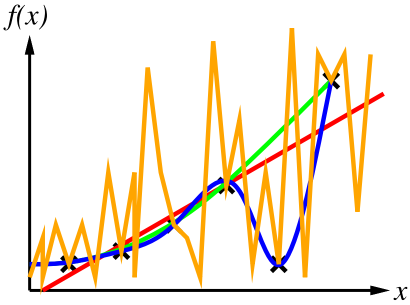
CS325 Artificial Intelligence
Chs. 18 & 4 – Supervised Machine Learning (cont)

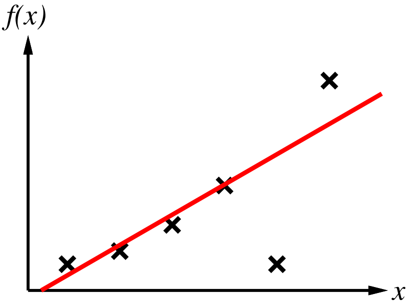
Cengiz Günay

Spring 2013

Model Complexity in Learning

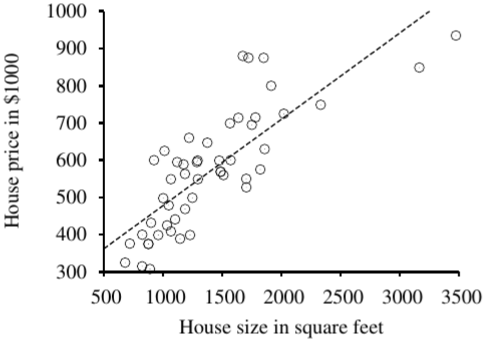


Model Complexity in Learning

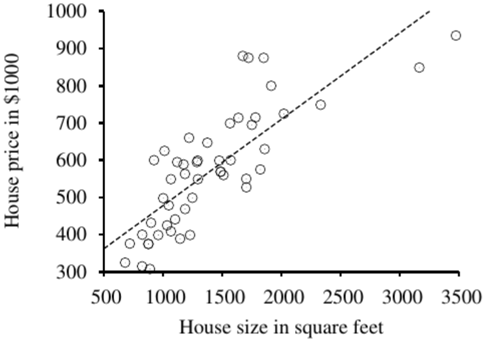


Let's start with the linear case. . .

Linear Regression

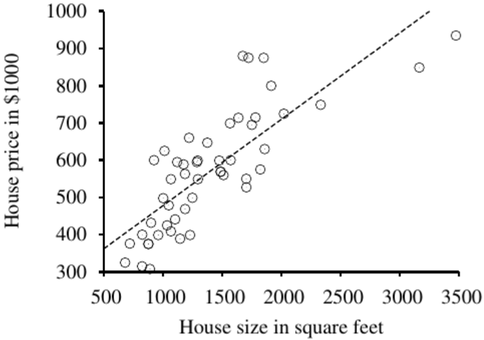


Linear Regression



$$\text{price} = f(\text{size})$$

Linear Regression



$$\text{price} = f(\text{size})$$

$$? = f(3000)$$

Regression—Finding the Parameters from Data

$$y = f(x) = w_1 x + w_0$$

x	y
2	7
-1	-2
5	16
-3	-8

$$w_0 = ?$$
$$w_1 = ?$$

Regression—Finding the Parameters from Data

$$y = f(x) = w_1 x + w_0$$

x	y
2	7
-1	-2
5	16
-3	-8

$$w_0 = 1$$
$$w_1 = 2$$

Linear Regression—Defining a Loss Function

$$y = f(x) = w_1 x + w_0$$

$$\mathrm{Loss}(f) = \sum_j (y_j - f(x_j))^2 = \sum_j (y_j - (w_1 x_j + w_0))^2$$

Linear Regression—Defining a Loss Function

$$y = f(x) = w_1 x + w_0$$

$$\mathrm{Loss}(f) = \sum_j (y_j - f(x_j))^2 = \sum_j (y_j - (w_1 x_j + w_0))^2$$

Minimum is where the derivative is zero:

$$\frac{\partial}{\partial w_0} \sum_j (y_j - (w_1 x_j + w_0))^2 = 0, \quad \frac{\partial}{\partial w_1} \sum_j (y_j - (w_1 x_j + w_0))^2 = 0$$

Linear Regression—Defining a Loss Function

$$y = f(x) = w_1 x + w_0$$

$$\mathrm{Loss}(f) = \sum_j (y_j - f(x_j))^2 = \sum_j (y_j - (w_1 x_j + w_0))^2$$

Minimum is where the derivative is zero:

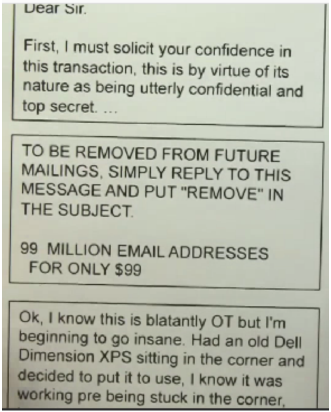
$$\frac{\partial}{\partial w_0} \sum_j (y_j - (w_1 x_j + w_0))^2 = 0, \quad \frac{\partial}{\partial w_1} \sum_j (y_j - (w_1 x_j + w_0))^2 = 0$$

Solution is:

$$w_1 = \frac{N(\sum x_j y_j) - (\sum x_j)(\sum y_j)}{N(\sum x_j^2) - (\sum y_j)^2}, \quad w_0 = \left(\sum y_j - w_1(\sum x_j)\right) / N$$

Remember Bayes Nets?
We can learn them from data, too.

- Everybody loves spam!


Dear Sir.

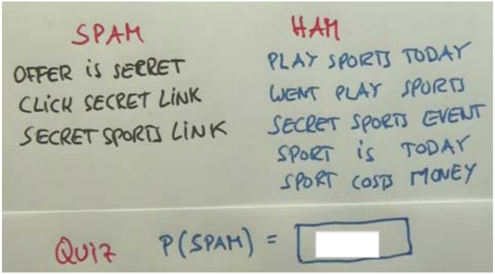
First, I must solicit your confidence in
this transaction, this is by virtue of its
nature as being utterly confidential and
top secret. ...

TO BE REMOVED FROM FUTURE
MAILINGS, SIMPLY REPLY TO THIS
MESSAGE AND PUT "REMOVE" IN
THE SUBJECT.

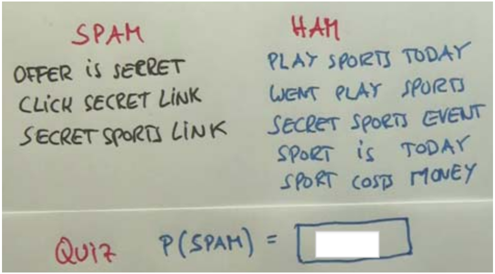
99 MILLION EMAIL ADDRESSES
FOR ONLY $99

Ok, I know this is blatantly OT but I'm
beginning to go insane. Had an old Dell
Dimension XPS sitting in the corner and
decided to put it to use, I know it was
working pre being stuck in the corner,

Maximum Likelihood: Guessing Spam Probability

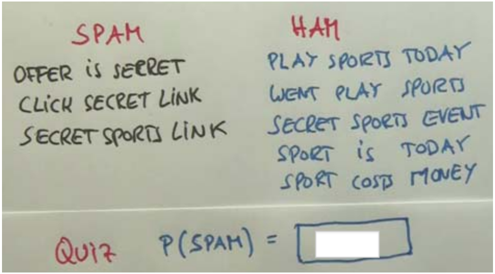


Maximum Likelihood: Guessing Spam Probability



Let's guess: $P(S) = \pi$

$$P(y_i) = \begin{cases} \pi & \text{if } y_i = S \\ 1 - \pi & \text{if } y_i = H \end{cases}$$



Let's guess: $P(S) = \pi$

$$P(y_i) = \begin{cases} \pi & \text{if } y_i = S \\ 1 - \pi & \text{if } y_i = H \end{cases}$$

Joint probability

$$\begin{aligned} P(\text{data}) &= \pi^{\text{count}(S)} \times (1 - \pi)^{\text{count}(H)} \\ &= \pi^3 \times (1 - \pi)^5 \end{aligned}$$

Maximum Likelihood: Guessing Spam Probability

Joint probability

$$
\begin{aligned}
P(\text{data}) &= \pi^{\text{count}(S)} \times (1-\pi)^{\text{count}(H)} \\
&= \pi^3 \times (1-\pi)^5
\end{aligned}
$$

Take log of both sides

$$
\log P(\text{data}) = 3 \log \pi + 5 \log(1-\pi)
$$

Joint probability

$$\begin{aligned} P(\text{data}) &= \pi^{\text{count}(S)} \times (1-\pi)^{\text{count}(H)} \\ &= \pi^3 \times (1-\pi)^5 \end{aligned}$$

Take log of both sides

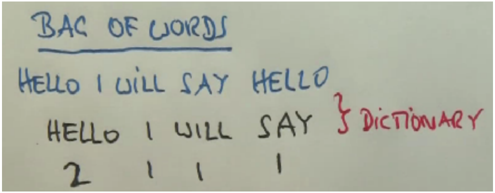
$$\log P(\text{data}) = 3 \log \pi + 5 \log(1-\pi)$$

Find max by zero derivative

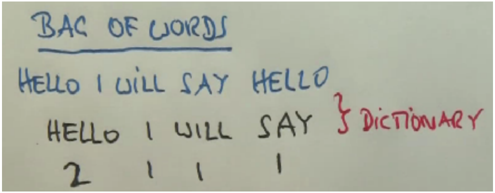
$$\frac{\nabla P(\text{data})}{\nabla \pi} = 0 = \frac{3}{\pi} - \frac{5}{1-\pi}$$

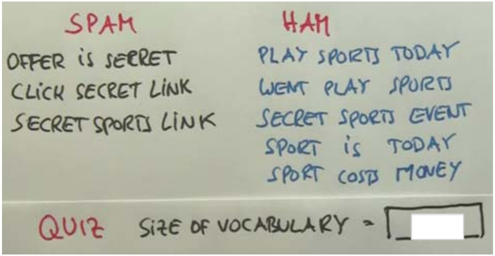
$$\pi = 3/8$$

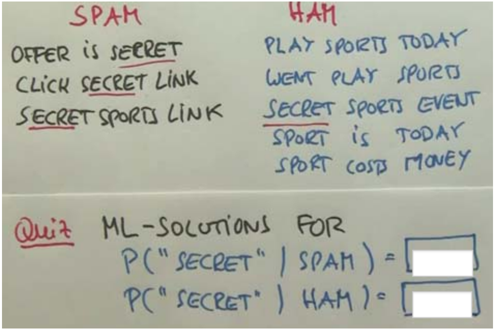
Bag of Words Representation



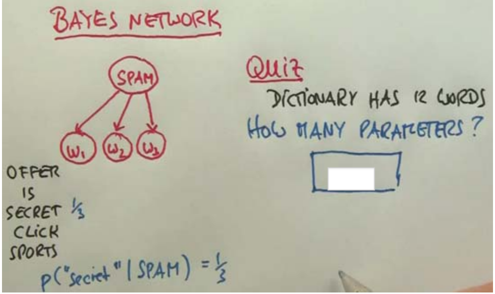
Bag of Words Representation



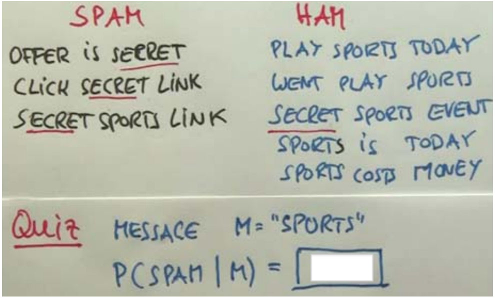



SPAM
HAM
OFFER IS SECRET
CLICK SECRET LINK
SECRET SPORTS LINK
PLAY SPORTS TODAY
WENT PLAY SPORTS
SECRET SPORTS EVENT
SPORT IS TODAY
SPORT COSTS MONEY
Quiz ML-SOLUTIONS FOR
P("SECRET" | SPAM) =
P("SECRET" | HAM) =

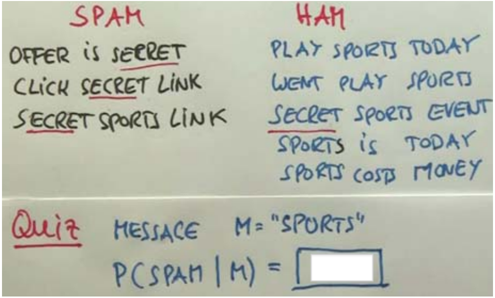
Finally, Back to Bayes Nets



Finally, Back to Bayes Nets

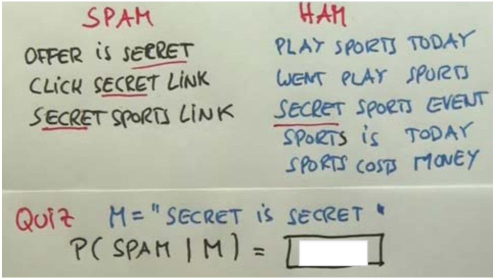


Finally, Back to Bayes Nets

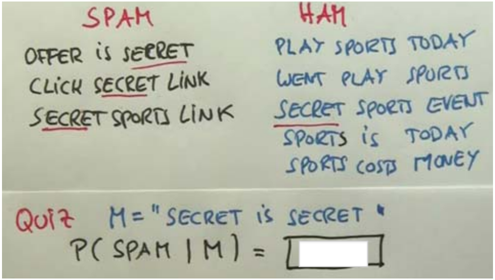


$$P(S|M) = \alpha P(M|S)P(S)$$

Finally, Back to Bayes Nets

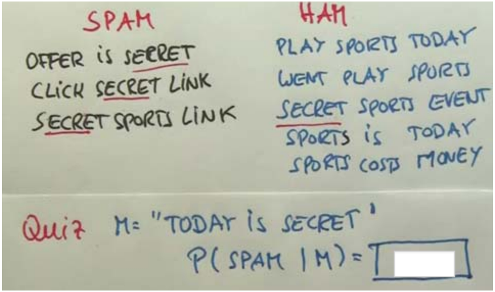


$$P(S|M) = \alpha P(M|S)P(S)$$

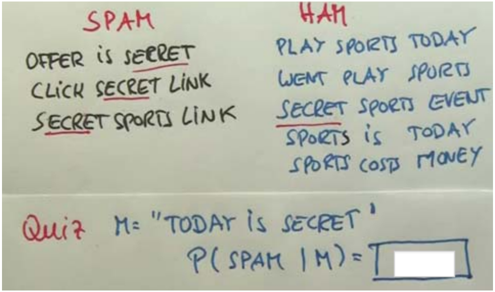


$$P(S|M) = \alpha P(M|S)P(S) = \alpha P(M_1, M_2, M_3|S)P(S)$$

Problems?

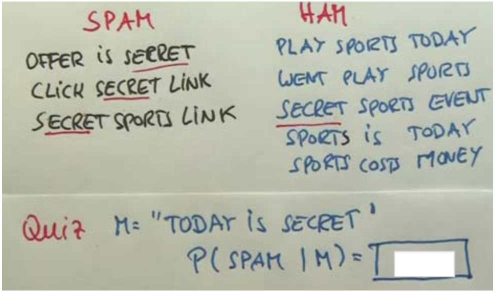

SPAM
OFFER is SECRET
CLICK SECRET LINK
SECRET SPORTS LINK
HAM
PLAY SPORTS TODAY
WENT PLAY SPORTS
SECRET SPORTS EVENT
SPORTS is TODAY
SPORTS COSTS MONEY
Quiz M= "TODAY is SECRET"
P(SPAM | M)=

Problems?



$$P(S|M) = 0$$

Problems?



$$P(S|M) = 0$$

- Need Laplace Smoothing (check the videos)

What If We Cannot Learn So Easily?

So far we calculated directly from data:

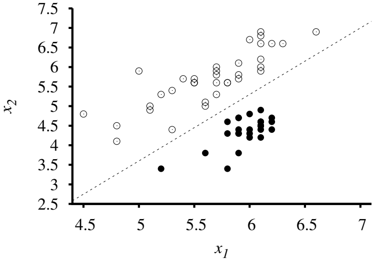
- **Linear regression coefficients** through explicit solution
- **Bayes net parameters** through maximal likelihood

What If We Cannot Learn So Easily?

So far we calculated directly from data:

- **Linear regression coefficients** through explicit solution
- **Bayes net parameters** through maximal likelihood

But cannot solve every problem with these. Classification solution is not unique:

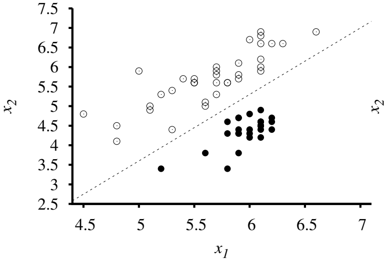


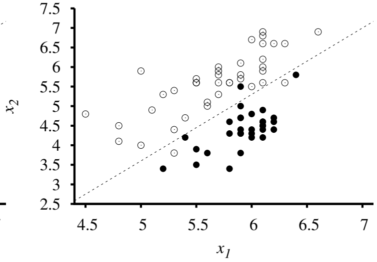
What If We Cannot Learn So Easily?

So far we calculated directly from data:

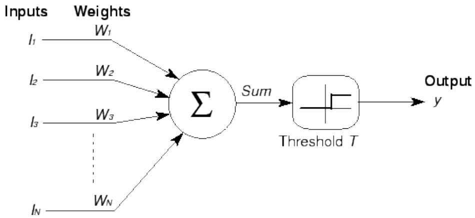
- **Linear regression coefficients** through explicit solution
- **Bayes net parameters** through maximal likelihood

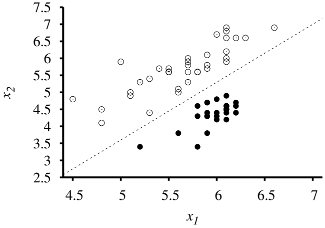
But cannot solve every problem with these. Classification solution is not unique:





Perceptron Also Calculates Linear Boundary



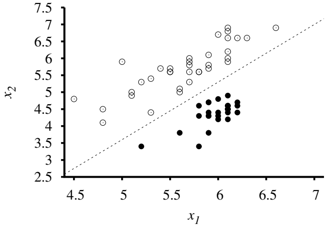


$$\text{sum} = \sum_{i=1}^{N} I_i W_i$$

$$y = \begin{cases} 1, & \text{if sum} \geq T \\ 0, & \text{if sum} < T \end{cases}$$

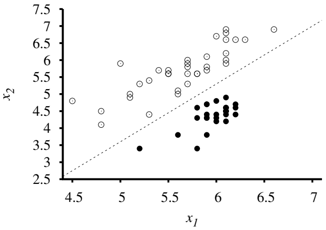
Line equation in 2D:

$$x_2 = a\,x_1 + b$$



Line equation in 2D:

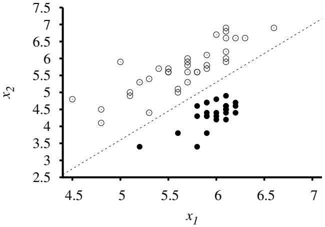
$$x_2 = a\,x_1 + b$$
$$-b = a\,x_1 - x_2$$



Perceptron, 2D Case

Line equation in 2D:

$$x_2 = a\,x_1 + b$$
$$-b = a\,x_1 - x_2$$



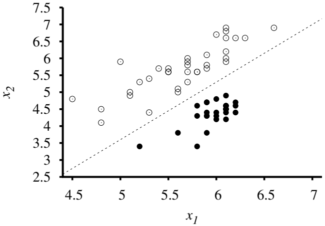
The perceptron boundary:

$$T = w_1 x_1 + w_2 x_2$$

Perceptron, 2D Case

Line equation in 2D:

$$
\begin{aligned}
x_2 &= a\,x_1 + b \\
-b &= a\,x_1 - x_2
\end{aligned}
$$



The perceptron boundary:

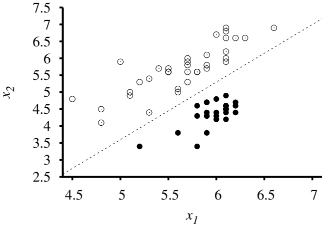
$$
T = w_1 x_1 + w_2 x_2
$$

$$
y = \begin{cases} 1, & \text{if sum} \geq T \\ 0, & \text{if sum} < T \end{cases}
$$

Perceptron, 2D Case

Line equation in 2D:

$$
\begin{aligned}
x_2 &= a\,x_1 + b \\
-b &= a\,x_1 - x_2
\end{aligned}
$$



The perceptron boundary:

$$
T = w_1 x_1 + w_2 x_2
$$

$$
y = \begin{cases} 1, & \text{if sum} \geq T \\ 0, & \text{if sum} < T \end{cases}
$$

How to learn it?

Use the Loss Function, Perceptron

Perceptron:

$$y = f_{\mathbf{w}}(\mathbf{x})$$

Over all samples:

$$\mathrm{Loss}(\mathbf{w}) = \sum_i (y_i - f_{\mathbf{w}}(\mathbf{x}_i))^2$$

Use the Loss Function, Perceptron

Perceptron:

$$y = f_{\mathbf{w}}(\mathbf{x})$$

Over all samples:

$$\mathrm{Loss}(\mathbf{w}) = \sum_i \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

Trying to find

$$\arg\min_{\mathbf{w}} \mathrm{Loss}(\mathbf{w})$$

Use the Loss Function, Perceptron

Perceptron:

$$y = f_{\mathbf{w}}(\mathbf{x})$$

Over all samples:

$$\mathrm{Loss}(\mathbf{w}) = \sum_i (y_i - f_{\mathbf{w}}(\mathbf{x}_i))^2$$

Trying to find

$$\arg\min_{\mathbf{w}} \mathrm{Loss}(\mathbf{w})$$

An incremental rule:

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} \mathrm{Loss}(\mathbf{w})$$

Use the Loss Function, Perceptron

Perceptron:

$$y = f_{\mathbf{w}}(\mathbf{x})$$

Over all samples:

$$\mathrm{Loss}(\mathbf{w}) = \sum_i (y_i - f_{\mathbf{w}}(\mathbf{x}_i))^2$$

Trying to find

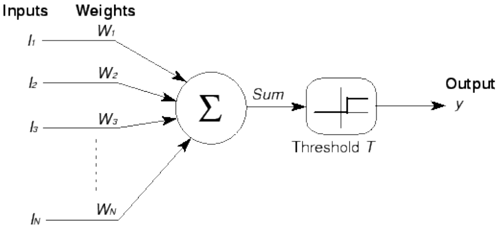
$$\arg\min_{\mathbf{w}} \mathrm{Loss}(\mathbf{w})$$

An incremental rule:

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} \mathrm{Loss}(\mathbf{w})$$

$$w_j \leftarrow w_j + \alpha (y - f_{\mathbf{w}}(\mathbf{x})) \times x_j$$

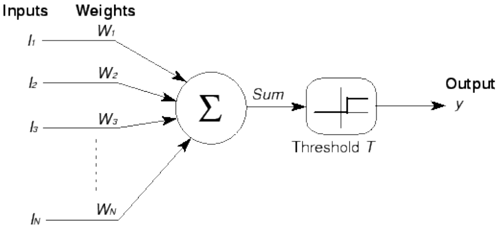
It's Called the Perceptron Learning Rule



$$w_j \leftarrow w_j + \alpha(y - f_{\mathbf{w}}(\mathbf{x})) \times x_j$$

		$y = 0$ Tom	$y = 1$ Jerry
x	Trucks	1	0
	Sedans	0	1
	Hybrids	0	1
	SUVs	1	0

It's Called the Perceptron Learning Rule



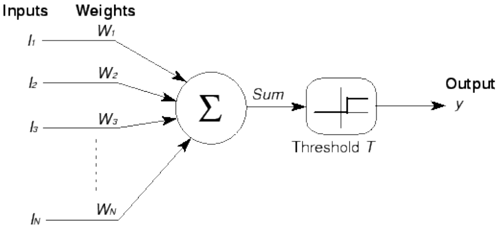
$$w_j \leftarrow w_j + \alpha(y - f_{\mathbf{w}}(\mathbf{x})) \times x_j$$

		$y = 0$ Tom	$y = 1$ Jerry
x	Trucks	1	0
	Sedans	0	1
	Hybrids	0	1
	SUVs	1	0

Start: $\mathbf{w} = 0$, $\alpha = 1$, $T = 1$.

For y_{Tom}:

It's Called the Perceptron Learning Rule



$$w_j \leftarrow w_j + \alpha(y - f_{\mathbf{w}}(\mathbf{x})) \times x_j$$

		$y=0$ Tom	$y=1$ Jerry
x	Trucks	1	0
	Sedans	0	1
	Hybrids	0	1
	SUVs	1	0

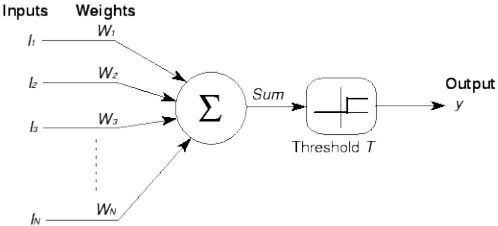
Start: $\mathbf{w} = 0,\ \alpha = 1,\ T = 1.$

For y_{Tom}:

$$w_{\text{Trucks}} \leftarrow w_{\text{Trucks}} + (0-0) \times 1$$

For y_{Jerry}:

It's Called the Perceptron Learning Rule



$$w_j \leftarrow w_j + \alpha(y - f_{\mathbf{w}}(\mathbf{x})) \times x_j$$

		$y = 0$ Tom	$y = 1$ Jerry
x	Trucks	1	0
	Sedans	0	1
	Hybrids	0	1
	SUVs	1	0

Start: $\mathbf{w} = 0,\ \alpha = 1,\ T = 1.$

For y_{Tom}:

$$w_{\text{Trucks}} \leftarrow w_{\text{Trucks}} + (0 - 0) \times 1$$

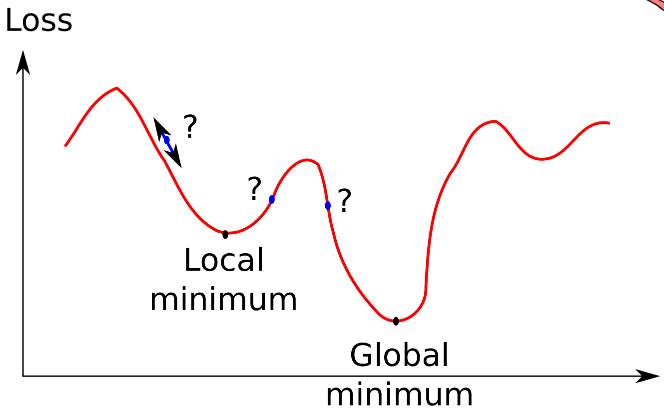
For y_{Jerry}:

$$w_{\text{Sedans}} \leftarrow w_{\text{Sedans}} + (1 - 0) \times 1$$

Gradient Descent on the Loss Function

In general,

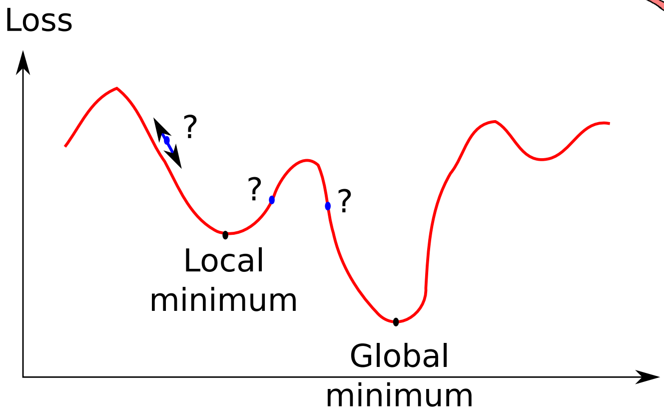
$$w_j \leftarrow w_j + \alpha \frac{\partial}{\partial w_j} \mathrm{Loss}(\mathbf{w})$$



Gradient Descent on the Loss Function

In general,

$$w_j \leftarrow w_j + \alpha \frac{\partial}{\partial w_j} \mathrm{Loss}(\mathbf{w})$$

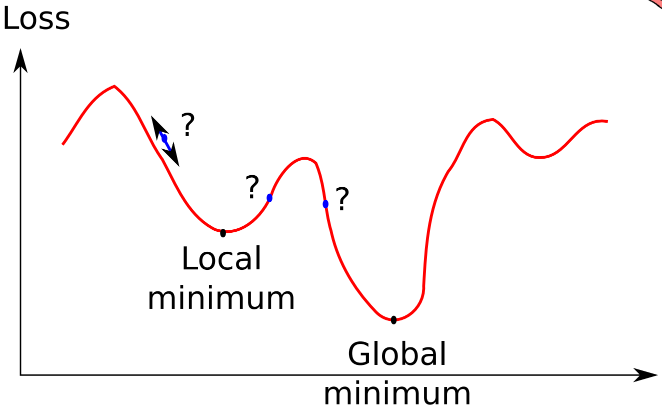


Adaptive α $\Rightarrow$ **Simulated Annealing**

Gradient Descent on the Loss Function

In general,

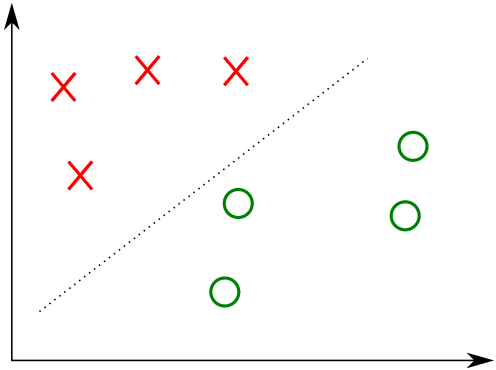
$$w_j \leftarrow w_j + \alpha \frac{\partial}{\partial w_j} \mathrm{Loss}(\mathbf{w})$$



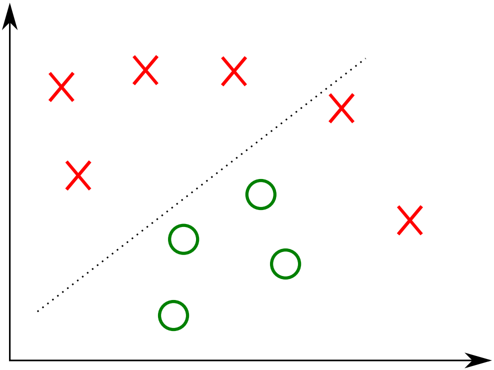
Adaptive α $\Rightarrow$**Simulated Annealing**

Major problem: **local minima**

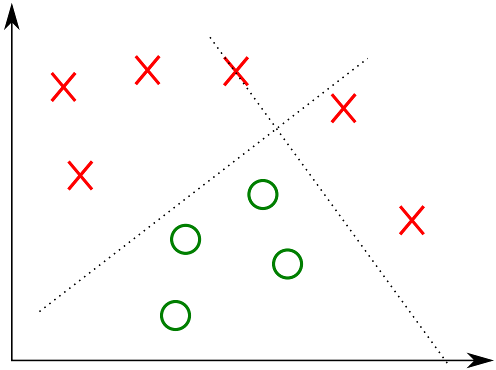
What If the Boundary is Non-linear?

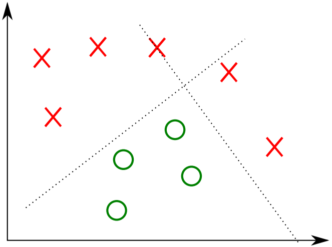


What If the Boundary is Non-linear?

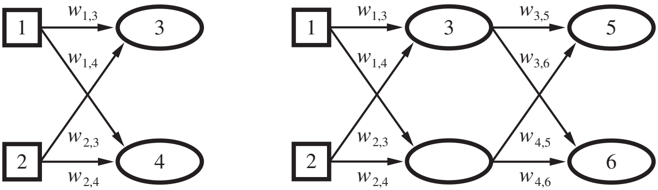


What If the Boundary is Non-linear?

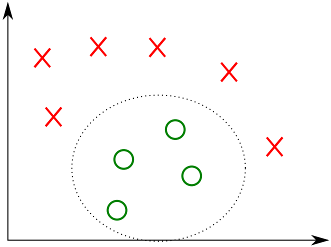




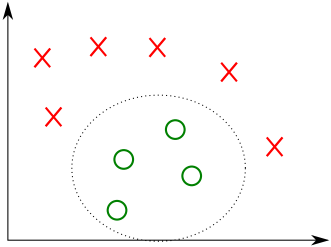
⇒**Multi-Layer Perceptrons**



Another Solution: Non-linear Kernels

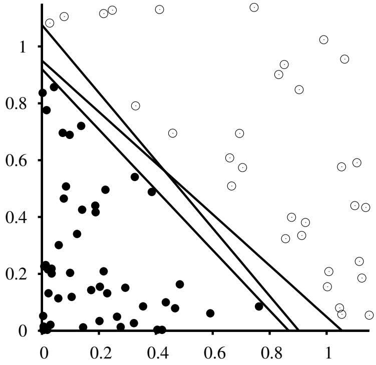


Another Solution: Non-linear Kernels

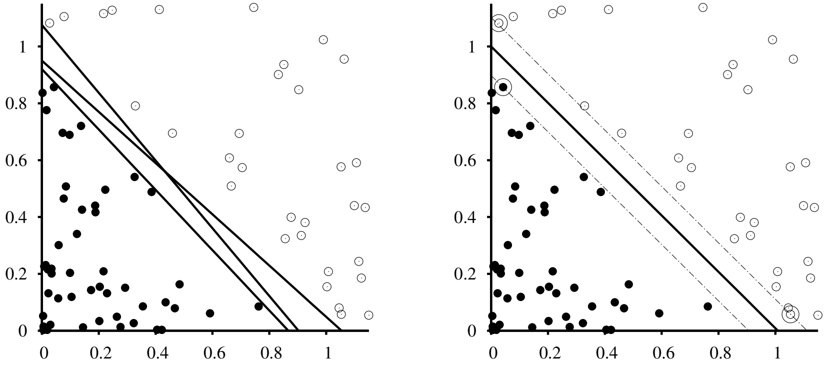


- Convert feature (input) space using non-linear kernel (e.g., radial distance)

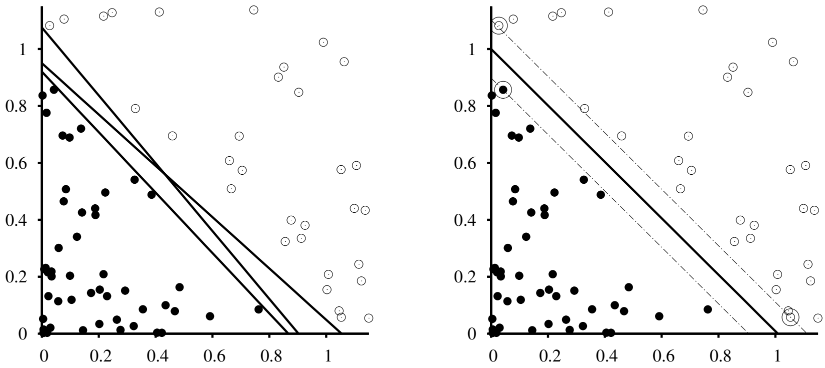
Optimal Boundary? Enter Support Vector Machines



Optimal Boundary? Enter Support Vector Machines

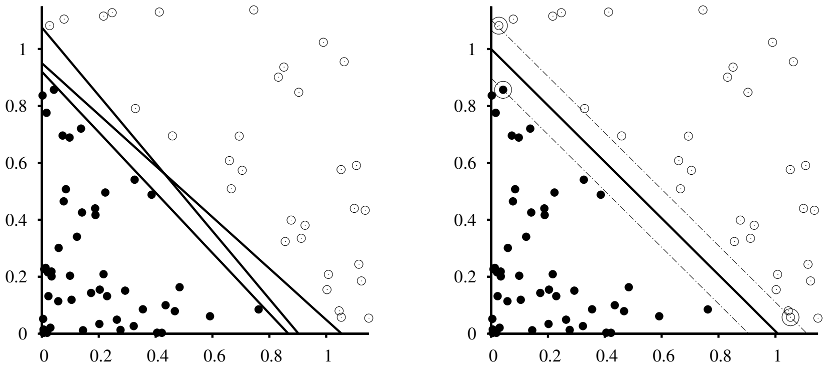


Optimal Boundary? Enter Support Vector Machines



- SVMs are guaranteed to find **optimal** solution ⇒ Statistical Learning Theory

Optimal Boundary? Enter Support Vector Machines



- SVMs are guaranteed to find **optimal** solution ⇒ Statistical Learning Theory
- **Kernel SVMs** are especially powerful because it can search in multi-dimensional kernel space

So Many Methods So Little Time... How to Choose?

- Problem choosing model complexity
- Kernel type
- Structural complexity of MLP or SVM

So Many Methods So Little Time. . . How to Choose?

- Problem choosing model complexity
- Kernel type
- Structural complexity of MLP or SVM

Solution, ask the data:

1. cross-validation
   Divide data into three sets: training, validate, test

So Many Methods So Little Time... How to Choose?

- Problem choosing model complexity
- Kernel type
- Structural complexity of MLP or SVM

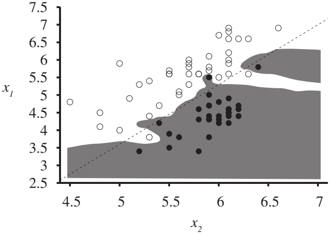
Solution, ask the data:

1. cross-validation
   Divide data into three sets: training, validate, test
2. regularization
   Add complexity minimization term to Loss function

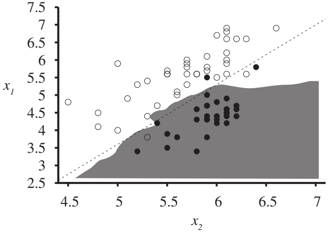
$$\text{Loss} = \sum (y_i - f(x_i))^2 + \beta \times \text{num params}$$

k-Nearest Neighbors algorithm:

- Keep all data points as lookup table
- Smoothing parameter, *k*



(k = 1)

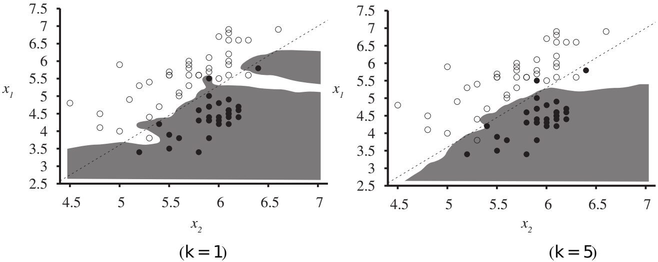


(k = 5)

Or, Get Rid of 'em Altogether: Non-parameric Models

k-Nearest Neighbors algorithm:

- Keep all data points as lookup table
- Smoothing parameter, *k*

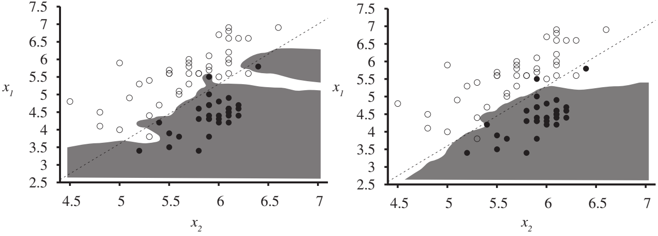


Problems?

Or, Get Rid of 'em Altogether: Non-parameric Models

k-Nearest Neighbors algorithm:

- Keep all data points as lookup table
- Smoothing parameter, *k*

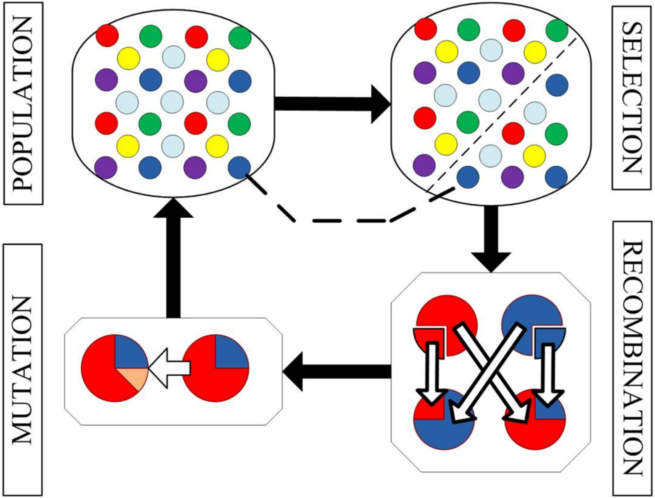


(k = 1) (k = 5)

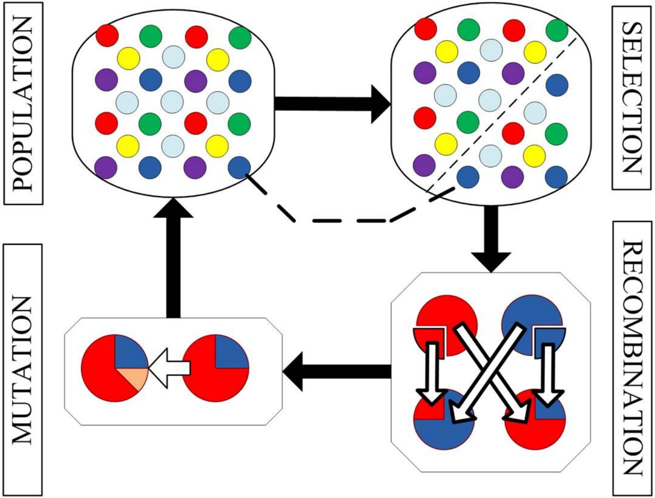
Problems?

- Number of data points
- Number of features

Finally, a Totally Different One: Genetic Algorithms



Finally, a Totally Different One: Genetic Algorithms



Problems:

- No local minima, takes longer, must design problem well

Summary of Supervised Machine Learning

- Can solve problems too complex for man-made algorithms
- Gets better with data (good for information age)
- Supervised learning with labels: regression and classification

Summary of Supervised Machine Learning

- Can solve problems too complex for man-made algorithms
- Gets better with data (good for information age)
- Supervised learning with labels: regression and classification
- Linear regression and Bayes nets calculated from data directly
- Classification by minimizing Loss function iteratively

Summary of Supervised Machine Learning

- Can solve problems too complex for man-made algorithms
- Gets better with data (good for information age)
- Supervised learning with labels: regression and classification
- Linear regression and Bayes nets calculated from data directly
- Classification by minimizing Loss function iteratively
- Local minima is a problem with gradient descent
- Non-linear problems can be solved with multiple boundaries or kernels
- Support vector machines find optimal solution faster

Summary of Supervised Machine Learning

- Can solve problems too complex for man-made algorithms
- Gets better with data (good for information age)
- Supervised learning with labels: regression and classification
- Linear regression and Bayes nets calculated from data directly
- Classification by minimizing Loss function iteratively
- Local minima is a problem with gradient descent
- Non-linear problems can be solved with multiple boundaries or kernels
- Support vector machines find optimal solution faster
- Parameter complexity can be reduced with cross validation and regularization
- Non-parametric models good for low-dimensional problems
- Genetic algorithms have no local minima