Induction: A process of reasoning (arguing) which infers a general conclusion based on individual cases

Supervised (Inductive) Learning

The University of Texas at Dallas

Supervised Learning

- **Given:** Training examples $\langle \mathbf{x}, f(\mathbf{x}) \rangle$ for some unknown function f.
- **Find:** A good approximation to f.

Example Applications

- **Credit risk assessment**

  $\mathbf{x}$: Properties of customer and proposed purchase.

  $f(\mathbf{x})$: Approve purchase or not.

- **Disease diagnosis**

  $\mathbf{x}$: Properties of patient (symptoms, lab tests)

  $f(\mathbf{x})$: Disease (or maybe, recommended therapy)

- **Face recognition**

  $\mathbf{x}$: Bitmap picture of person's face

  $f(\mathbf{x})$: Name of the person.

- **Automatic Steering**

  $\mathbf{x}$: Bitmap picture of road surface in front of car.

  $f(\mathbf{x})$: Degrees to turn the steering wheel.

Appropriate Applications for Supervised Learning

- **Situations where there is no human expert**

  $\mathbf{x}$: Bond graph for a new molecule.

  $f(\mathbf{x})$: Predicted binding strength to AIDS protease molecule.

- **Situations where humans can perform the task but can't describe how they do it.**

  $\mathbf{x}$: Bitmap picture of hand-written character

  $f(\mathbf{x})$: Ascii code of the character

- **Situations where the desired function is changing frequently**

  $\mathbf{x}$: Description of stock prices and trades for last 10 days.

  $f(\mathbf{x})$: Recommended stock transactions

- **Situations where each user needs a customized function** f

  $\mathbf{x}$: Incoming email message.

  $f(\mathbf{x})$: Importance score for presenting to user (or deleting without presenting).

A learning problem!

X	0	X
0	X	0
0	X	X

X	0	X
X	X	0
X	0	0

X	X	X
0	X	X
0	0	0

f(x)=1

0	X	0
X	0	X
0	X	X

0	0	X
X	X	0
0	X	X

0	X	X
X	0	0
0	X	X

f(x)=0

0	X	X
0	X	0
X	X	0

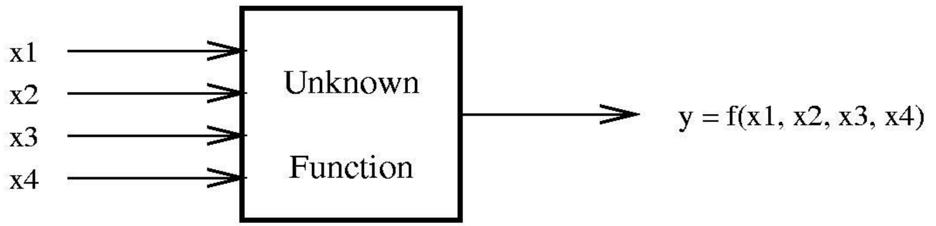
f(x)=?

If you prefer the training data in this form!

X1	X2	X3	X4	X5	X6	X7	X8	X9	f(x)
X	0	X	0	X	0	0	X	X	1
X	0	X	X	X	0	X	0	0	1
X	X	X	0	X	X	0	0	0	1
0	X	0	X	0	X	0	X	X	0
0	0	X	X	X	0	0	X	X	0
0	X	X	X	0	0	0	X	X	0
0	X	X	0	X	0	X	X	0	?

- x: a 9-dimensional vector
- f(x): a function or a program that takes the vector as input and outputs either a 0 or a 1
- **Task**: given the training examples, find a good approximation to f so that in future if you see an unseen vector "x" you will be able to figure out the value of f(x)

A Learning Problem



Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

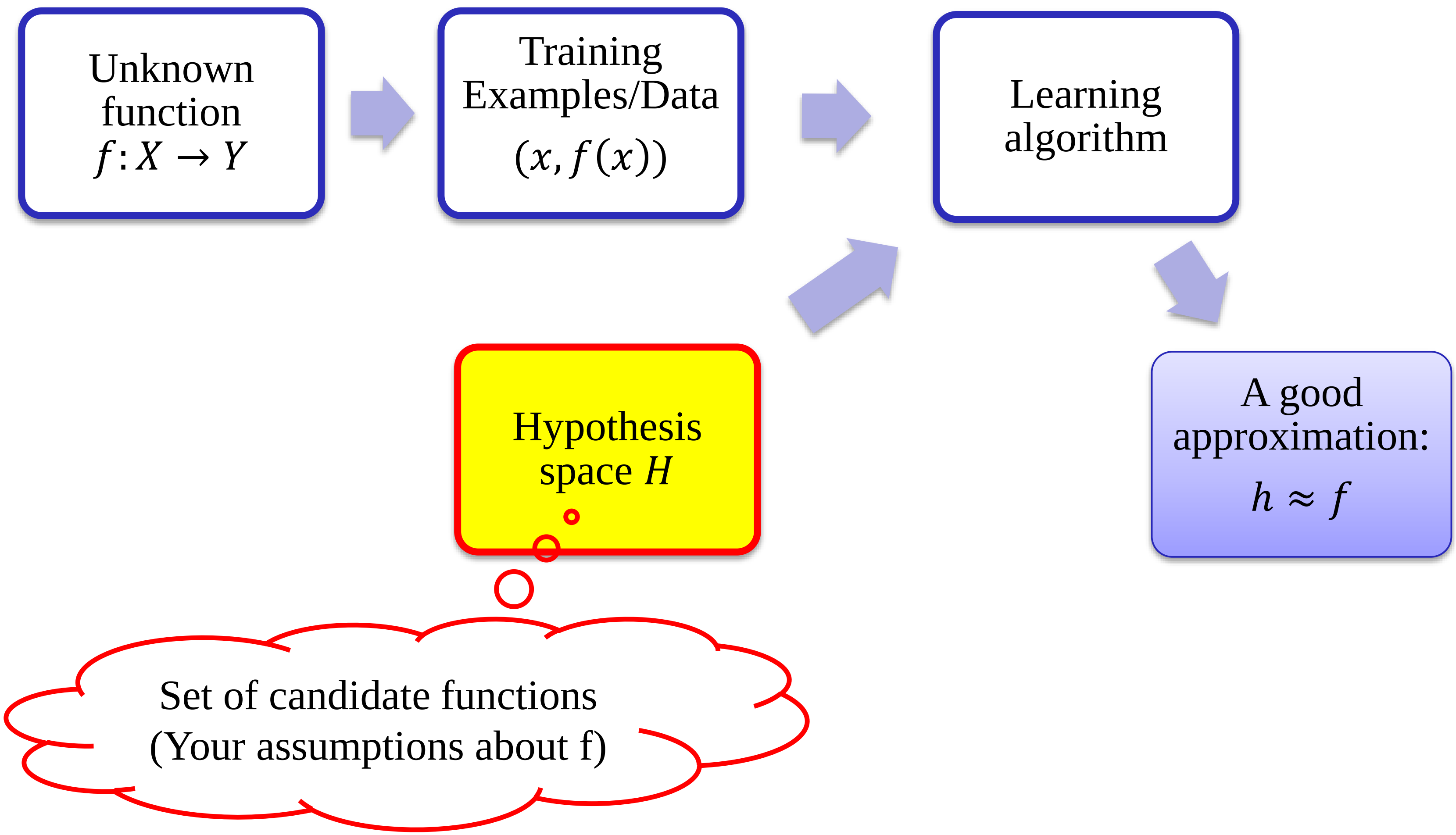
A simpler example for analysis!

Classification problem

Given data or examples, find the function f?

How to find a good approximation to f?

- A possible/plausible technique



Hypothesis Spaces

- **Complete Ignorance.** There are $2^{16} = 65536$ possible boolean functions over four input features. We can't figure out which one is correct until we've seen every possible input-output pair. After 7 examples, we still have 2^9 possibilities.

You are assuming that the unknown function f could be any one of the 2^{16} functions!

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

It turns out that out of the 2^{16} possible functions, 2^9 classify all points in the training data correctly!

Hypothesis Spaces (2)

- **Simple Rules.** There are only 16 simple conjunctive rules.

You are assuming that the unknown function f could be any one of the 16 conjunctive rules!

Unfortunately, none of them work

Rule	Counterexample
$\Rightarrow y$	1
$x_1 \Rightarrow y$	3
$x_2 \Rightarrow y$	2
$x_3 \Rightarrow y$	1
$x_4 \Rightarrow y$	7
$x_1 \wedge x_2 \Rightarrow y$	3
$x_1 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \Rightarrow y$	3
$x_2 \wedge x_4 \Rightarrow y$	3
$x_3 \wedge x_4 \Rightarrow y$	4
$x_1 \wedge x_2 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

No simple rule explains the data. The same is true for simple clauses.

Hypothesis Space (3)

- **m-of-n rules**. There are 32 possible rules (includes simple conjunctions and clauses).

At least *m* of the *n* variables must be true

You are assuming that the unknown function f could be any one of the 32 m-of-n rules!

Only one of them, the one marked by "***" works!

	Counterexample			
variables	1-of	2-of	3-of	4-of
$\{x_1\}$	3	–	–	–
$\{x_2\}$	2	–	–	–
$\{x_3\}$	1	–	–	–
$\{x_4\}$	7	–	–	–
$\{x_1, x_2\}$	3	3	–	–
$\{x_1, x_3\}$	4	3	–	–
$\{x_1, x_4\}$	6	3	–	–
$\{x_2, x_3\}$	2	3	–	–
$\{x_2, x_4\}$	2	3	–	–
$\{x_3, x_4\}$	4	4	–	–
$\{x_1, x_2, x_3\}$	1	3	3	–
$\{x_1, x_2, x_4\}$	2	3	3	–
$\{x_1, x_3, x_4\}$	1	***	3	–
$\{x_2, x_3, x_4\}$	1	5	3	–
$\{x_1, x_2, x_3, x_4\}$	1	5	3	3

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Two Views of Learning

- **Learning is the removal of our remaining uncertainty**. Suppose we *knew* that the unknown function was an m-of-n boolean function, then we could use the training examples to infer which function it is.
- **Learning requires guessing a good, small hypothesis class**. We can start with a very small class and enlarge it until it contains an hypothesis that fits the data.

We could be wrong!

- **Our prior knowledge might be wrong**
- **Our guess of the hypothesis class could be wrong**
  The smaller the hypothesis class, the more likely we are wrong.

Example: $x_4 \wedge Oneof\{x_1, x_3\} \Rightarrow y$ is also consistent with the training data.

Example: $x_4 \wedge \neg x_2 \Rightarrow y$ is also consistent with the training data.

If either of these is the unknown function, then we will make errors when we are given new x values.

Two Strategies for Machine Learning

- **Develop Languages for Expressing Prior Knowledge:** Rule grammars and stochastic models.
- **Develop Flexible Hypothesis Spaces:** Nested collections of hypotheses. Decision trees, rules, neural networks, cases.

In either case:

- **Develop Algorithms for Finding an Hypothesis that Fits the Data**

Terminology

- **Training example.** An example of the form $\langle \mathbf{x}, f(\mathbf{x}) \rangle$.
- **Target function (target concept).** The true function f.
- **Hypothesis**. A proposed function h believed to be similar to f.
- **Concept**. A boolean function. Examples for which $f(\mathbf{x}) = 1$ are called **positive examples** or **positive instances** of the concept. Examples for which $f(\mathbf{x}) = 0$ are called **negative examples** or **negative instances.**
- **Classifier**. A discrete-valued function. The possible values $f(\mathbf{x}) \in \{1, \ldots, K\}$ are called the **classes** or **class labels**.
- **Hypothesis Space**. The space of all hypotheses that can, in principle, be output by a learning algorithm.
- **Version Space**. The space of all hypotheses in the hypothesis space that have not yet been ruled out by a training example.

Key Issues in Machine Learning

- **What are good hypothesis spaces?**
  Which spaces have been useful in practical applications and why?
- **What algorithms can work with these spaces?**
  Are there general design principles for machine learning algorithms?
- **How can we optimize accuracy on future data points?**
  This is sometimes called the "problem of overfitting".
- **How can we have confidence in the results?**
  How much training data is required to find accurate hypotheses? (the *statistical question*)
- **Are some learning problems computationally intractable?**
  (the *computational question*)
- **How can we formulate application problems as machine learning problems?** (the *engineering question*)

Steps in Supervised Learning

1. Determine the representation for “x,f(x)” and determine what “x” to use

   Feature Engineering

2. Gather a training set (not all data is kosher)

   Data Cleaning

3. Select a suitable evaluation method
4. Find a suitable learning algorithm among a plethora of available choices
   – Issues discussed on the previous slide

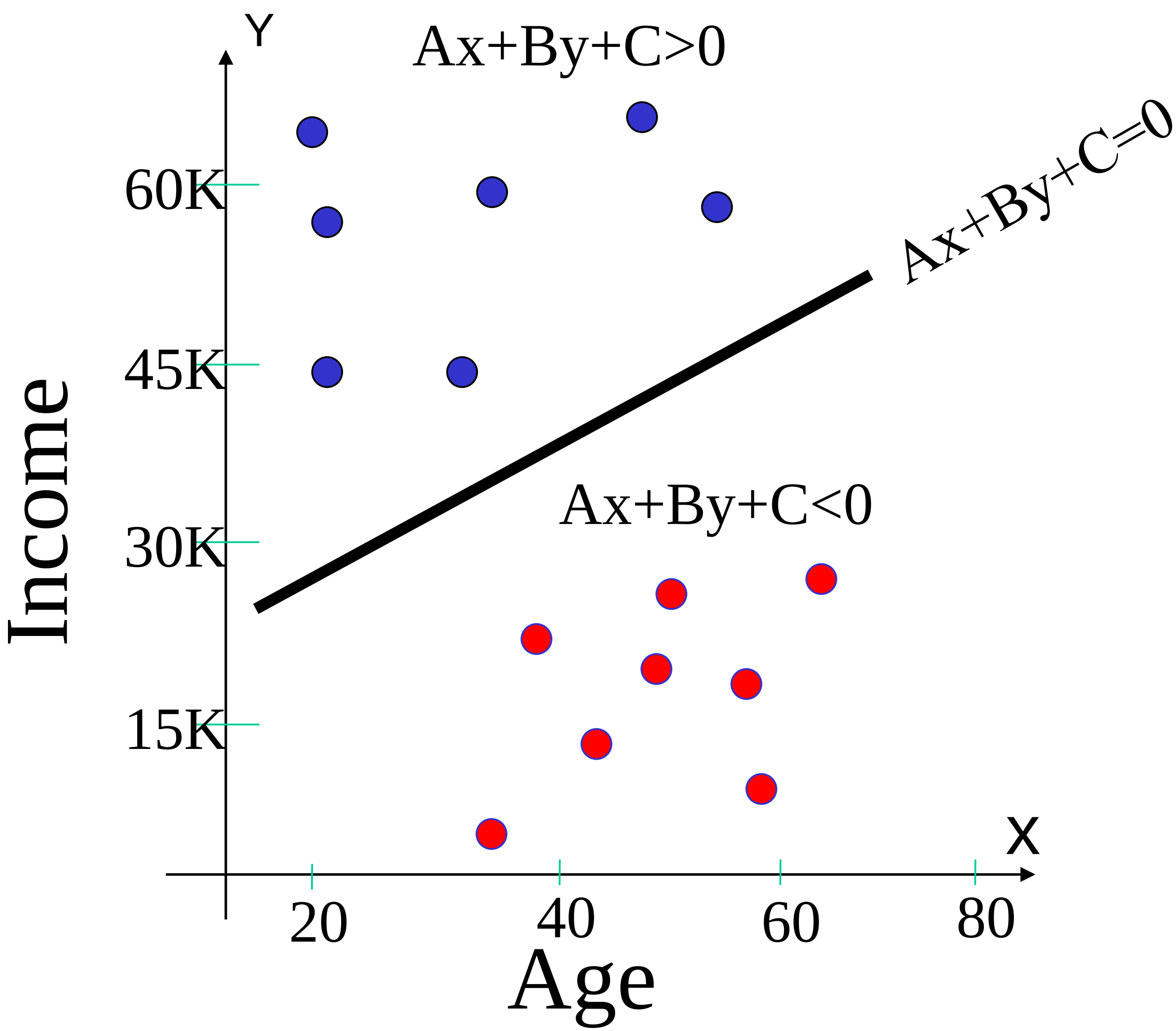
Feature Engineering is the Key

- Most effort in ML projects is constructing features
- Black art: Intuition, creativity required
  - Understand properties of the task at hand
  - How the features interact with or limit the algorithm you are using.
- ML is an iterative process
  - Try different types of features, experiment with each and then decide which feature set/algorithm combination to use

A sample machine learning Algorithm

- 2-way classification problem
  - +ve and –ve classes
- Representation: Lines (Ax+By=C)
  - Specifically
    - if Ax+By+C >0 then classify "+ve"
    - Else classify as "-ve"
- Evaluation: Number of mis-classified examples
- Optimization: An algorithm that searches for the three parameters: A, B and C.

Toy Example



Blue circles: Good credit (low risk)

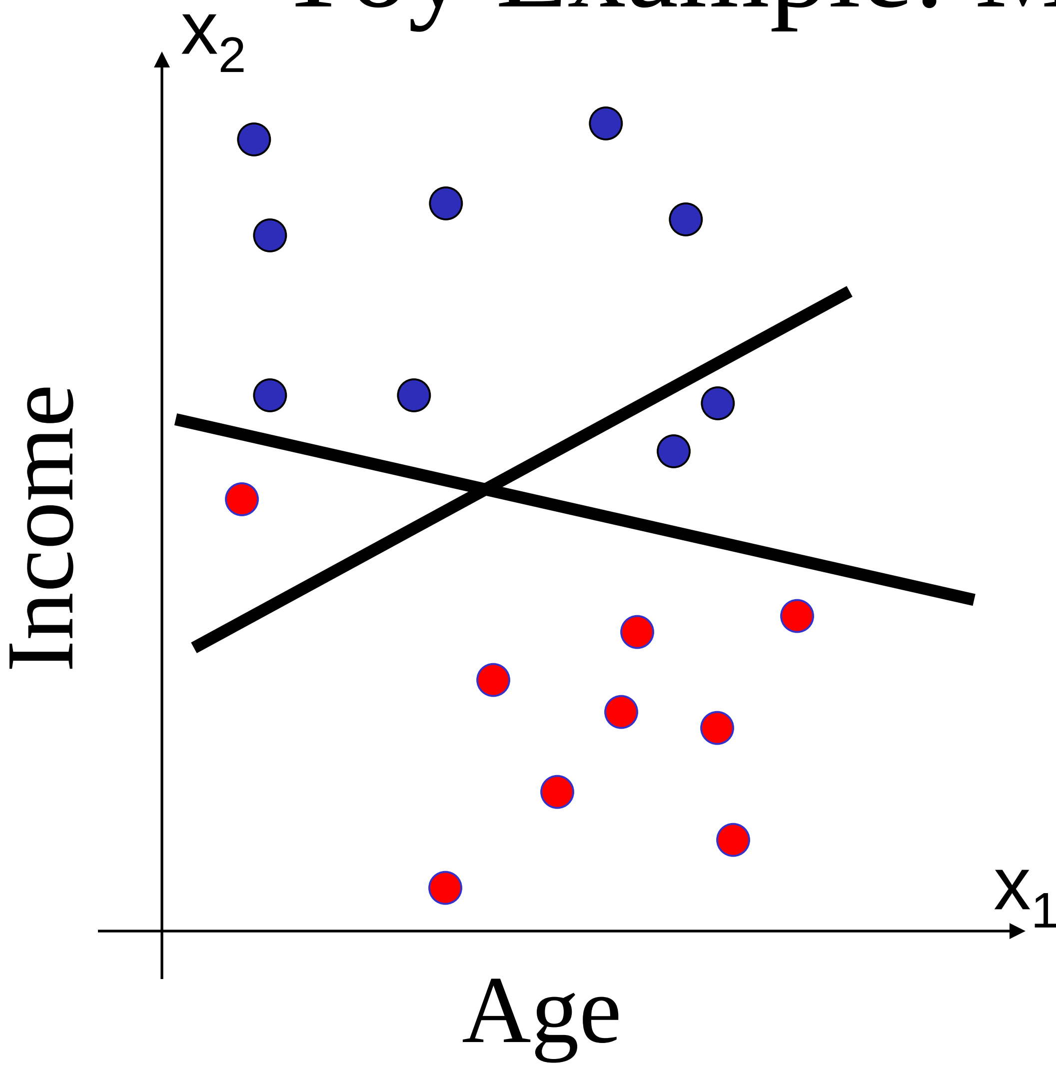
Red circles: Bad credit (high risk)

Problem: Fit a line that separates the two such that the error is minimized.

How do machine learners solve this problem?

- Try different lines until you find one that separates the data into two
- A more plausible alternative
  - Begin with a random line
  - Repeat until no errors
  - For each point
    - If the current line says +ve and point is –ve then decrease A, B and C
    - If the current line says –ve and the point is +ve then increase A, B, and C

Toy Example: More data



Blue circles: Good credit (low risk)
Red circles: Bad credit (high risk)

Problem: Fit a line that separates the two such that the error is minimized.

Learning = Representation + Evaluation + Optimization

- Combinations of just three elements

Representation	Evaluation	Optimization
Instances	Accuracy	Greedy search
Hyperplanes	Precision/Recall	Branch & bound
Decision trees	Squared error	Gradient descent
Sets of rules	Likelihood	Quasi-Newton
Neural networks	Posterior prob.	Linear progr.
Graphical models	Margin	Quadratic progr.
Etc.	Etc.	Etc.