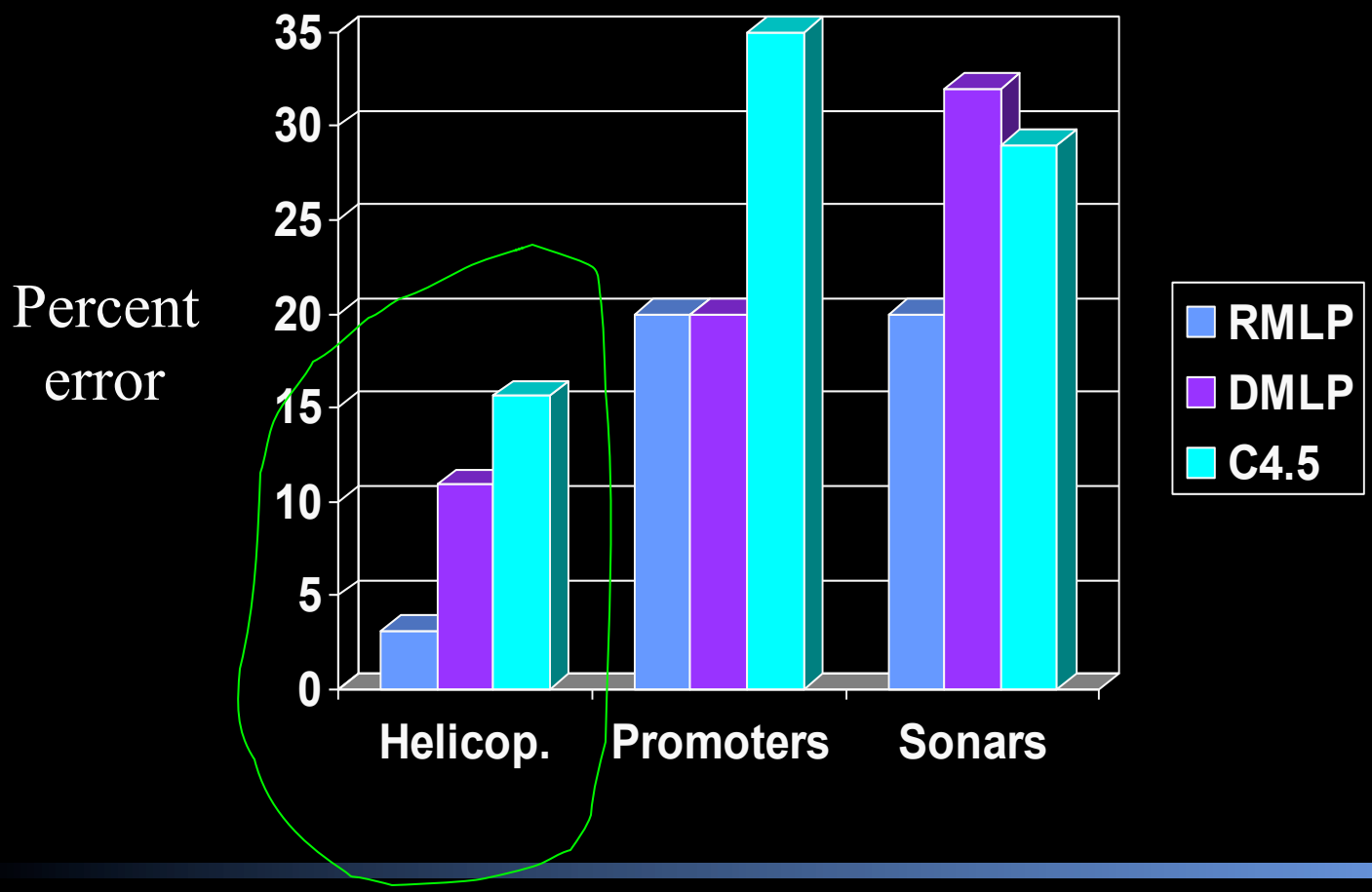
Application II: Helicopter Gearbox Monitoring

- Idea (with M. Gluck and C. Myers)



Application II: Helicopter Gearbox Monitoring

Results (errors)



Application III: Nuclear Explosions versus Earthquakes

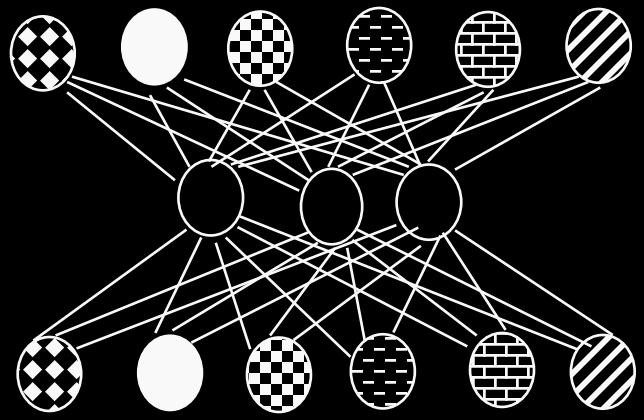
- **Practical Motivation:** To ensure that the Comprehensive Test Ban Treaty is globally respected.
- **Approach:** To monitor vibration patterns in the ground. World-wide Earthquakes and Nuclear explosions can be detected locally. Apply Machine Learning Technique to discriminate between the two different kinds of signals.
- **Challenge:**
  - Earthquakes and Nuclear explosions have similar patterns. Very sensitive learning techniques must be designed for the task.
  - There are very few instances of nuclear explosions (only very few weapons' tests) ➔ Huge class imbalance
- **Data Source**
  - Little Skull Mountain Earthquakes + Large aftershocks (29/06/92)
  - Lawrence Livermore Labs Testing site for Nuclear explosions (1978 to 1992)

Application III: Nuclear Explosions versus Earthquakes

- Idea (with Todd Eavis): **XMLP**

-.9 -.4 -.2 -.7 -.3 -.6
.2 .3 .6 .1 .5 .2

→ -1.9 -1.4 -1.2 -1.7 -1.3 -1.6
1.2 1.3 1.6 1.1 1.5 1.2



.2 .3 .6 .1 .5 .2 Pos
.9 .4 .2 .7 .3 .6 Neg

Note: the original idea did not show very good discrimination effects, so We exaggerated the difference between Pos and Neg data by adding +1 toeach node in the case of a positive example and -1 to each node in the case of a negative example.

Application III: Nuclear Explosions versus Earthquakes

- Results (errors)

Negative Samples	MLP	XMLP
1	42.1%	**37.8%**
2	33.6%	**29.4%**
5	19.9%	**17.8%**
10	**14.7%**	18.8%

Application IV: Network Event Correlation

- **Practical Motivation:** Computer Networks are more and more often attacked. Intrusion Detection Systems are capable of detecting attacks. However, they issue a very large number of false alarms. We want to learn how to correlate similar types of attacks in order to allow a human operator to process groups of alarms together rather than individual alarms one by one.
- **Idea:** (with M. Dondo and R. Smith) Use RMLP as a soft-clustering system in order to suggest potential groupings.

Application IV: Network Event Correlation

Results

Conclusions

- Machine Learning has proven useful in many areas.
- There are free tools available on the Web that are easy to use and that do not require much prior knowledge of Machine Learning. The most notable/used suite of tools is called WEKA, http://www.cs.waikato.ac.nz/ml/weka/
- There is a lot of ongoing research in Machine Learning that develops new approaches for different types of data challenges.
- Researchers in Machine learning/Data mining (including myself!) generally welcome the opportunity to try their ideas on new kinds of data sets. Collaborations can benefit both the machine learning and applied communities.