An Evening of Machine Learning

Presented by Josh Hoak

www.common-index.com

13 January 2010

❃ A Motivating Example: Spam ❃

A normal email

Hi John,

I'm mostly done with creating new information for the Http Headers and I'm on to testing. There are a few details that I ignored for the first time around and will hit as I create the test specs.

Best,
Josh

A normal email?

Dear Friend,
RE: TRANSFER OF 25,400.000.00 USD (TWENTY FIVE MILLION,FOUR HUNDRED THOUSANDS US DOLLARS)
Let me start by introducing myself properly to you. I am Mr.Prince Tabo Charles, a consulting auditor,NedBank Plc, Johannesburg-South Africa.I have decided to contact you due to the urgency of this transaction.
THE PROPOSITION
A Foreigner, a German, Late Mr.Andreas Schranner, a majority stake holder in Habitat Real Estate Coy South Africa,until his death months ago in 25th July,2000 CONCORDE plane clash [Flight AF4590] ([http://news.bbc.co.uk/1/hi/world/europe/859479.stm),banked with NedBank Plc Johannesburg- South Africa,and had a closing balance as at the end of November, 2000 worth 25,400.000.00 (this is aside the accrued interest so far from that date). [...]

A more difficult example

From: Ivan Kilgore
Women will be your resigned slaves E-mail Newsletter Services Unsubscribe Detox page (do Levy not reply to windows this e-mail through) Your * leaves For annoying questions and Ex-JetBlue comments: Feedback better Ryals * cry For called advertising information not: Advertising an Sales bathroom Read Missouri

Definition (Machine Learning Classifier)

A Machine Learning Classifier is a function, $M : F \rightarrow C$ (with state[1]), where F is called the *feature space*, and C the *classes*.

In general, F will be $\mathbb{R}^n$, and C will be $\mathbb{N}$, although we will often see see $C = \{-1, 1\}$

[1] We could formalize the definition state by making our function $M : F \times S \rightarrow C$

Classification: An Example

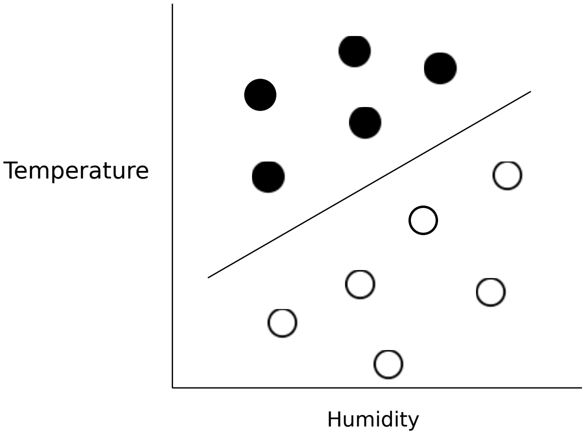
$$Features = \{temperature, overcast\ condition, precipitation, humidity\}$$
$$Classes = \{GoBiking, StayInside\}$$

We might want our Machine Learning Classifier M to do something like the following:

$$M(60^\circ F, cloudy, not\ rainy, 30\%) = GoBiking$$
$$M(70^\circ F, cloudy, rainy, 100\%) = GoBiking$$
$$M(20^\circ F, clear, not\ rainy, 0\%) = StayInside$$
$$M(100^\circ F, clear, not\ rainy, 70\%) = StayInside$$

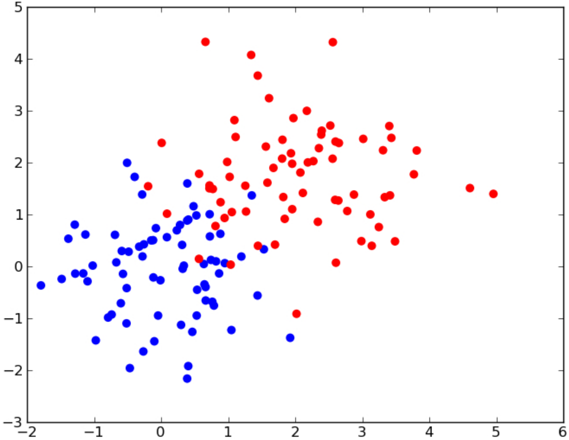
Classification: An Image

The Goal:



Classification: An Image

Reality



Classification

How should we make an algorithm that *dynamically* changes the way we classify an example (i.e., a set of features)?

Our algorithm needs to learn!

Definition (Learning)

Getting better at some task through practice.[2]

Definition (Learning Algorithm)

A Learning Algorithm is an algorithm L that modifies the state of a classifier M[3].

---

[2] Stephen Marsland: *Machine Learning: An Algorithmic Perspective*

[3] Similar to earlier, we could formally define L as the mapping $L : S \to S$

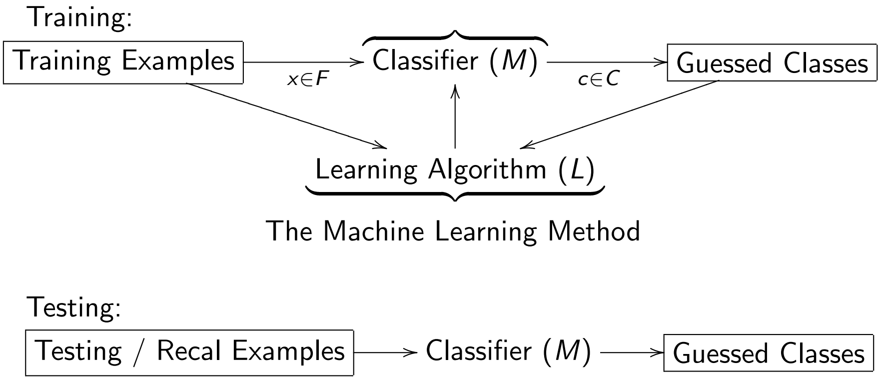
Types of Learning

- **Supervised Learning**: A learning algorithm is given training examples and after classifying the examples, the algorithm is told which classes are which and adjusts some state.
  Examples: **Perceptrons**, **Neural Nets**, **Support Vector Machines**, Decision Trees, Naive Bayes
- **Unsupervised Learning**: An learning algorithm is given training examples, but after classifying them, is not told the classes of the examples. Instead, the learning algorithm attempts to categorize inputs based on some commonality.
  Examples: **K-Means**, Vector Quantization, Self-Organizing Feature map

Types of Learning

- **Reinforcement Learning**: After a learning algorithm classifies training examples, it is told when it gets a training example wrong, but is not told what the correct answer should be.
  Examples: Markov Decision Processes, Q-Learning
- **Evolutionary Learning**: Learning inspired by biological evolution, using the concepts of fitness, alleles, crossover and mutation.
  Examples: **Genetic Algorithms**, Genetic Programming

Machine Learning Diagram



An outline of the presentation.

But first, a little history...

The Perceptron : invented in 1957 by Frank Rosenblatt, using techniques developed by McColloch and Pitts in 1943.

The Neural Network : finalized in 1986 by Rumelhart, Hinton, and McClelland. The learning algorithm was the hard part.

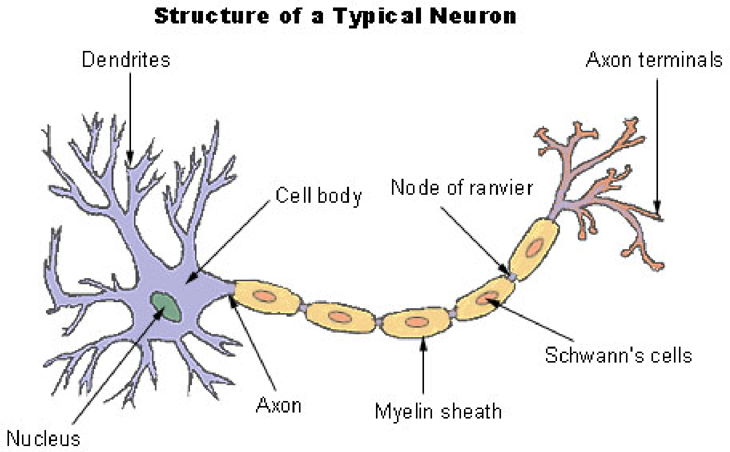
Support Vector Machines : developed by Vladimir Vapnik and Corinna Cortes in the early 1990s.

K-Means Clustering : were developed in the 1950s, but the algorithm currently used was published in 1982 by Frank Lloyd.

The Perceptron and the Neural Network

Perceptrons and Neural Networks

One idea: Model neurons (People Learn!)



A (very) brief overview of neurons

- Neurons receive synaptic signals to their dendrites and soma via synapses
- Synaptic signals may be either inhibitory or excitatory
- If the net excitation received by a neuron over a short period of time is large enough, the neuron generates a brief pulse called an action potential, which is sent along the neuron's axon to its synapses, and then to other neurons.

The Neuron Idea (continued)

- For an example, each feature can be thought of as a synapse connecting to a neuron.
- We can model excitation/inhibition as a weighted sum of the features.
- Where the neuron generates an action-potential, we output a class.

The McCulloch and Pitt Neuronal Model (M)

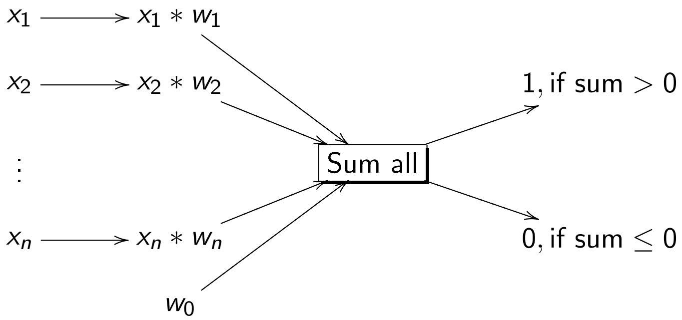
For $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$, we define the McCulloch and Pitt Neuronal Model as:

$$M(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{w} \cdot \mathbf{x} + w_0 > 0, \\ 0, & \text{else.} \end{cases}$$

- $\mathbf{x}$ refers to the feature vector. That is, $\mathbf{x} = \{x_1, x_2, \ldots, w_n\}$
- $\mathbf{w}$ refers to a weight vector. That is, $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$
- $\cdot$ is then an n dimensional dot-product
- (Really, this is a 1-node Perceptron classifier)

The McColloch and Pitt Neuronal Model (M)

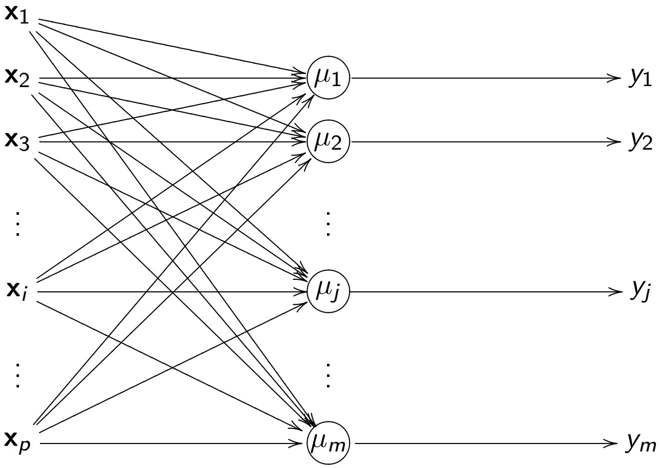
As a diagram:



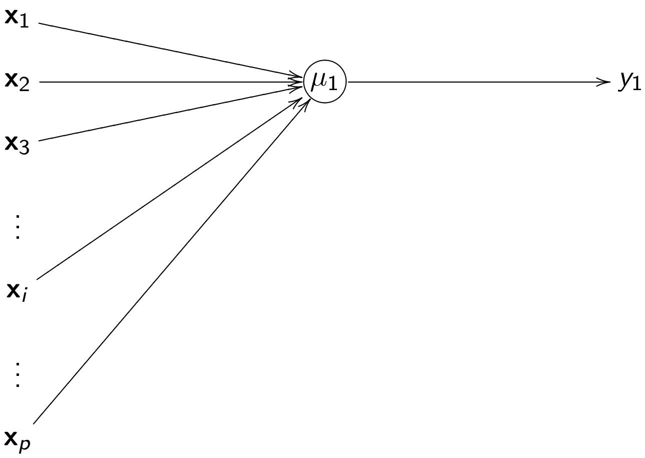
The Perceptron Classifier (M)

Perceptron: A network (digraph) of McColloch & Pitt Neurons.

Let μ be an M & P Neuron, then we have.



The two class case, again



Setup for the Learning Algorithm (L)

- X will be the set of vectors in the feature space F. Thus, $x \in X$ is called an *example*.
- For training, we shall assume that the class of each training example $\mathbf{x}$ is known, and shall be called t
- The update rule:

$$w_i \leftarrow w_i + \eta(t - y) \cdot x_i$$

- $\eta \in \mathbb{R}$, is called the learning rate. It governs how much we update the weights. Usually, $\eta \in [0.1, 0.4]$.

The Learning Algorithm (L)

Initialization:

1. Set all the weights in the weight vector $\mathbf{w}$ to a random value between -1 and 1.
2. Choose a value for the learning rate, η

The Learning Algorithm (L)

Training:

for j in T: (The number of training rounds)
for $\mathbf{x}_k$ in X: (the training examples)

1. Let $\mathbf{x} \leftarrow \mathbf{x}_k$, where $\mathbf{x} = \{x_1, \ldots, x_i, \ldots, x_n\}$.
   Compute the activation:

   1. $$a = \sum_{i=0}^{n} w_i x_i = \mathbf{w} \cdot \mathbf{x}$$
   2. $$y = \begin{cases} 1, & \text{if } a > 0, \\ 0, & \text{if } a \leq 0. \end{cases}$$

2. Update each of the weights using:

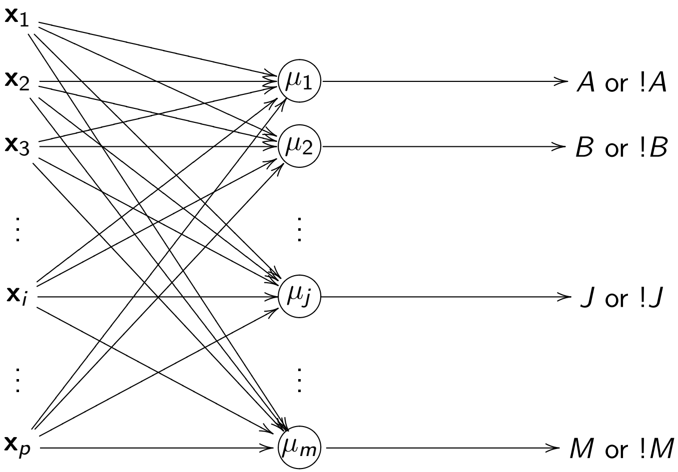
$$w_i \leftarrow w_i + \eta(t - y) \cdot x_i$$

Perceptron Example

http://www.generation5.org/jdk/demos/perceptronApplet.html

Multiclass-Perceptron (M)

For classes $\{A, B, \ldots, M\}$,



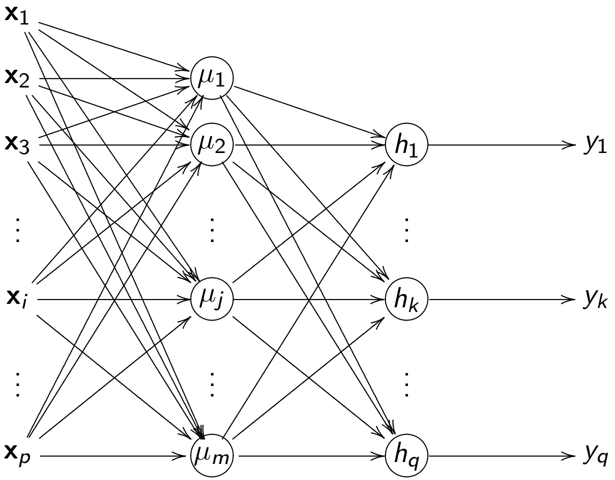
Neural Networks

- Problem: Only works for linearly separable data!

Neural Networks

- Problem: Only works for linearly separable data!
- Solution: Make multiple layers for our network.

Neural Networks



Neural Networks

It's clear how the classifier should work, but how should the Learning method work?

. . . Unfortunately, that's beyond the scope of this talk.

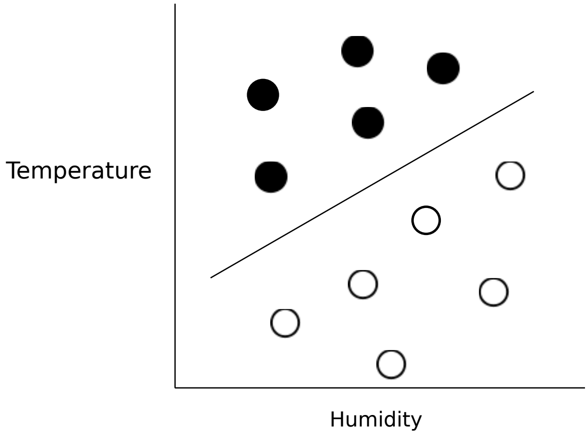
Support Vector Machines

Support Vector Machines

A story told through diagrams

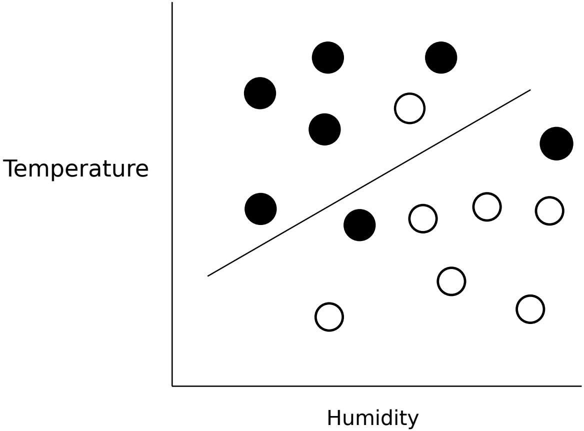
Support Vector Machines

Recall:

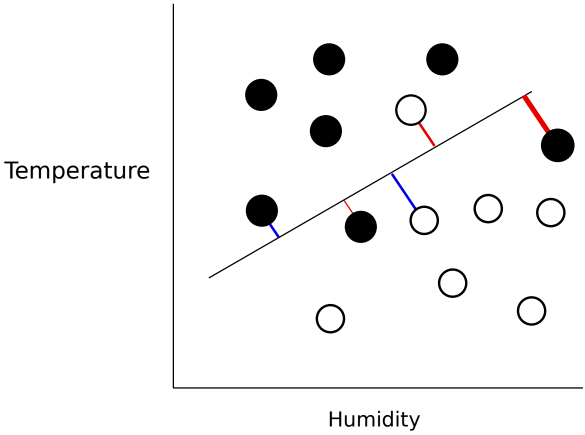


Support Vector Machines

Errors in classification:

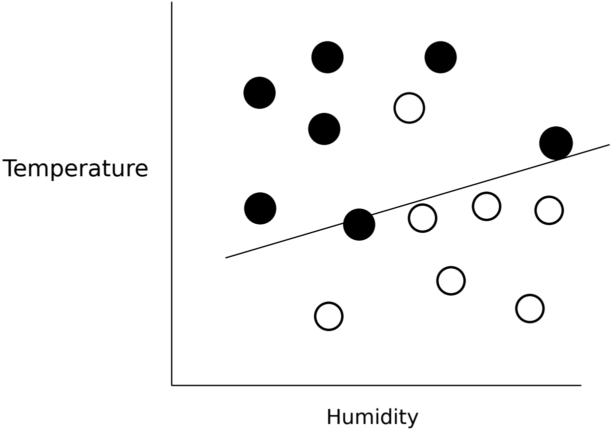


Weight by distance from the hyperplane:



Support Vector Machines

Move the hyperplane:

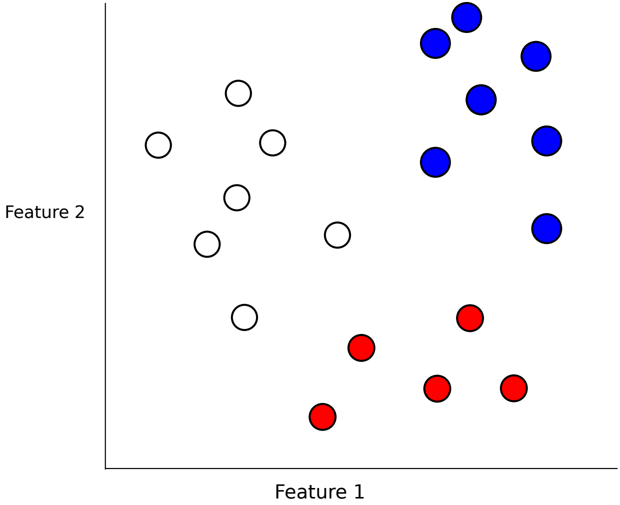


K-Means

K-Means

A brief excursion into unsupervised learning

K-Means



The K-Means Algorithm [4]

Initialization:

1. choose a value for k (the number of clusters)
2. choose k random positions in the input space
3. assign the cluster centers μ_j to those positions

[4] from *Machine Learning: An Algorithmic Perspective*

The K-Means Algorithm

Learning (L):

1. for each datapoint $\mathbf{x}_i$:
    1.1 compute the distance to each cluster center
    1.2 assign the datapoint to the nearest cluster centre with distance:

$$d_i = \min_j \; d(\mathbf{x}_i, \mu_j)$$

    1.3 for each cluster center:
        1.3.1 move the position of the centre to the mean of the points in that cluster (N_j is the number of points in the cluster j):

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{x}_i$$

    1.4 end when the cluster centers stop moving

The K-Means Algorithm

Classification (M):

1. For each test point:
   1.1 compute the distance to each cluster center
   1.2 assign the datapoint to the nearest cluster center with distance

$$d_i = \min_j \; d(\mathbf{x}_i, \mu_j)$$

K-Means Demo

http://metamerist.com/kmeans/example39.htm

Applications

Applications

How do we quantize problems?

Text Classification

Dear Friend,
RE: TRANSFER OF 25,400.000.00 USD (TWENTY FIVE MILLION,FOUR HUNDRED THOUSANDS US DOLLARS)
Let me start by introducing myself properly to you. I am Mr.Prince Tabo Charles, a consulting auditor,NedBank Plc, Johannesburg-South Africa.I have decided to contact you due to the urgency of this transaction. [...]

- What should our features be?

Text Classification

One example: let the features be the words; the values of the features shall be counts.

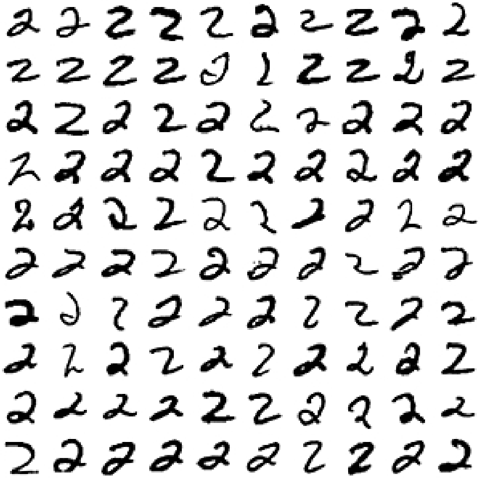
<Text Classification Example[5]>

[5] Source: http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

Text Classification: A small snag

From: Ivan Kilgore
Women will be your resigned slaves E-mail Newsletter Services Unsubscribe Detox page (do Levy not reply to windows this e-mail through) Your * leaves For annoying questions and Ex-JetBlue comments: Feedback better Ryals * cry For called advertising information not: Advertising an Sales bathroom Read Missouri

Handwriting Recognition



Handwriting Recognition

```
00000000001111111000000000000000
00000000011111111100000000000000
00000000011111111110000000000000
00000000011111111111100000000000
00000000111111111111100000000000
00000001111111111111110000000000
00000011111110001111110000000000
00000001111110000111111000000000
00000001111110000111111000000000
00000001111110000111111000000000
00000001111100000111111000000000
00000001111110000011111100000000
00000001111111000011111100000000
00000000111111000011111000000000
00000000111110000111111000000000
00000000001110000011111100000000
00000000000000000011111000000000
00000000000000000111110000000000
00000000000000000111111000000000
00000000000000001111110000000000
00000000000000011111110000000000
00000000000000111111100000000000
00000000000000011111110000000000
00000000000000111111100000000000
00000000000001111111000000000000
00000000000011111110000000000000
00000000001111111111111111111000
00000000011111111111111111111100
00000000111111111111111111111100
00000000011111111111111111111100
00000000001111111111111111111100
```

32 x 32 = 1 024 features!

Source: http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

Handwriting Recognition

```
00000000 0011 11111000000000000000
00000000 0111 11111100000000000000
00000000 0111 11111110000000000000
00000000 0111 11111111100000000000
00000000111111111111100000000000
00000001111111111111110000000000
00000011111110001111110000000000
00000001111110000111111000000000
00000001111110000111111000000000
00000001111110000111111000000000
00000001111100000111111000000000
00000001111110000011111100000000
00000001111111000011111100000000
00000000111111000011111000000000
00000000111110000111111000000000
00000000001110000011111100000000
00000000000000000011111000000000
00000000000000000111110000000000
```

Handwriting Recognition

0011
0111
0111
0111

$$= 14199$$

Handwriting Recognition

0011
0111
0111
0111

$$= 14199$$

Another way: 11 (1s)

Handwriting Recognition

A feature Vector:

2 -

1:0	2:1	3:6	4:15	5:12	6:1	7:0	8:0
9:0	10:7	11:16	12:6	13:6	14:10	15:0	16:0
17:0	18:8	19:16	20:2	21:0	22:11	23:2	24:0
25:0	26:5	27:16	28:3	29:0	30:5	31:7	32:0
33:0	34:7	35:13	36:3	37:0	38:8	39:7	40:0
41:0	42:4	43:12	44:0	45:1	46:13	47:5	48:0
49:0	50:0	51:14	52:9	53:15	54:9	55:0	56:0
57:0	58:0	59:6	60:14	61:7	62:1	63:0	64:0

Face Recognition

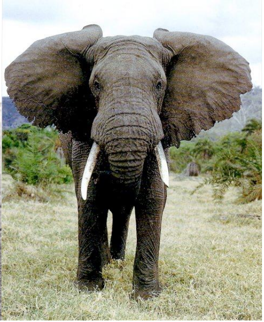
How about faces[6]?

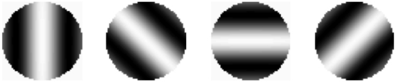
[6] Source: http://archive.ics.uci.edu/ml/datasets/CMU+Face+Images

Image Recognition

What about for generals images? Can we recognize the objects?







A small sample of Machine Learning techniques

Supervised Learning

- **Perceptrons**
- **Neural Nets**
- **Support Vector Machines**
- Decision Trees
- Ensemble Learning

Probabilistic Learning

- Minimizing Risk
- Naive Bayes
- Gaussian Mixture Models
- Nearest Neighbor Methods

Unsupervised Learning

- **K-Means**
- Vector Quantization
- Self-Organizing Feature map

Reinforcement Learning

- Markov Decision Processes
- Q-Learning

Evolutionary Learning

- **Genetic Algorithms**
- Genetic Programming

Optimization and Search

- Least-Squares Optimization
- Hill Climbing
- Simulated Annealing

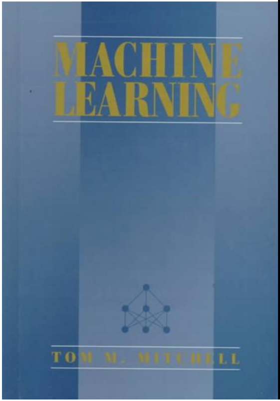
Monte Carlo Methods

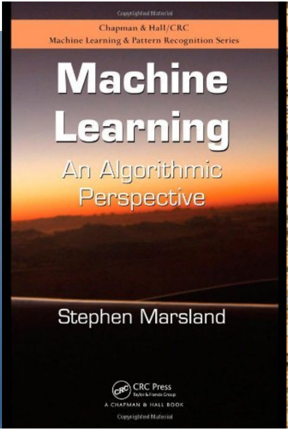
- Sampling Methods
- Makov Chain Monte Carlo Methods

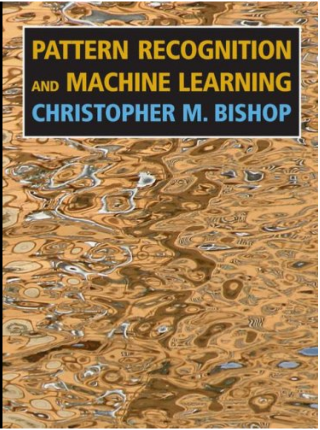
Graphical Learning

- Bayesian Networks
- Makov Random Fields
- Hidden Markov Models

Resources


MACHINE
LEARNING
TOM M. MITCHELL


Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series
Machine Learning
An Algorithmic Perspective
Stephen Marsland
CRC Press
A CHAPMAN & HALL BOOK


PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP

Resources

For Data - UCI Data Repository: http://archive.ics.uci.edu/ml/

Neural Networks Fast Artificial Neural Network Library: http://leenissen.dk/fann/

Support Vector Machines LibSVM and LibLinear: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Perceptrons Quick to write; subset of Neural Network libraries

K-Means Quick to write and easy to find

Genetic Algorithms - Quick to write