COMP 652: Machine Learning

Instructor: Prof. Ted Perkins

E-mail: perkins@mcb.mcgill.ca

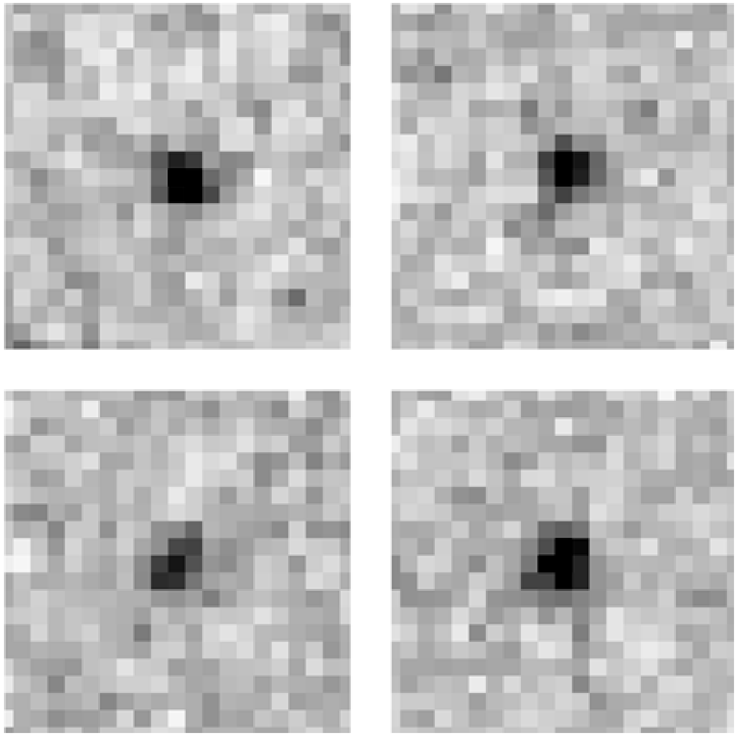
Web:
http://www.mcb.mcgill.ca/~perkins/COMP652_Fall2008/index.html

Today

1. Machine learning: Examples and motivation
2. Administrative: syllabus
3. Types of machine learning: supervised, unsupervised, reinforcement
4. Supervised learning intro (with examples)

Categorizing faint objects in a Sky Survey (Usama Fayyad)

- B&W digital images of virtually entire sky taken at high resolution
- Astronomers could not examine each image in detail, and catalogue the objects observed
- Machine learning was used to automate the categorization and cataloguing of $\geq 10^9$ faint objects



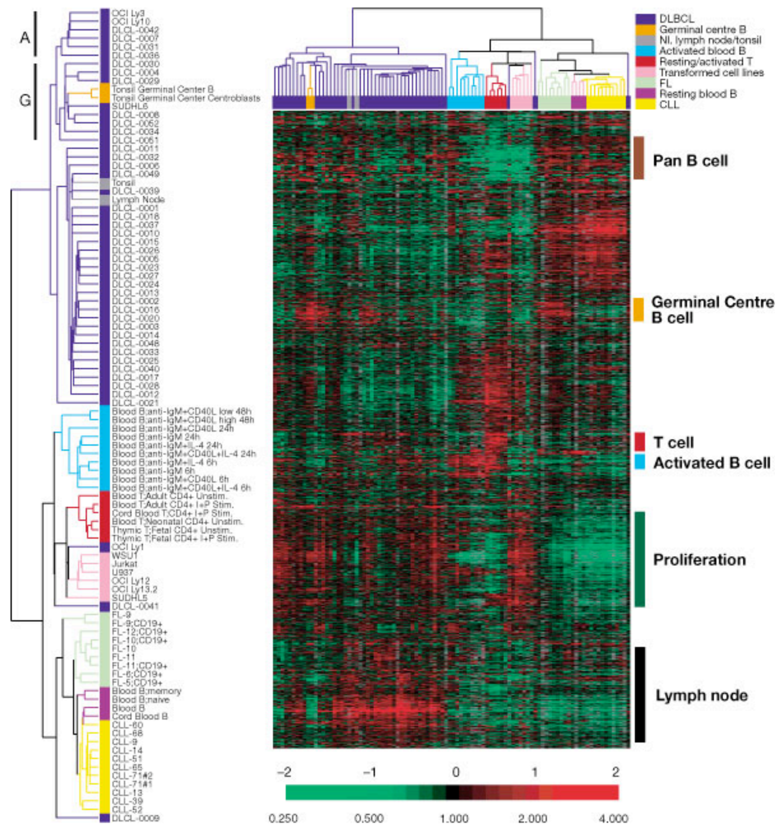
Face detection and recognition

How would you write a computer program to:

- ☐ Detect faces in a scene?
- ☐ Recognize the face of a particular person?

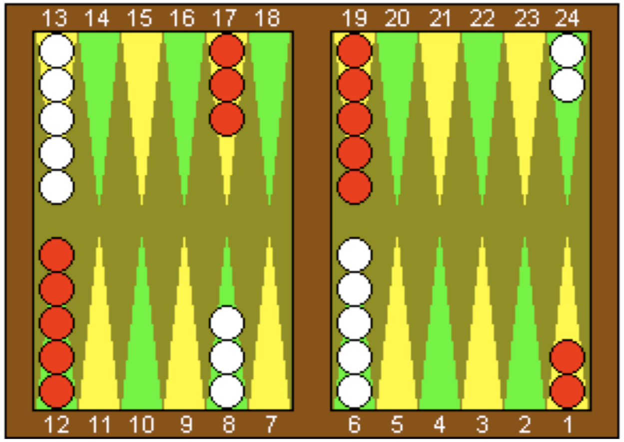
Oncology (Alizadeh et al.)

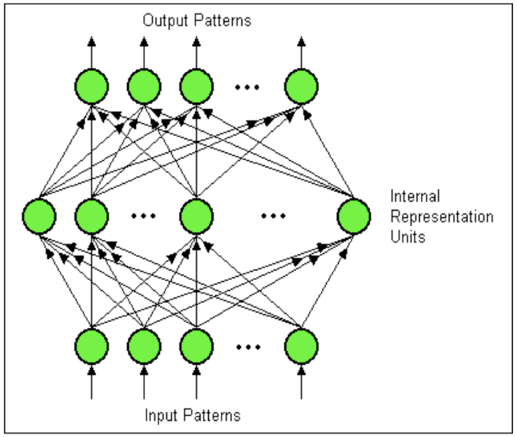
- Activity levels of all (≈25,000) genes were measured in lymphoma patients
- Cluster analysis determined three different subtypes (where only two were known before), having different clinical outcomes



Backgammon (Tesauro)

- Starting with expert knowledge, the TD-gammon program learned to play backgammon by playing millions of games against itself . . .
- And became (arguably) the best player in the world!





And many more...

- Bioinformatics: sequence alignment, analyzing microarray data, information integration, ...
- Computer vision: object recognition, tracking, segmentation, active vision, ...
- Robotics: state estimation, map building, decision making
- Graphics: building realistic simulations
- Speech: recognition, speaker identification
- Financial analysis: option pricing, portfolio allocation
- E-commerce: automated trading agents, data mining, spam, ...
- Medicine: diagnosis, treatment, drug design,...
- Computer games: building adaptive opponents
- Multimedia: retrieval across diverse databases

What makes a good machine learning problem?

- Problems involving very large datasets
- Problems involving complex relationships between variables
- Problems involving numerical reasoning
- Problems for which expert opinions are not readily available / cost effective / rapid enough
- ...

Basically, anything that could be done by computer (in principle), but which is hard to program directly.

Types of machine learning problems

We'll discuss three major types of problems:

1. Supervised learning
   - Given data comprising input-output pairs
   - Create an output-predictor for new inputs
2. Unsupervised learning
   - Given data objects, look for "patterns": clusters, variable relationships, ...
   - Or, "compress" data in some sense
3. Reinforcement learning
   - An AI interacts with environment, receiving rewards and punishment
   - Must learn to behave optimally

Machine learning algorithms / representations / topics

- Linear, polynomial and logistic regression and/or classification
- Artificial neural networks
- Decision and regression trees
- "Nonparameteric" or instance-based methods
- Computational learning theory
- Ensemble methods
- Value function approximation
- Flat and hierarchical clustering
- Dimensionality reduction (PCA, ICA, ...)

Syllabus!

Supervised learning

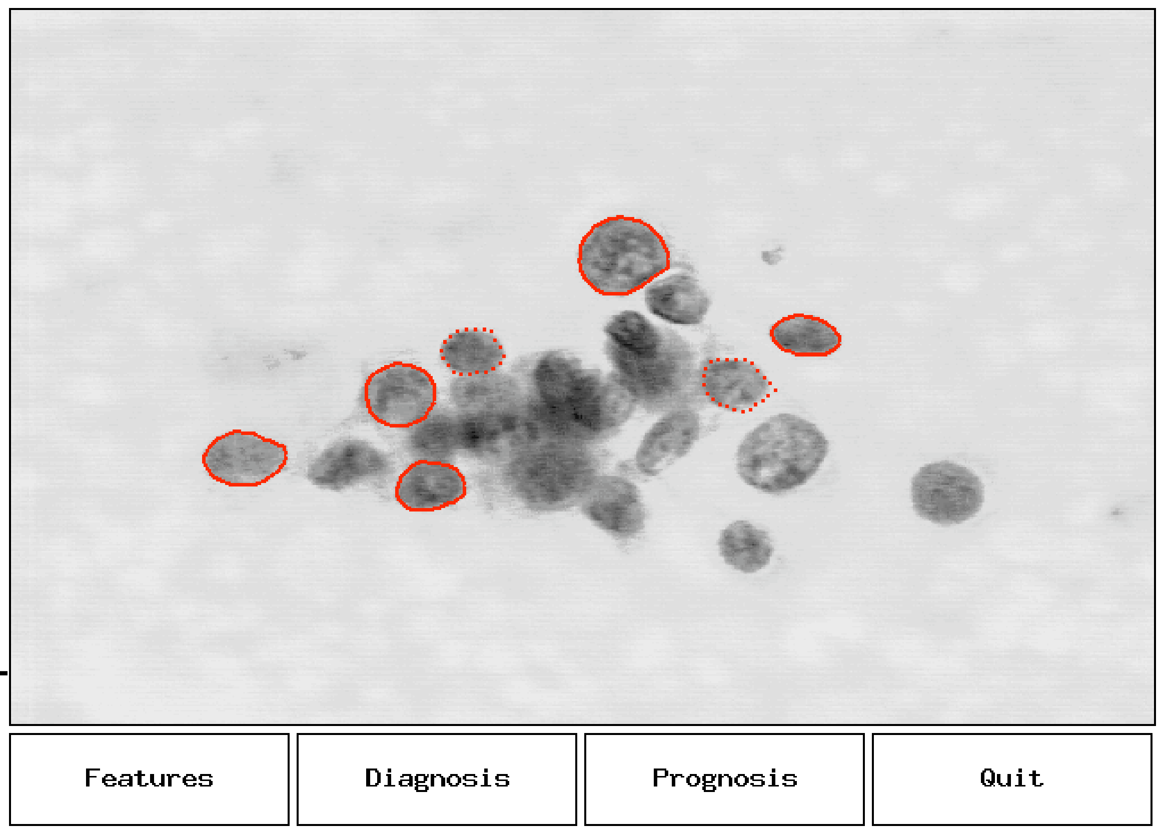
Supervised learning

- An example: Wisconsin Breast Cancer
- Formalization
- Supervised learning flowchart
- Univariate linear regression

Wisconsin Breast Cancer Data from UC Irvine ML Repository

- Cell samples were taken from tumors in breast cancer patients before surgery, and imaged
- Tumors were excised
- Patients were followed to determine whether or not the cancer recurred, and how long until recurrence or disease free

Cell Nuclei of Fine Needle Aspirate



Steps to solving a supervised learning problem

1. Collect data
2. Decide on inputs and output(s), including encoding
3. ...

Input features

- Researchers computed 30 different features of the cells' nuclei in the image.
  - Features relate to radius, "texture", area, smoothness, concavity, etc. of the nuclei
  - For each image, mean, standard error, and max of these properties across nuclei
- The result is a data table:

tumor size	texture	perimeter	...	outcome	time
18.02	27.6	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27
...					

Terminology

tumor size	texture	perimeter	...	outcome	time
18.02	27.6	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27
...					

- The columns are called <u>inputs</u> or <u>input variables</u> or <u>features</u> or <u>attributes</u>
- "outcome" and "time" are called <u>outputs</u> or <u>output variables</u> or <u>targets</u>
- A row in the table is called a <u>training example</u> or <u>sample</u> or <u>instance</u>
- The whole table is called the <u>training/data set</u>

- Usually, features are chosen based on some combination of expert knowledge, guesswork, and experimentation
- Sometimes, their choice/definition is also part of the learning problem, called a <u>feature selection</u> or <u>construction</u> problem

More generally

- Typically, a training example i has the form: $(x_{i,1} \ldots x_{i,n}, y_i)$ where n is the number of attributes (32 in our case).
- We will use the notation $\mathbf{x_i}$ to denote the column vector with elements $x_{i,1}, \ldots x_{i,n}$.
  (These are all the input feature values for one training example.)
- The training set D consists of m training examples
- Let $\mathcal{X}$ denote the space of input values (e.g., $\Re^{32}$)
- Let $\mathcal{Y}$ denote the space of output values (e.g. $\{N, R\}$, or $\Re$)

Supervised learning (almost) defined

Given a data set $D \subset \mathcal{X} \times \mathcal{Y}$, find a function:

$$h : \mathcal{X} \to \mathcal{Y}$$

such that $h(\mathbf{x})$ is a *"good predictor"* for the value of y.
h is called a *hypothesis*

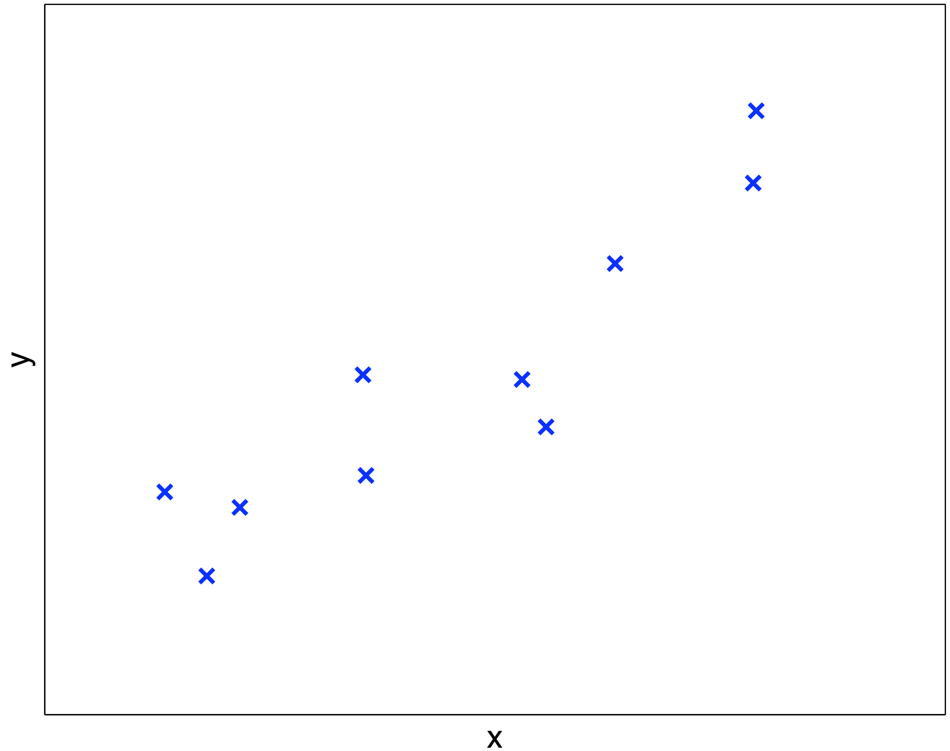
- If $\mathcal{Y} = \mathbb{R}$, this problem is called *regression*
- If $\mathcal{Y}$ is a finite discrete set, the problem is called *classification*
- If $\mathcal{Y}$ has 2 elements, the problem is called *binary classification* or *concept learning*
- The hypothesis h comes from a hypothesis class (or space) $\mathcal{H}$ of possible solutions.

(Note: Sometimes for classification problems we output the probability of each of the possible outputs.)

Steps to solving a supervised learning problem

1. Collect data
2. Decide on inputs and output(s), including encoding.
   This determines $\mathcal{X}$ and $\mathcal{Y}$.
3. Choose a hypothesis class.
   This determines $\mathcal{H}$.
4. ...

An abstract example



x	y
0.86	2.49
0.09	0.83
-0.85	-0.25
0.87	3.10
-0.44	0.87
-0.43	0.02
-1.10	-0.12
0.40	1.81
-0.96	-0.83
0.17	0.43

What hypothesis class do we choose to model the how y depends on x?

Linear hypotheses

- Suppose y was a linear function of $\mathbf{x}$:

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

- w_i are called *parameters* or *weights*
- To simplify notation, we always add an attribute $x_0 = 1$ to the other n attributes (also called *bias term* or *intercept term*):

$$h_{\mathbf{w}}(\mathbf{x}) = \sum_{i=0}^{n} w_i x_i = \mathbf{w}^{\mathbf{T}}\mathbf{x}$$

where $\mathbf{w}$ and $\mathbf{x}$ are vectors of length $n+1$.

How should we pick $\mathbf{w}$? No $\mathbf{w}$ exactly fits data...

Error minimization!

- Intuitively, $\mathbf{w}$ should make the predictions of $h_{\mathbf{w}}$ close to the true values y on the data we have
- Hence, we will define an *error function* or *cost function* to measure how much our prediction differs from the "true" answer
- We will pick $\mathbf{w}$ such that the error function is minimized

What error function should we choose?

Least mean squares (LMS)

- Main idea: try to make $h_{\mathbf{w}}(x)$ close to y on the examples in the training set
- We define a *sum-of-squares* error function

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{m}(h_{\mathbf{w}}(\mathbf{x_i}) - y_i)^2$$

- We will choose $\mathbf{w}$ such as to minimize $J(\mathbf{w})$

Steps to solving a supervised learning problem

1. Collect data
2. Decide on inputs and output(s), including encoding.
   This determines $\mathcal{X}$ and $\mathcal{Y}$.
3. Choose a hypothesis class.
   This determines $\mathcal{H}$.
4. Choose an error function (cost function) to define the best hypothesis
5. Choose an algorithm for searching efficiently through the space of hypotheses.

Minimizing LMS for a linear hypothesis class

$$
\begin{aligned}
J(\mathbf{w}) &= \frac{1}{2}\sum_{i=1}^{m}(h_{\mathbf{w}}(\mathbf{x_i}) - y_i)^2 \\
&= \frac{1}{2}\sum_{i=1}^{m}\left(\left(\sum_{j=0}^{n} w_j x_{i,j}\right) - y_i\right)^2
\end{aligned}
$$

How do we do it?

We had some discussion on the board, but this is pretty much where we ended. . .