Introduction to Artificial Intelligence

Supervised Learning

Jianmin Li

Department of Computer Science and Technology
Tsinghua University

Spring, 2024

Outline

- Classification (Naive Bayes)
- Regression (Linear, Smoother)
- Linear Seperation (Perceptron, SVMs)
- Non-parametric classification (KNN)

Machine Learning: Definition

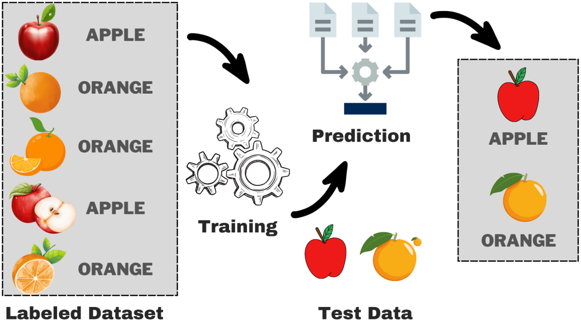
- Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to *learn* without being explicitly programmed.
- Tom Mitchel (1998) Well posed Learning Problem: A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine Learning: Type

What	Parameters	Structure	Hidden concepts	
What from	Supervised	Unsupervised	Reinforcement	Self-supervised
What for	Prediction	Diagnosis	Compression	Discovery
How	Passive	Active	Online	Offline
Output	Classification	Regression	Clustering	
Details	Generative	Discriminative		

Supervised Learning

- Given a training set:
  - $(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_n, y_n)$
  - where each y_i was generated by an unknown $y = f(x)$
- Discover a function h that approximates the true function f.



from https://www.kdnuggets.com/understanding-supervised-learning-theory-and-overview

Classification Example: Spam Filter

- Input: x = email
- Output: y = "spam" or "ham"
- Setup:
  - Get a large collection of example emails, each labeled "spam" or "ham"
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails
- Features: The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: $dd, CAPS
  - Non-text: SenderInContacts
  - ...

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial 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, Iknow 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, but when I plugged it in, hit the power nothing happened.

A Spam Filter

- Naive Bayes spam filter
- Data:
  - Collection of emails, labeled spam or ham
  - Note: someone has to hand label all this data!
  - Split into training, held-out, test sets
- Classifier
  - Learn on the training set
  - (Tune it on a held-out set)
  - Test it on new emails

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial 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, Iknow 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, but when I plugged it in, hit the power nothing happened.

✓

Naive Bayes for Text

- Bag-of-Words Naive Bayes:
  - Predict unknown class label (spam vs. ham)
  - Assume evidence features (e.g. the words) are independent
- Generative model

$$P(C, W_1, W_2, \cdots, W_n) = P(C)\prod_i P(W_i|C)$$

  - W_i, Word at position i, not ith word in the dictionary
- Tied distributions and bag-of-words
  - Usually, each variable gets its own conditional probability distribution $P(F|Y)$
  - In a bag-of-words model
    - Each position is identically distributed
    - All positions share the same conditional probs $P(W|C)$

General Naive Bayes

- General probabilistic model

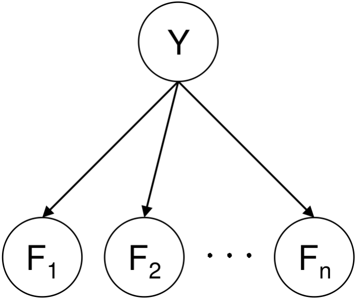
$$P(Y, F_1, \ldots, F_n)$$

  - $|Y| \times |F|^n$ parameters

- General naive Bayes model

$$P(Y, F_1, \ldots, F_n) = P(Y) \prod_i P(F_i | Y)$$

  - $n \times |Y| \times |F|$ parameters
  - We only specify how each feature depends on the class
  - Total number of parameters is linear in n



Example: Spam Filtering

- Model:

$$P(C, W_1, W_2, \cdots, W_n) = P(C)\prod_i P(W_i|C)$$

- What are the parameters?

$P(C)$

ham :	0.66
spam:	0.33

$P(W|\text{spam})$

the :	0.0156
to :	0.0153
and :	0.0115
of :	0.0095
you :	0.0093
a :	0.0086
with:	0.0080
from:	0.0075
...	

$P(W|\text{ham})$

the :	0.0210
to :	0.0133
of :	0.0119
2002:	0.0110
with:	0.0108
from:	0.0107
and :	0.0105
a :	0.0100
...	

- Where do these tables come from? counts from examples!

Spam Example

Gray would you like to lose weight while you sleep.

Word	$P(w\|\text{spam})$	$P(w\|\text{ham})$	Total Spam	Total Ham
(prior)	0.33333	0.66666	-1.1	-0.4

Spam Example

Gray would you like to lose weight while you sleep.

Word	$P(w \mid \text{spam})$	$P(w \mid \text{ham})$	Total Spam	Total Ham
(prior)	0.33333	0.66666	-1.1	-0.4
Gray	0.00002	0.00021	-11.9	-8.9

Spam Example

Gray would you like to lose weight while you sleep.

Word	$P(w\|\text{spam})$	$P(w\|\text{ham})$	Total Spam	Total Ham
(prior)	0.33333	0.66666	-1.1	-0.4
Gray	0.00002	0.00021	-11.9	-8.9
would	0.00069	0.00084	-19.2	-16.0

Spam Example

Gray would you like to lose weight while you sleep.

Word	$P(w \mid \text{spam})$	$P(w \mid \text{ham})$	Total Spam	Total Ham
(prior)	0.33333	0.66666	-1.1	-0.4
Gray	0.00002	0.00021	-11.9	-8.9
would	0.00069	0.00084	-19.2	-16.0
you	0.00881	0.00304	-23.9	-21.8

Spam Example

Gray would you like to lose weight while you sleep.

Word	$P(w\|\text{spam})$	$P(w\|\text{ham})$	Total Spam	Total Ham
(prior)	0.33333	0.66666	-1.1	-0.4
Gray	0.00002	0.00021	-11.9	-8.9
would	0.00069	0.00084	-19.2	-16.0
you	0.00881	0.00304	-23.9	-21.8
like	0.00086	0.00083	-31.0	-28.8
to	0.01517	0.01339	-35.2	-33.2
lose	0.00008	0.00002	-44.6	-44.0
weight	0.00016	0.00002	-53.3	-54.8
while	0.00027	0.00027	-61.6	-63.0
you	0.00881	0.00304	-66.3	-68.8
sleep	0.00006	0.00001	-76.0	-80.3

$$P(\text{spam}|\text{words}) = \frac{e^{-76.0}}{e^{-76.0} + e^{-80.3}} = 98.7\%$$

Example: Overfitting

- Posteriors determined by relative probabilities (odds ratios):

$$\frac{P(W|ham)}{P(W|spam)}$$

south-west	:	inf
nation	:	inf
morally	:	inf
nicely	:	inf
extent	:	inf
seriously	:	inf
...		

$$\frac{P(W|spam)}{P(W|ham)}$$

screens	:	inf
minute	:	inf
guaranteed	:	inf
\$205.00	:	inf
delivery	:	inf
signature	:	inf
...		

Example: Overfitting

- Posteriors determined by relative probabilities (odds ratios):

$$\frac{P(W|ham)}{P(W|spam)}$$

south-west	:	inf
nation	:	inf
morally	:	inf
nicely	:	inf
extent	:	inf
seriously	:	inf
...		

$$\frac{P(W|spam)}{P(W|ham)}$$

screens	:	inf
minute	:	inf
guaranteed	:	inf
$205.00	:	inf
delivery	:	inf
signature	:	inf
...		

What went wrong here?

Generalization and Overfitting

- Raw counts will overfit the training data!
  - Unlike that every occurrence of "minute" is 100% spam
  - Unlike that every occurrence of "seriously" is 100% ham
  - What about all the words that don't occur in the training set at all? 0/0?
  - In general, we can not go around giving unseen events zero probability
- At the extreme, imagine using the entire email as the only feature
  - Would get the training data perfect (if deterministic labeling)
  - Would not generalize at all
  - Just making the bag-of-words assumption gives us some generalization, but is not enough
- To generalize better, we need to smooth or regularize the estimates

Estimation: Smoothing

- Maximum likelihood estimates:

$$P_{ML}(x) = \frac{count(x)}{\text{total samples}}$$

$$P_{ML}(r) = 1/3$$

- Problems with maximum likelihood estimates:
  - If I flip a coin once, and it's head, what's the estimate for P(heads)?
  - What if I flip 10 times with 8 heads?
  - What if I flip 10M times with 8M heads?
- Basic idea:
  - We have some prior expectation about parameters (here, the probability of heads)
  - Given little evidence, we should skew towards our prior
  - Given a lot of evidence, we should listen to the data

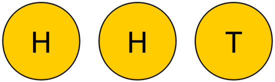
Estimation: Laplace Smoothing

- Laplace's estimate (extended):
  - Pretend you saw every outcome k extra times

$$P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$$

  - What's Laplace with k = 0?
  - k is the strength of the prior
- Laplace for conditionals:
  - Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x,y)+k}{c(y)+k|X|}$$



$$P_{LAP,0}(X) =$$

$$P_{LAP,1}(X) =$$

$$P_{LAP,100}(X) =$$

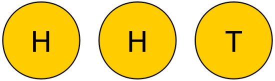
Estimation: Laplace Smoothing

- Laplace's estimate (extended):
  - Pretend you saw every outcome k extra times

$$P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$$

  - What's Laplace with k = 0?
  - k is the strength of the prior
- Laplace for conditionals:
  - Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x,y)+k}{c(y)+k|X|}$$

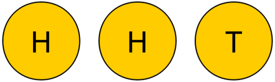


$$P_{LAP,0}(X) = < \frac{2}{3}, \frac{1}{3} >$$

$$P_{LAP,1}(X) =$$

$$P_{LAP,100}(X) =$$

Estimation: Laplace Smoothing



- Laplace's estimate (extended):
  - Pretend you saw every outcome k extra times

$$P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$$

  - What's Laplace with k = 0?
  - k is the strength of the prior
- Laplace for conditionals:
  - Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x,y)+k}{c(y)+k|X|}$$

$$P_{LAP,0}(X) = < \frac{2}{3}, \frac{1}{3} >$$

$$P_{LAP,1}(X) = < \frac{3}{5}, \frac{2}{5} >$$

$$P_{LAP,100}(X) =$$

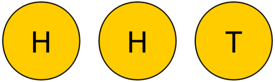
Estimation: Laplace Smoothing

- Laplace's estimate (extended):
  - Pretend you saw every outcome k extra times

$$P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$$

  - What's Laplace with k = 0?
  - k is the strength of the prior
- Laplace for conditionals:
  - Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x,y)+k}{c(y)+k|X|}$$



$$P_{LAP,0}(X) = < \frac{2}{3}, \frac{1}{3} >$$

$$P_{LAP,1}(X) = < \frac{3}{5}, \frac{2}{5} >$$

$$P_{LAP,100}(X) = < \frac{102}{203}, \frac{101}{203} >$$

Estimation: Linear Interpolation

- Another option: linear interpolation
  - Also get P(X) from the data
  - Make sure the estimate of P(X|Y) isn't too different from P(X)

$$P_{LIN}(x|y) = \alpha \hat{P}(x|y) + (1.0 - \alpha)\hat{P}(x)$$

  - What if α is 0? 1?

Real NB: Smoothing

- For real classification problems, smoothing is critical
- New odds ratios:

$$\frac{P(W|ham)}{P(W|spam)}$$

helvetica	:	11.4
seems	:	10.8
group	:	10.2
ago	:	8.4
areas	:	8.3
...		

$$\frac{P(W|spam)}{P(W|ham)}$$

verdana	:	28.8
Credit	:	28.4
ORDER	:	27.2
<FONT>	:	26.9
money	:	26.5
...		

Real NB: Smoothing

- For real classification problems, smoothing is critical
- New odds ratios:

$$\frac{P(W|ham)}{P(W|spam)}$$

helvetica	:	11.4
seems	:	10.8
group	:	10.2
ago	:	8.4
areas	:	8.3
...		

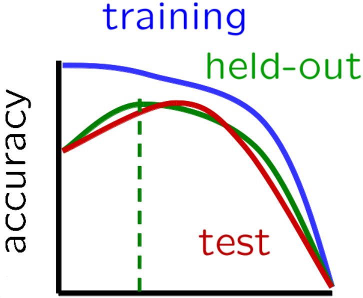
$$\frac{P(W|spam)}{P(W|ham)}$$

verdana	:	28.8
Credit	:	28.4
ORDER	:	27.2
<FONT>	:	26.9
money	:	26.5
...		

Do these make more sense?

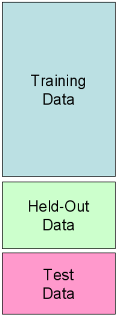
Tuning on Held-Out Data

- Now we' ve got two kinds of unknown parameters:
  - the probabilities $P(Y|X)$, $P(Y)$
  - Hyperparameters, like the amount of smoothing to do: k
- Where to learn?
  - Learn parameters from training data
  - Must tune hyperparameters on different data. Why?
  - For each value of the hyperparameters, train and test on the held-out (validation) data
  - Choose the best value and do a final test on the test data



How to Learn

- Data: labeled instances, e.g. emails marked spam/ham
  - Training set
  - Held out (validation) set
  - Test set
- Features: attribute-value pairs which characterize each x
- Experimentation cycle
  - Learn parameters (e.g. model probabilities) on training set
  - Tune hyperparameters on held-out set
  - Compute accuracy on test set
  - Very important: never "peek" at the test set!
- Evaluation
  - Accuracy: fraction of instances predicted correctly
- Overfitting and generalization
  - Want a classifier which does well on test data
  - Overfitting: fitting the training data very closely, but not generalizing well to test data

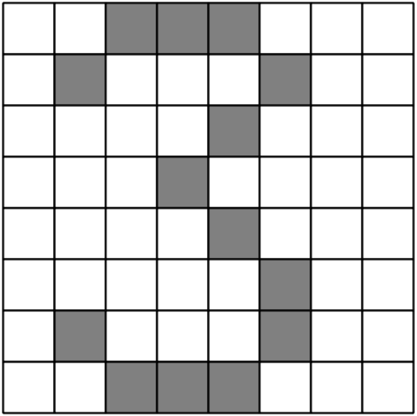


What to Do about Errors?

- Need more features - words aren't enough!
  - Have you emailed the sender before?
  - Have 1K other people just gotten the same email?
  - Is the sending information consistent?
  - Is the email in ALL CAPS?
  - Do inline URLs point where they say they point?
  - Does the email address you by (your) name?
- Can add these information sources as new variables in the Naive Bayes model

A Digit Recognizer

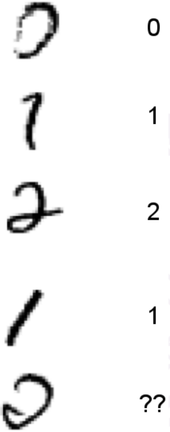
- Input: x = pixel grids



- Output: y = a digit (0-9)

Example: Digit Recognition

- Input: x = pixel grids
- Output: y = a digit 0-9
- Setup:
  - Get a large collection of example images, each labeled with a digit
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future digit images
- Features:
  - The attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape Patterns: NumComponents, AspectRatio, NumLoops
  - ...



Naive Bayes for Digits

- Simple version:
  - One feature F_{ij} for each grid position $< i, j >$
  - Boolean features
  - Each input maps to a feature vector, e.g.

$$\Rightarrow < F_{0,0} = 0, F_{0,1} = 0, F_{0,2} = 1, F_{0,3} = 1, \cdots, F_{15,15} = 0 >$$

- Here: lots of features, each is binary valued Naive Bayes model:

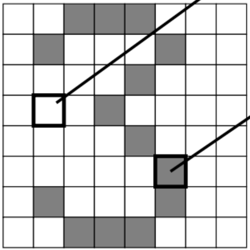
$$P(Y | F_{0,0}, \cdots, F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{ij} | Y)$$

Learning Model Parameters

$$P(Y, F_1, \cdots, F_n) = P(Y)\prod_i P(F_i|Y)$$

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

Problem: Overfitting

$P(\text{features}, C = 2)$

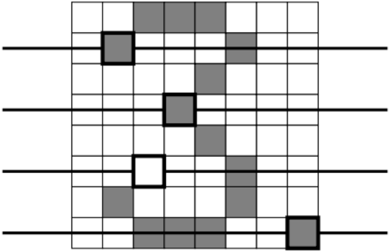
$P(C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.8$

$P(\text{on}|C = 2) = 0.1$

$P(\text{off}|C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.01$



$P(\text{features}, C = 3)$

$P(C = 3) = 0.1$

$P(\text{on}|C = 3) = 0.8$

$P(\text{on}|C = 3) = 0.9$

$P(\text{off}|C = 3) = 0.7$

$P(\text{on}|C = 3) = 0.0$

Problem: Overfitting

$P(\text{features}, C = 2)$

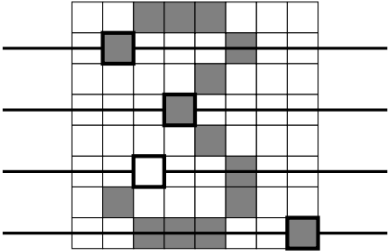
$P(C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.8$

$P(\text{on}|C = 2) = 0.1$

$P(\text{off}|C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.01$



$P(\text{features}, C = 3)$

$P(C = 3) = 0.1$

$P(\text{on}|C = 3) = 0.8$

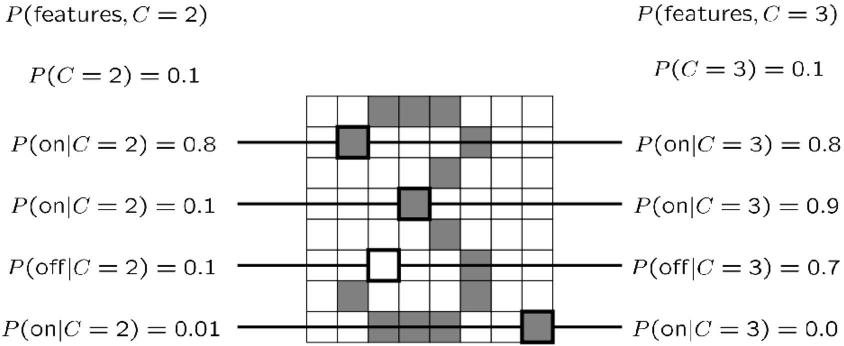
$P(\text{on}|C = 3) = 0.9$

$P(\text{off}|C = 3) = 0.7$

$P(\text{on}|C = 3) = 0.0$

2 Wins!

Problem: Overfitting



2 Wins! X

Outline

- Classification (Naive Bayes)
- Regression (Linear, Smoother)
- Linear Seperation (Perceptron, SVMs)
- Non-parametric classification (KNN)

Regression (Linear, Smoothing)

- Linear model
  - $y = mx + b$
  - $h_w(x) = y = w_1 x + w_0$
- Find best values for parameters
  - "maximize goodness of fit"
  - "maximize probability" or "minimize loss"

Regression: Minimizing Loss

- Assume true function f is given by

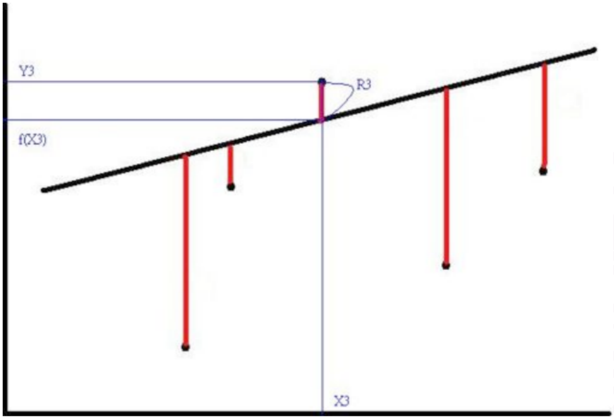
$$y = f(x) = mx + b + noise$$

  - where noise is normally distributed

- Then most probable values of parameters found by minimizing squared-error loss:

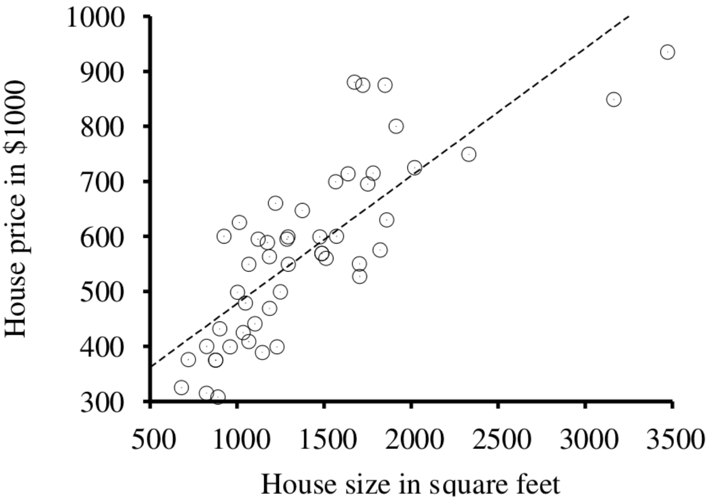
$$Loss(h_w) = \sum_j (y_j\text{-}h_w(x_j))^2$$

Regression: Minimizing Loss

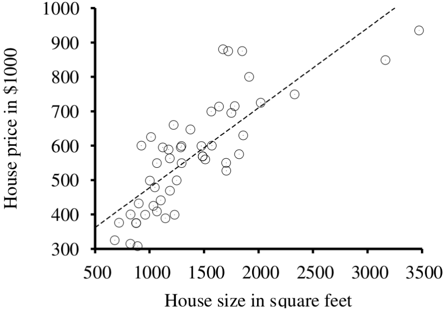


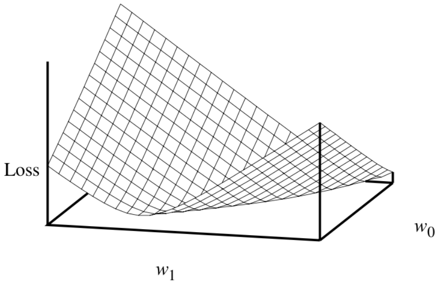
Choose weights to minimize sum of squared errors

Regression: Minimizing Loss



Regression: Minimizing Loss





$$y = w_1 x + w_0$$

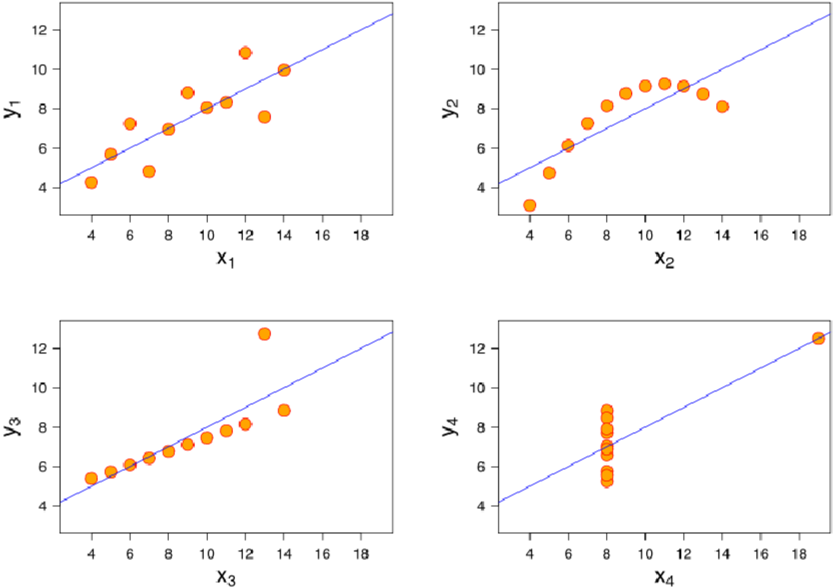
Linear algebra gives an exact solution to the minimization problem

Linear Algebra Solution

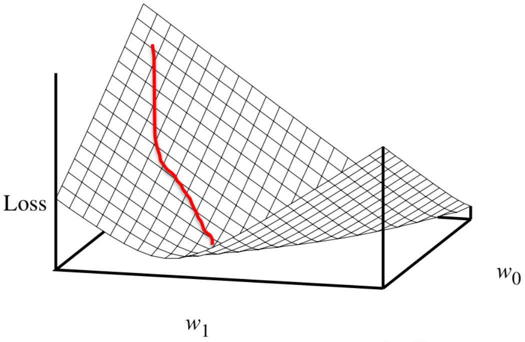
$$w_1 = \frac{M \sum x_i y_i - \sum x_i \sum y_i}{M \sum x_i^2 - \left(\sum x_i\right)^2}$$

$$w_0 = \frac{1}{M} \sum y_i - \frac{w_1}{M} \sum x_i$$

Don't Always Trust Linear Models



Regression by Gradient Descent



w = any point
loop until convergence do:
    for each w_i in w do:
        $w_i = w_i - \alpha \frac{\partial Loss(w)}{w_i}$

Multivariate Regression

- You learned this in math class too

$$h_w(x) = w \cdot x = wx^T = \sum_i w_i x_i$$

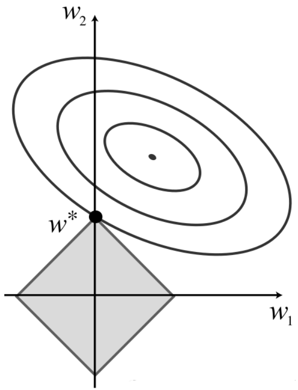
- The most probable set of weights, $w^\star$ (minimizing squared error):

$$w^\star = (X^T X)^{-1} X^T y$$

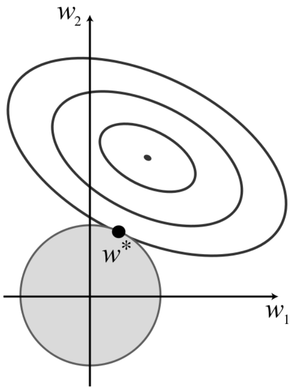
Overfitting

- To avoid overfitting, don't just minimize loss
- Maximize probability, including prior over w
- Can be stated as minimization:
  - $\mathrm{Cost}(h) = \mathrm{EmpiricalLoss}(h) + \lambda \mathrm{Complexity}(h)$
- For linear models, consider
  - $\mathrm{Complexity}(h_w) = L_q(w) = \sum_i |w_i|^q$
  - L_1 regularization minimizes sum of abs. values
  - L_2 regularization minimizes sum of squares

Regularization and Sparsity



L_1 regularization



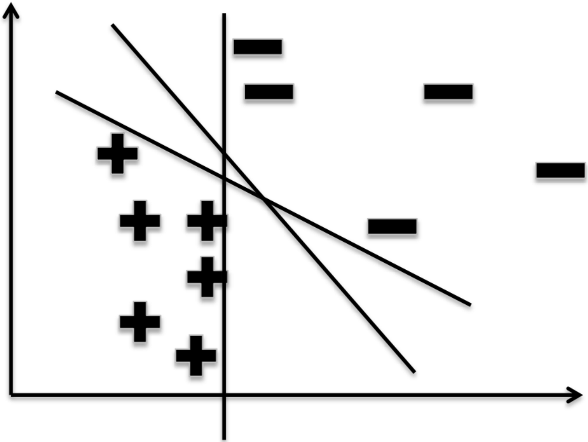
L_2 regularization

$$\mathrm{Cost}(h) = \mathrm{EmpiricalLoss}(h) + \lambda \mathrm{Complexity}(h)$$

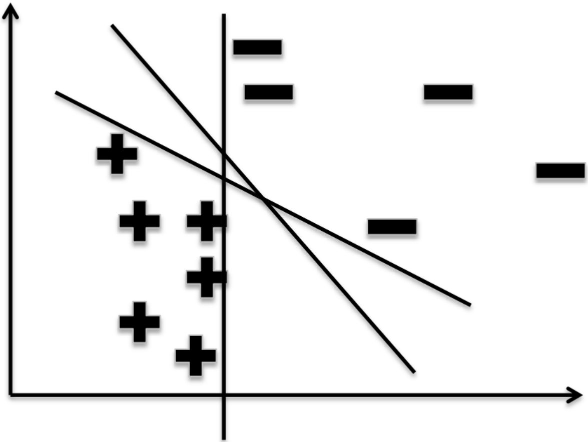
Outline

- Classification (Naive Bayes)
- Regression (Linear, Smoother)
- Linear Seperation (Perceptron, SVMs)
- Non-parametric classification (KNN)

Linear Separator



Perceptron

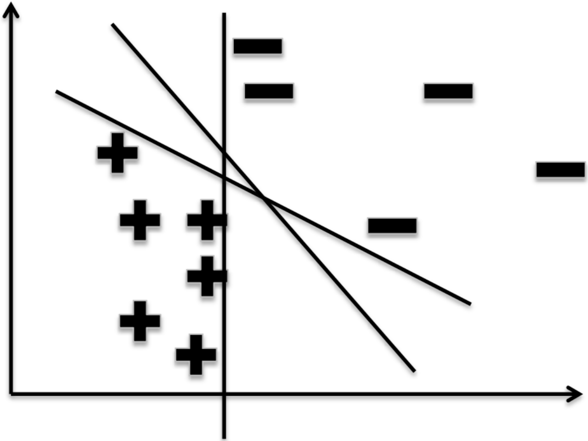


$$f(x) = \begin{cases} 1 & \text{if } w_1 x + w_0 \geq 0 \\ 0 & \text{if } w_1 x + w_0 < 0 \end{cases}$$

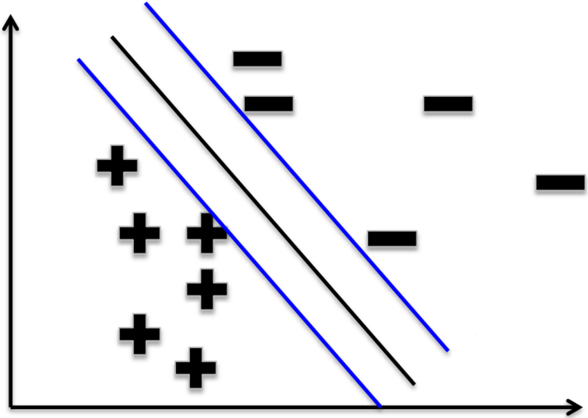
Perceptron Algorithm

- Start with random w_0, w_1
- Pick training example $< x, y >$
- Update (α is learning rate)
  - $w_1 \leftarrow w_1 + \alpha(y - f(x))x$
  - $w_0 \leftarrow w_0 + \alpha(y - f(x))$
- Converges to linear separator (if exists)
- Picks a linear separator (a good one?)

What Linear Separator to Pick?

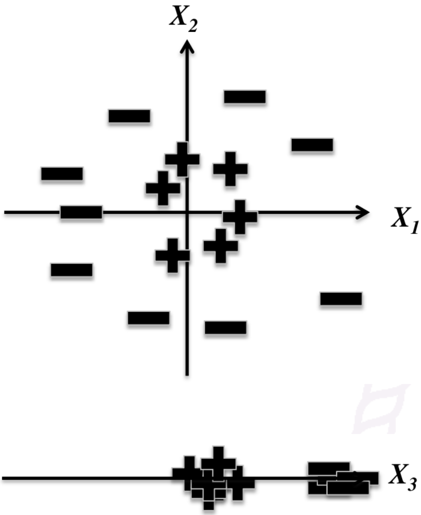


What Linear Separator to Pick?



Maximizes the "margin"$\Longrightarrow$Support Vector Machines

Non-Separable Data?



- Not linearly separable for x_1, x_2
- What if we add a feature?
- $x_3 = x_1^2 + x_2^2$
- Kernel Trick

Outline

- Classification (Naive Bayes)
- Regression (Linear, Smoother)
- Linear Seperation (Perceptron, SVMs)
- Non-parametric classification (KNN)

Nonparametric Models

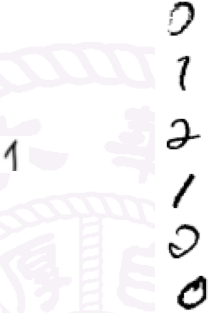
- If the process of learning good values for parameters is prone to overfitting, can we do without parameters?

Nearest-Neighbor Classification

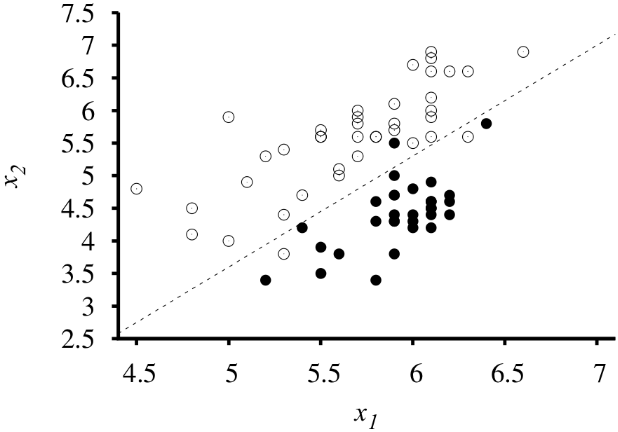
- Nearest neighbor for digits:
  - Take new image
  - Compare to all training images
  - Assign based on closest example
- Encoding: image is vector of intensities:
  -  $\rightarrow < 0.0, 0.0, 0.3, 0.8, 0.7, 0.1, \cdots, 0.0 >$
- What's the similarity function?
  - Dot product of two images vectors

$$sim(x,y) = x \cdot y = \sum_i x_i y_i$$

- Usually normalize vectors so $||x|| = 1$
- min = 0 (when?), max = 1 (when?)

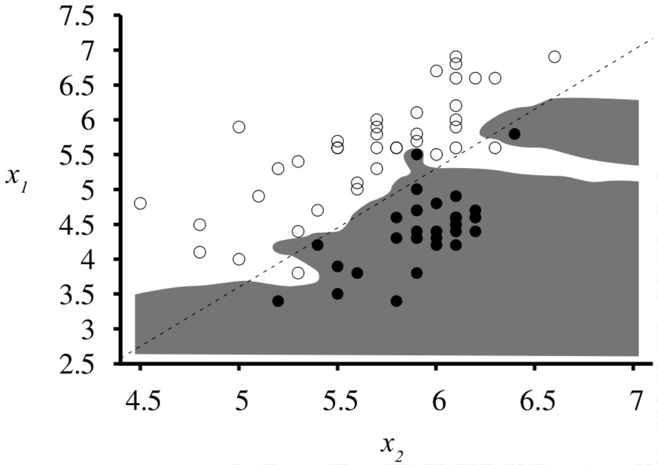


Earthquakes and Nuclear Explosions



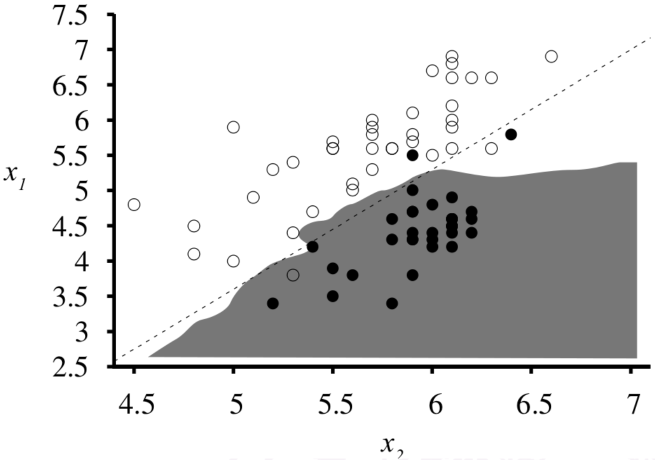
Using logistic regression (similar to linear regression) to do linear classification

K=1 Nearest Neighbors



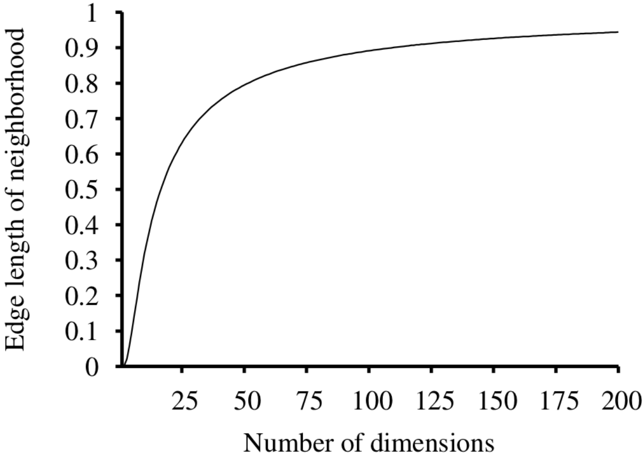
Using nearest neighbors to do classification

K=5 Nearest Neighbors



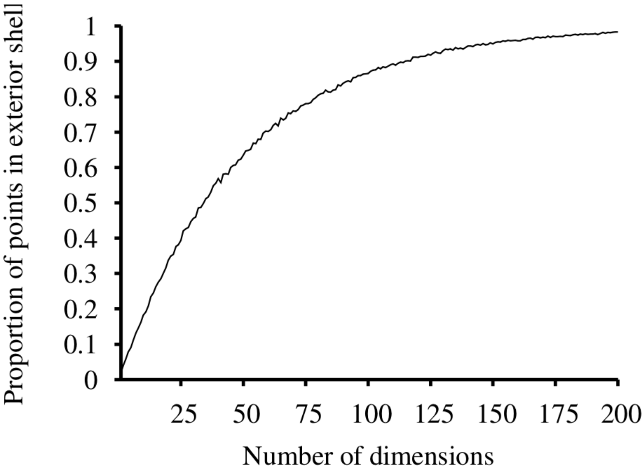
Even with no parameters, you still have hyperparameters!

Curse of Dimension



Average neighborhood size for 10-nearest neighbors, n dimensions, 1M uniform points

Curse of Dimension



Proportion of points that are within the outer shell, 1% of thickness of the hypercube

Summary

- Classification (Naive Bayes)
- Regression (Linear, Smoother)
- Linear Seperation (Perceptron, SVMs)
- Non-parametric classification (KNN)