TP3 Non-Linear Classification:
Decision Trees, SVM

0: Decision Trees

- Using the dataset of the next figure, we want to build a decision tree which classifies Y as T or F given the binary variables A,B,C

(a) Draw the tree that would be learned by the greedy algorithm with zero training error

(b) Is this tree optimal (i.e., does it get zero training error with minimal depth)?

  - If it is not optimal, draw the optimal tree as well

- Hint: Recall the Minimal Description Principle

A	B	C	Y
F	F	F	F
T	F	T	T
T	T	F	T
T	T	T	F

0: Cost of Decision Trees

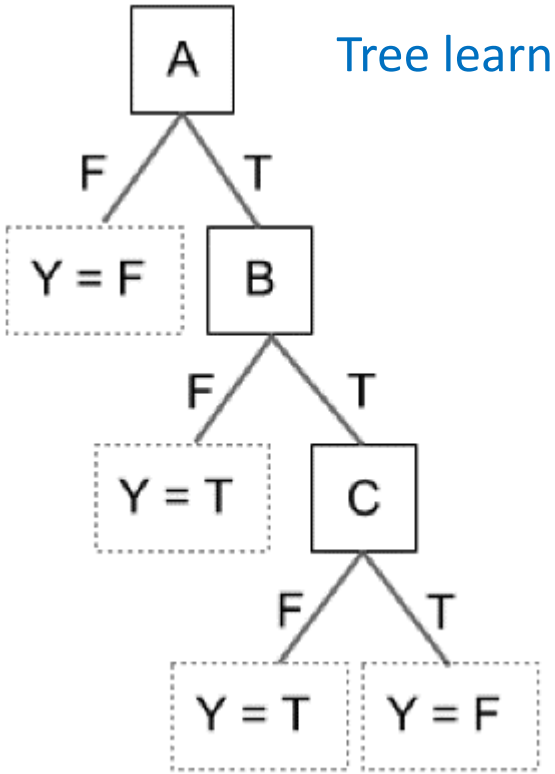
- The total description length of a tree is given by:

  Cost(tree, data) = Cost(tree) + Cost(data|tree)

- Each internal node of the tree is encoded by the ID of the splitting attribute
  - If there are m features, the cost of encoding each feature is $\log_2 m$ bits
- Each leaf is encoded using the ID of the class it is associated with
  - If there are k classes, the cost of encoding a class is $\log_2 k$ bits
- Cost(tree) is the cost of encoding all the nodes in the tree
  - For simplicity assume that the total cost of the tree is obtained by adding up the costs of encoding each internal node and each leaf node
- Cost(data|tree) is encoded using the number of classification errors the tree commits on the training set
  - Each error is encoded by $\log_2 n$ bits, where n is the total # of training examples

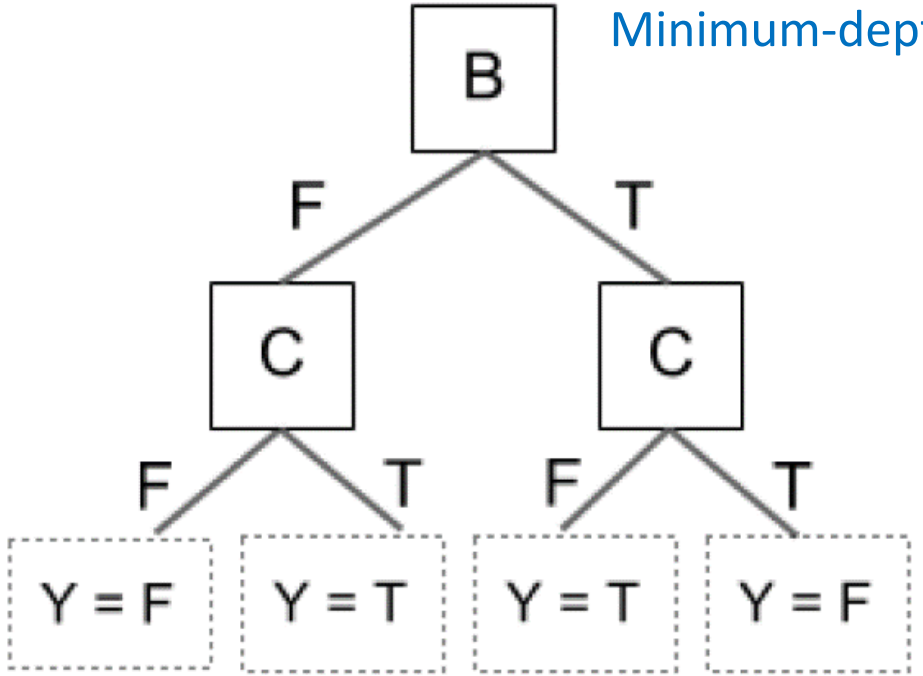
0: Solution

a)



Tree learned by ID3:

Minimum-depth tree:



b) Although we get a better information gain by first splitting on A, Y is just a function of B/C: Y = B **xor** C

- Thus, we can build a tree of depth 2 which classifies correctly and is optimal

0: Decision Trees

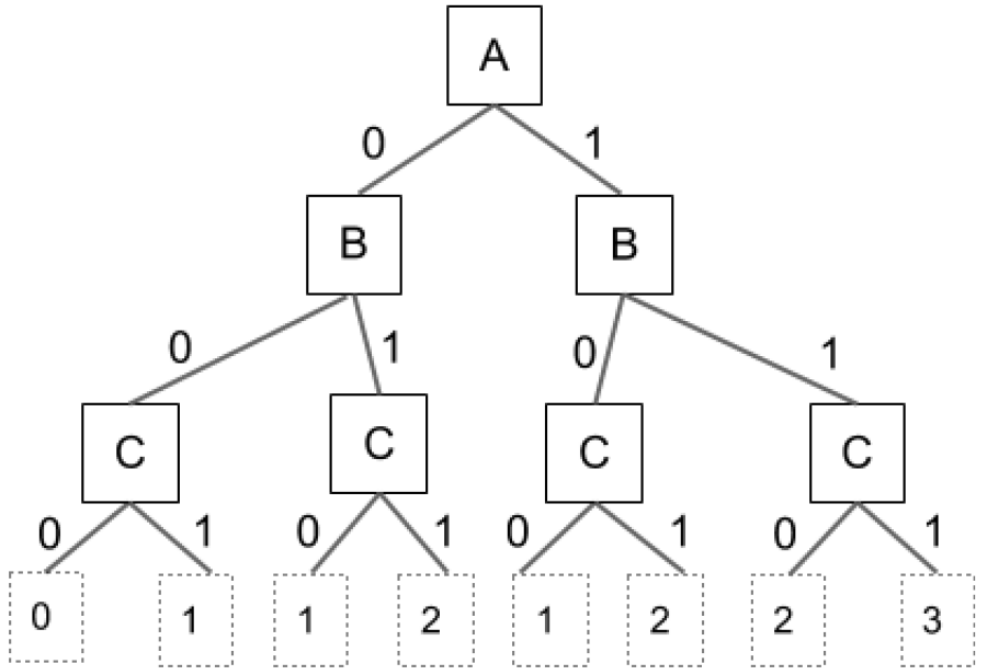
c) In class we used decision trees for classification, but we can use them for regression as well (i.e., learning a function from features to real values)

- Let's imagine that our data has 3 binary features A,B,C, which take values 0/1, and we want to learn a function which counts the number of features which have value 1
- Draw the decision tree which represents this function
  - How many leaf nodes does it have?

A	B	C	Y
0	0	0	0
1	0	0	1
0	1	0	1
…			
1	1	1	3

0: Solutions

c) We can represent the function with a decision tree containing 7 internal nodes and 8 leaves



0 Bonus: Decision Trees (10%)

- Suppose we learned a decision tree from a training set with binary output values (class = 0 or class = 1)
- We find that for a leaf node l,

  (1) there are M training examples falling into it; and

  (2) its entropy is H
- Sketch a simple algorithm which takes as input M and H and that outputs the number of training examples misclassified by leaf node l

1: Splitting Criteria

- Consider the training examples shown in the next Table for a binary classification problem

(a) What is the entropy of this collection of training examples with respect to the positive class?

(b) What are the information gain of a_1 and a_2 relative to these training examples?

(c) For the continuous attribute a_3, compute the information gain for every possible split

(d) What is the best split (among a_1, a_2, and a_3) according to the information gain?

(e) What is the best split (between a_1 and a_2) according to the classification error rate?

(f) What is the best split (between a_1 and a_2) according to the Gini index?

Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

1: Solution

(a) There are four positive examples and five negative examples

- Thus, P(+) = 4/9 and P(−) = 5/9

- The entropy of the training examples is $-4/9 \log_2(4/9) - 5/9 \log_2(5/9) = 0.9911$

(b) For attribute a_1, the corresponding counts are:

a_1	+	-
T	3	1
F	1	4

- The entropy for a_1 is

$$\frac{4}{9}\left[-(3/4)\log_2(3/4) - (1/4)\log_2(1/4)\right] + \frac{5}{9}\left[-(1/5)\log_2(1/5) - (4/5)\log_2(4/5)\right] = 0.7616.$$

- Therefore, the information gain for a_1 is 0.9911 − 0.7616 = 0.2294

- For a_2, the corresponding counts are:

a_2	+	-
T	2	3
F	2	2

- The entropy for a_2 is

$$\frac{5}{9}\left[-(2/5)\log_2(2/5) - (3/5)\log_2(3/5)\right] + \frac{4}{9}\left[-(2/4)\log_2(2/4) - (2/4)\log_2(2/4)\right] = 0.9839.$$

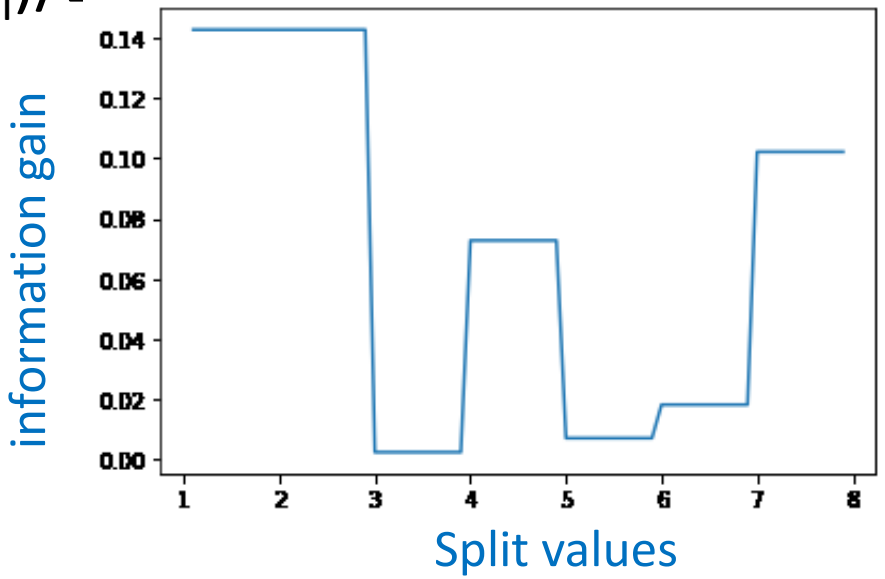
- Therefore, the information gain for a_2 is 0.9911 − 0.9839 = 0.0072

1: Solution

(c) Recall the algorithm for selecting thresholds:

- Sort data according to X into $\{x_1,\ldots,x_m\}$
- Consider split points of the form $x_i + (x_{i+1}-x_i)/2$

a_3	Class label	Split point	Entropy	Info Gain
1.0	+	2.0	0.8484	0.1427
3.0	-	3.5	0.9885	0.0026
4.0	+	4.5	0.9183	0.0728
5.0 5.0	- -	5.5	0.9839	0.0072
6.0	+	6.5	0.9728	0.0183
7.0 7.0	+ -	7.5	0.8889	0.1022



$$H(Class|a_3 : s) = -P(a_3 < s)\,[P(Class_+|a_3 < s)\log_2 P(Class_+|a_3 < s) + P(Class_-|a_3 < s)\log_2 P(Class_-|a_3 < s)]$$
$$- P(a_3 \geq s)\,[P(Class_+|a_3 \geq s)\log_2 P(Class_+|a_3 \geq s) + P(Class_-|a_3 \geq s)\log_2 P(Class_-|a_3 \geq s)]$$

$$IG(a_3 : s) = H(Class) - H(Class|a_3 : s)$$

- The best split for a_3 occurs at split point equals to 2

(d) According to information gain, a_1 produces the best split

1: Solution

(d) The error rate for the data without partitioning on any attribute is

$\varepsilon_{rate}(D) = \min \{P(Class_+), P(Class_-)\} = \min \{4/9, 5/9\} = 4/9$

- After splitting on attribute a_1
- For a_1: error rate = 2/9

	+	-
T	3	1
F	1	3

$\varepsilon_{rate}(a_1) = \min\{P(Class_+ \& a_1 = F), P(Class_- \& a_1 = F)\} + \min\{P(Class_+ \& a_1 = T), P(Class_- \& a_1 = T)\} = \min\{1/9; 3/9\} + \min\{3/9; 1/9\} = 2/9$

- After splitting on attribute a_2
- For a_2: error rate = 4/9

	+	-
T	2	3
F	2	2

$\varepsilon_{rate}(a_2) = \min\{P(Class_+ \& a_2 = F), P(Class_- \& a_2 = F)\} + \min\{P(Class_+ \& a_2 = T), P(Class_- \& a_2 = T)\} = \min\{2/9; 2/9\} + \min\{2/9; 3/9\} = 4/9$

- Therefore, according to error rate, a_1 produces the best split

1: Solution

(e) Recall the Gini impurity of a tree :

$$G^T(S) = \frac{|S_L|}{|S|} G^T(S_L) + \frac{|S_R|}{|S|} G^T(S_R)$$

left substree

right substree

where $G(S) = \sum_{k=1}^{c} p_k(1-p_k)$ = 1 - $\sum_{k=1}^{c} p_k^2$

- For a_1, the gini index is

$$\frac{4}{9}\left[1-(3/4)^2-(1/4)^2\right] + \frac{5}{9}\left[1-(1/5)^2-(4/5)^2\right] = 0.3444.$$

- For a_2, the gini index is

$$\frac{5}{9}\left[1-(2/5)^2-(3/5)^2\right] + \frac{4}{9}\left[1-(2/4)^2-(2/4)^2\right] = 0.4889.$$

- Since the gini index for a_1 is smaller, it produces the better split

2: Maximal Margin Classifier

Obs.	X_1	X_2	Y
1	3	4	Red
2	2	2	Red
3	4	4	Red
4	1	4	Red
5	2	1	Blue
6	4	3	Blue
7	4	1	Blue

- We explore a *maximal margin linear classifier* on a training dataset
  - We are given n=7 observations in p=2 dimensions
  - For each observation there is an associated class label
  - The scatter plot of the observations is given in the next Figure

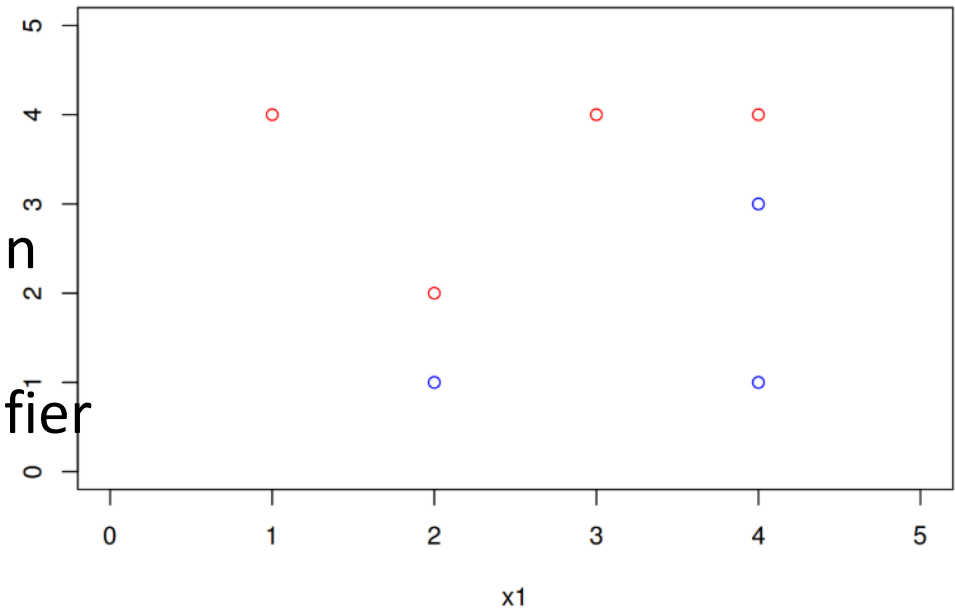
(a) Sketch the optimal separating hyperplane and provide the equation for this hyperplane

(b) Describe the classification rule for the maximal margin classifier

- It should be something along the lines of "Classify to Red if $\beta_0+\beta_1X_1+\beta_2X_2>0$ and classify to Blue otherwise"
- Provide the values for β_0, β_1 and β_2

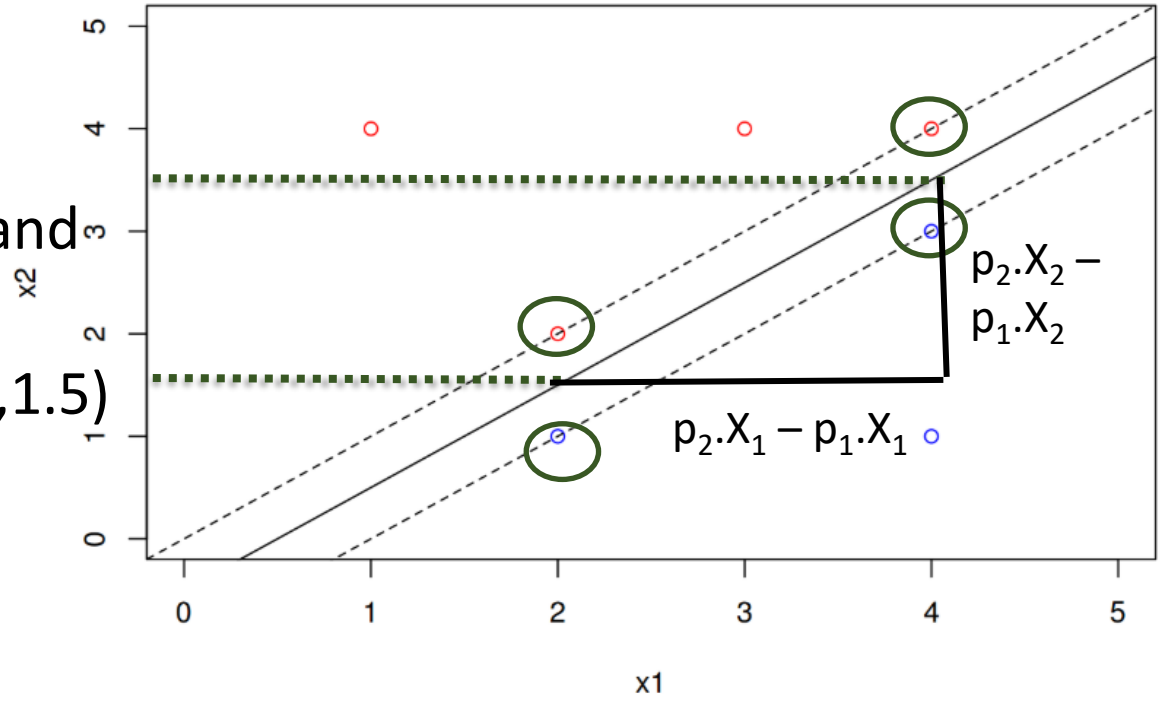
(c) On your sketch, indicate the margin for the maximal margin hyperplane

(d) Indicate the support vectors for the maximal margin classifier



2: Solution

(a) As shown in the plot, the optimal separating hyperplane must lie between the observations (2,1) and (2,2) and between the observations (4,3) and (4,4)



- So, it is a line that passes through the points p_1=(2,1.5) and p_2=(4,3.5) which equation is **$X_1 - X_2 - 0.5 = 0$**
  - Find the slope of the line: $\nabla X_2 / \nabla X_1$ =

    $p_2.X_2 - p_1.X_2 \,/\, p_2.X_1 - p_1.X_1 = 2/2 = 1$
  - Find the intercept: Plug one of the points p_1 in the equation
    $X_2 - 1.5 = 1\,(X_1 - 2) \Rightarrow X_2 = X_1 - 0.5 \Rightarrow X_1 - X_2 - 0.5 = 0$

(b) The rule is Classify to Red if $X_1 - X_2 - 0.5 < 0$, and Classify to Blue otherwise

(c) The margin here is equal to .25

(d) The support vectors are the points (2,1), (2,2), (4,3) and (4,4)

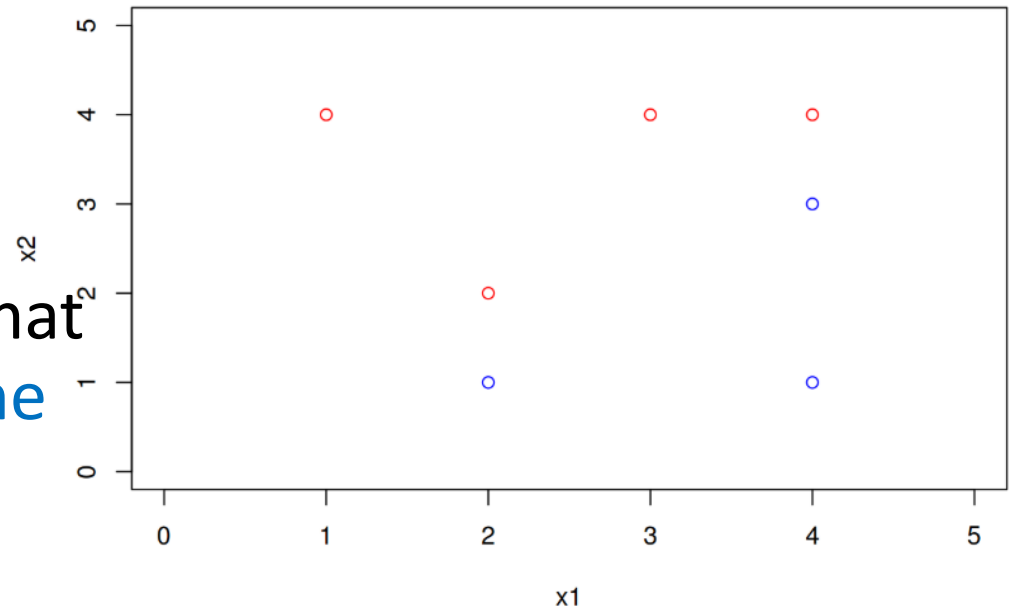
2: Maximal Margin Classifier

- We explore a *maximal margin linear classifier* on a training dataset
  - We are given n=7 observations in p=2 dimensions
  - For each observation there is an associated class label
  - The scatter plot of the observations is given in the next Figure

Obs.	X_1	X_2	Y
1	3	4	Red
2	2	2	Red
3	4	4	Red
4	1	4	Red
5	2	1	Blue
6	4	3	Blue
7	4	1	Blue

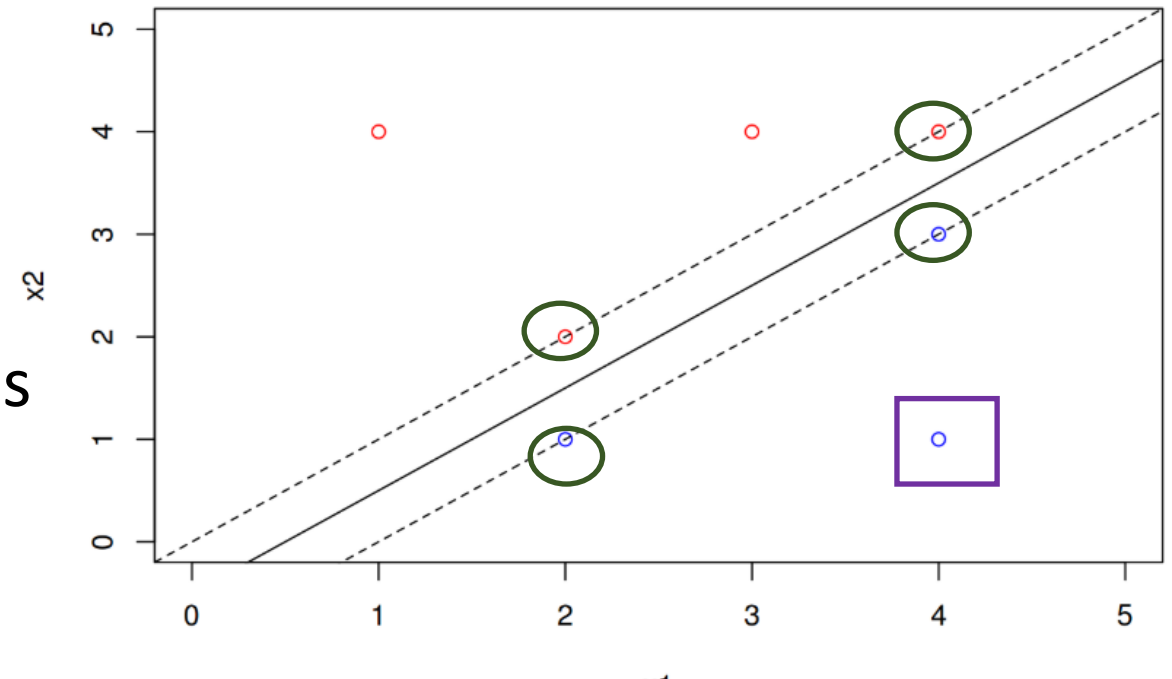
(e) Argue that a slight movement of the seventh observation would not affect the maximal margin hyperplane

(f) Draw an additional observation on the plot so that two classes are no longer separable by a hyperplane

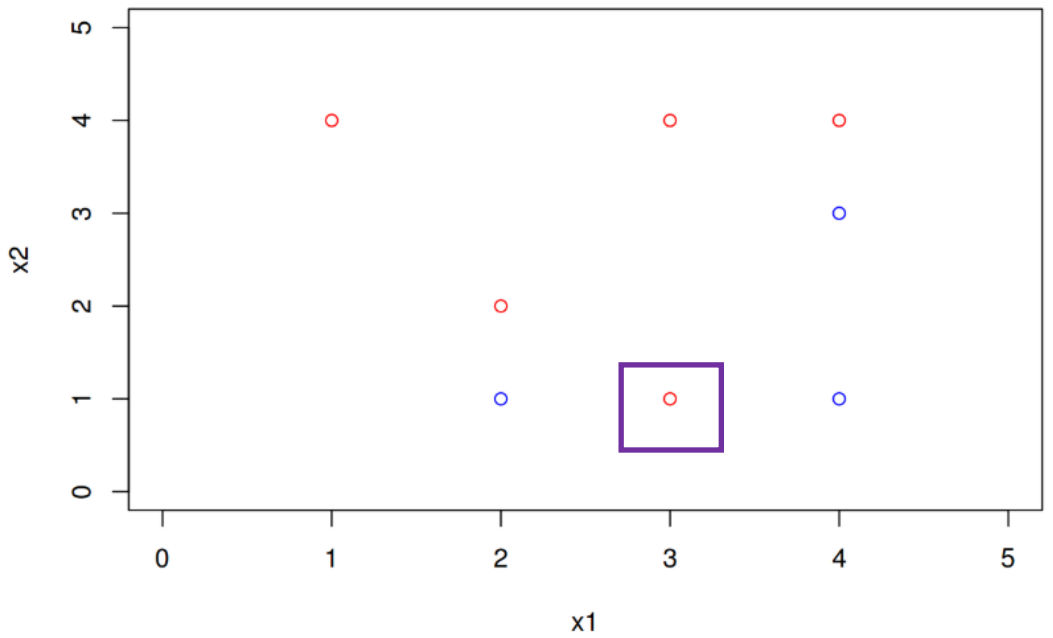


2: Solution

(e) By examining the plot, it is clear that if we moved the observation (4,1), we would not change the maximal margin hyperplane as it is not a support vector



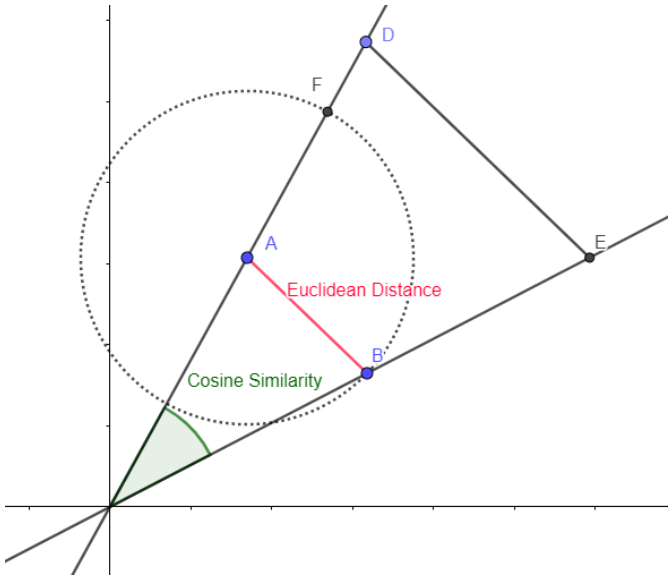
(f) When the red point (3,1) is added to the plot, the two classes are obviously not separable by a linear hyperplane anymore



3: Nearest Neighbor Classification

- We mentioned that the Euclidean/L2 distance is often used as the default distance for nearest neighbor (NNC) classification
  - It is defined as $$E(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||_2 = \sqrt{\sum_{d=1}^{D}(x_d - x'_d)^2}$$
- In some applications such as information retrieval, the cosine distance is widely used too
  - It is defined as $$C(\mathbf{x}, \mathbf{x}') = 1 - \frac{\mathbf{x}^\intercal \mathbf{x}'}{||\mathbf{x}||_2 ||\mathbf{x}'||_2} = 1 - \frac{\sum_{d=1}^{D}(x_d \cdot x'_d)}{||\mathbf{x}||_2 ||\mathbf{x}'||_2}$$

  where the L2 norm of x is defined as $$||\mathbf{x}||_2 = \sqrt{\sum_{d=1}^{D} x_d^2}.$$



- Show that, if data is normalized with unit L2 norm, that is, $\|\mathbf{x}\|_2 = 1$ for all **x** in the training and test sets, changing the distance function from the Euclidean distance to the cosine distance will NOT affect the NNC classification results

3: Feature Scaling

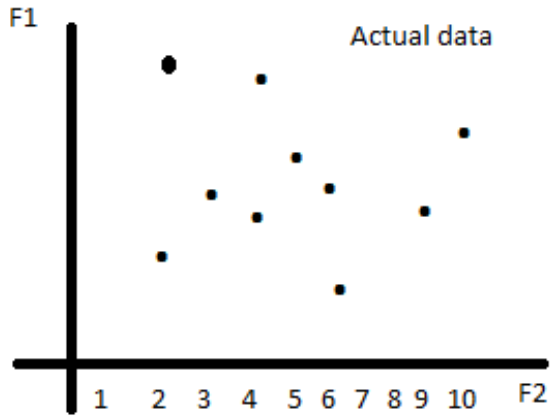
- Standardize the range of independent variables (features of data)
- Standardization or Z-score normalization
  - Rescale the data so that the mean is zero and the standard deviation from the mean (standard scores) is one

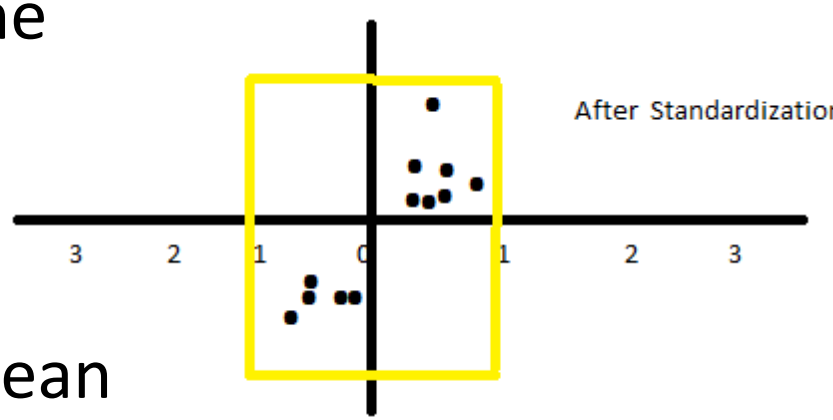
$$X_{stand} = X - \mu / \sigma$$

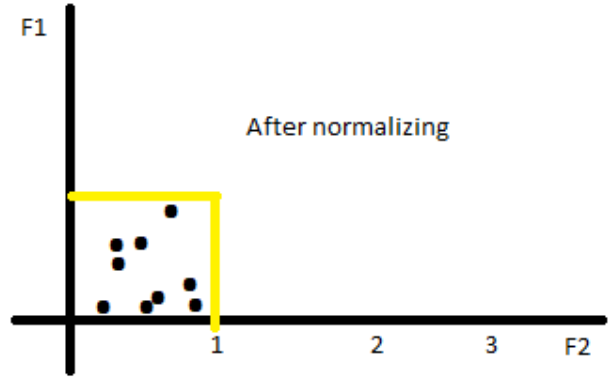
μ is mean, σ is a standard deviation from the mean

- Min-Max normalization
  - Scale the data to a fixed range – between 0 and 1

$$X_{norm} = X - X_{min} / X_{max} - X_{min}$$







3: Solution

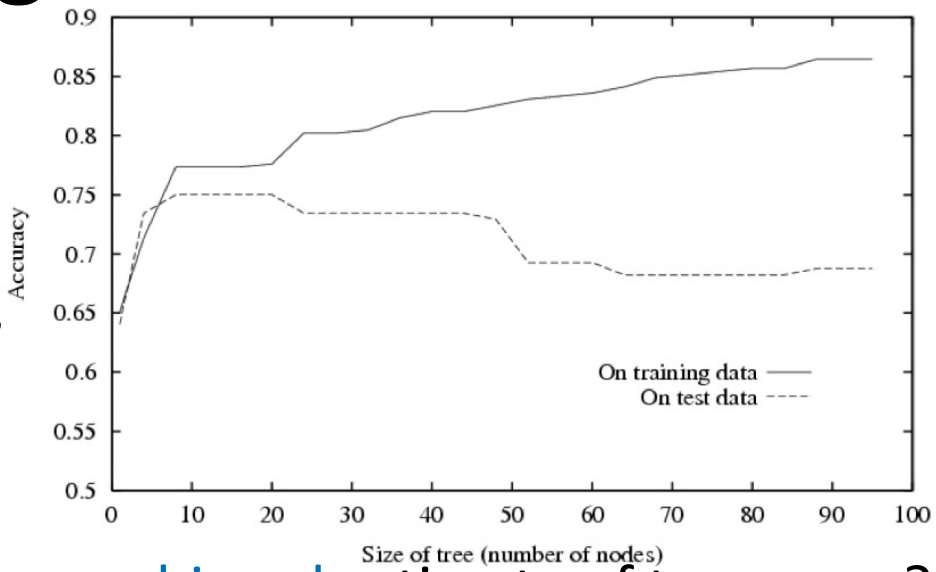
- When data is normalized, we have

$$E(\mathbf{x}, \mathbf{x}')^2 = \sum_{d=1}^{D}(x_d - x'_d)^2 = \sum_{d=1}^{D}(x_d^2 + {x'_d}^2 - 2x_d x'_d) =$$
$$2(1 - \sum_{d=1}^{D}(x_d \cdot x'_d)) = 2C(\mathbf{x}, \mathbf{x}')$$

- Therefore, the nearest neighbor of a point in terms of the cosine distance is exactly the same as its nearest neighbor in terms of the L2 distance, which means the prediction of NNC remains the same

4: Overfitting and PAC Learning

- Consider the training set accuracy and test set accuracy curves plotted above, during decision tree learning, as the number of nodes in the decision tree grows
- This decision tree is being used to learn a function f : X -> Y, where training and test set examples are drawn independently at random from an underlying distribution P(X), after which the trainer provides a noise-free label Y



(a) Does the training error at each point on this curve provide an unbiased estimate of true error?

(b) Does the test error at each point on this curve provide an unbiased estimate of true error ?

(c) Does the training accuracy minus the test accuracy provides an unbiased estimate of the degree of overfitting?

(d) Does the variance in test accuracy will increase as we increase the number of test examples?

(e) Given the above plot of training and test accuracy, which size of the decision tree would you choose to use for classifying future examples?

(f) What is the amount of overfitting in the tree you selected?

4: Solution

(a) The training error is an optimistically biased estimate of the true error, because the hypothesis was chosen based on its fit to the training examples

(b) The expected value of test error (taken over different draws of random test sets) is equal to true error

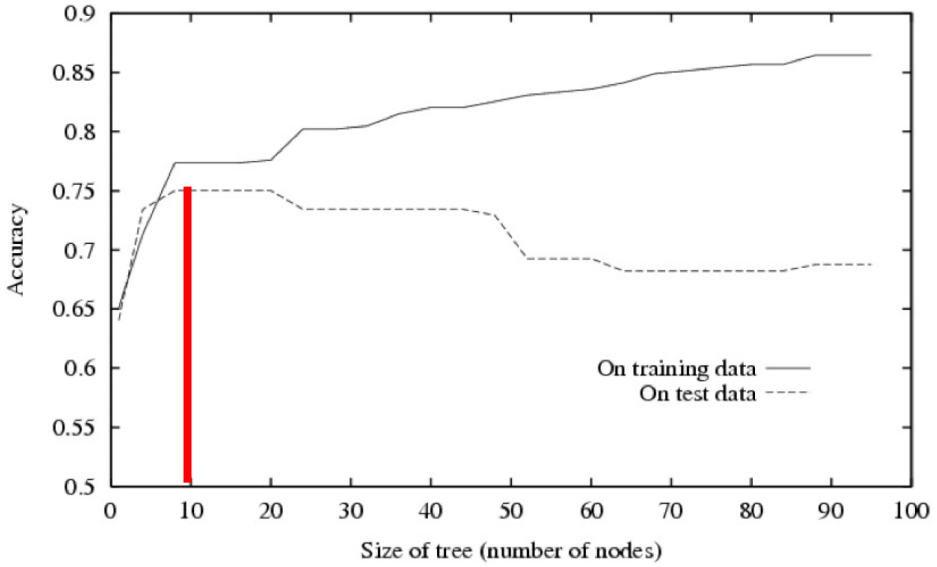
(c) We defined overfitting as test error minus training error, which is equal to the training accuracy minus test accuracy

(d) The variance in test accuracy will decrease as we increase the size of the test set

(e) The simple tree with 10 nodes

- This has the highest test accuracy of any of the trees, and hence the highest expected true accuracy

(f) overfitting = training accuracy minus test accuracy
= 0.77 - 0.74 = 0.03



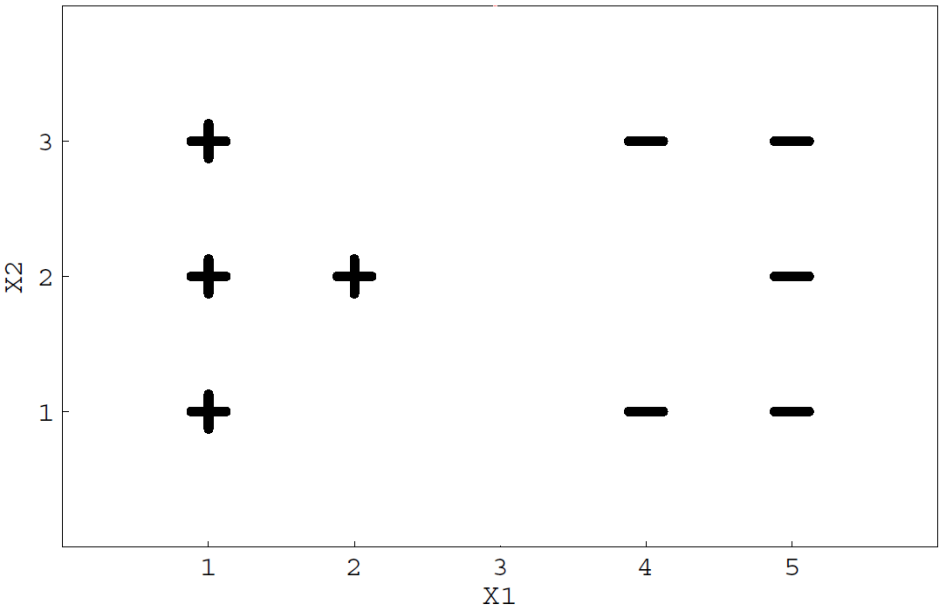
5: Linear SVM

- Suppose we are using a linear SVM, with some large C value, to build a classifier for the data set depicted in the next Figure

(a) Draw the decision boundary of linear SVM

(b) Which examples when removed from the training set will result to a different decision boundary, than training SVN on the full dataset?

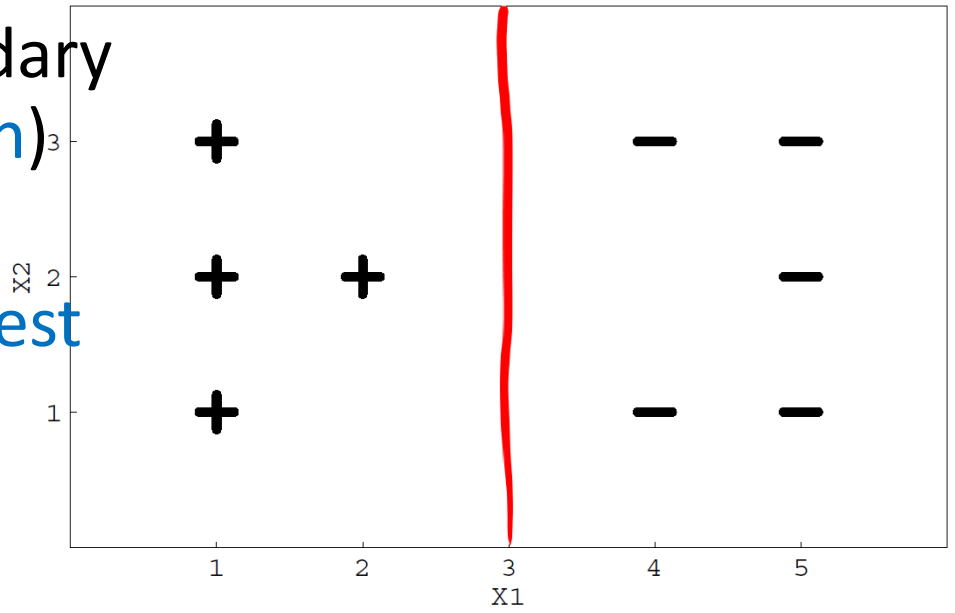
(c)



5: Solution

(a) Because of the large C value, the decision boundary will classify all the examples correctly (~hard margin)

- Among hyperplanes that classify the examples correctly, it will have the largest distance to closest examples

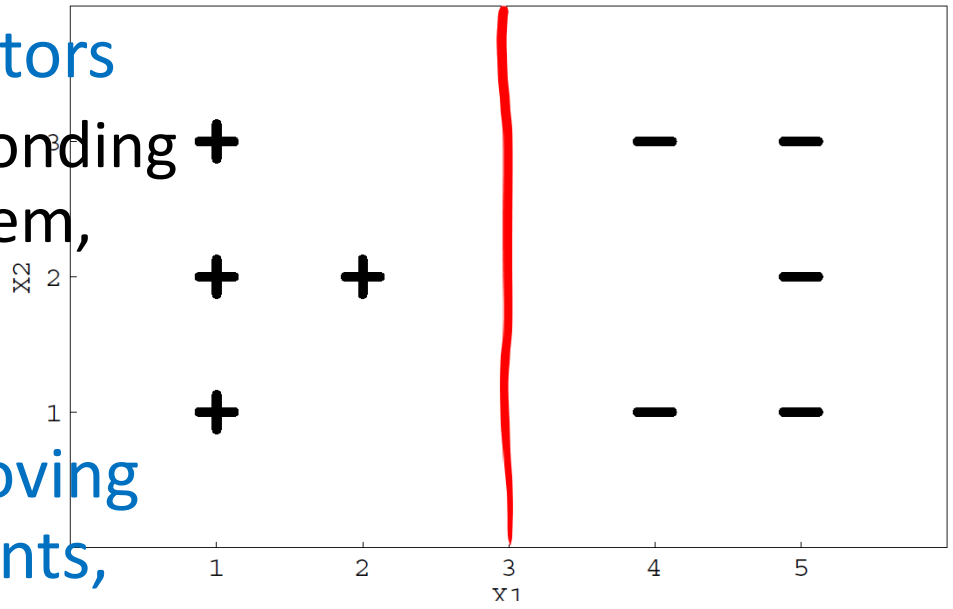


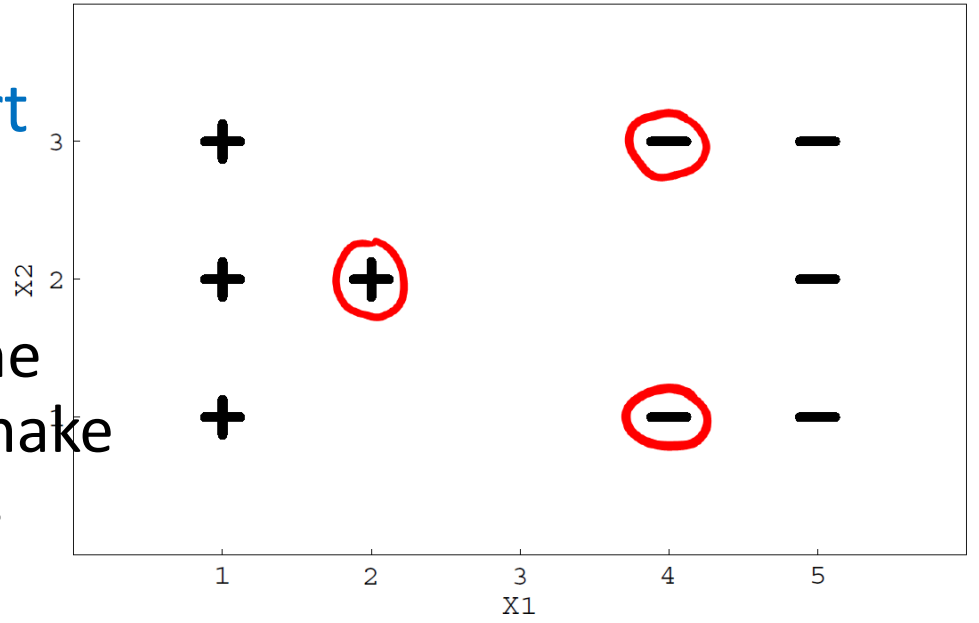
5: Solution

(b) The closest three examples are the support vectors

- all the other examples are such that their corresponding constraints are not tight in the optimization problem, so removing them will not create a solution with smaller objective function value (i.e., norm of w)

- These 3 examples are positioned such that removing any one of them introduces slack in the constraints, allowing for a solution with a smaller objective function value and with a different third support vector
  - in this case, because each of these new (replacement) support vectors is not close to the old separator, the decision boundary shifts to make its distance to that example equal to the others



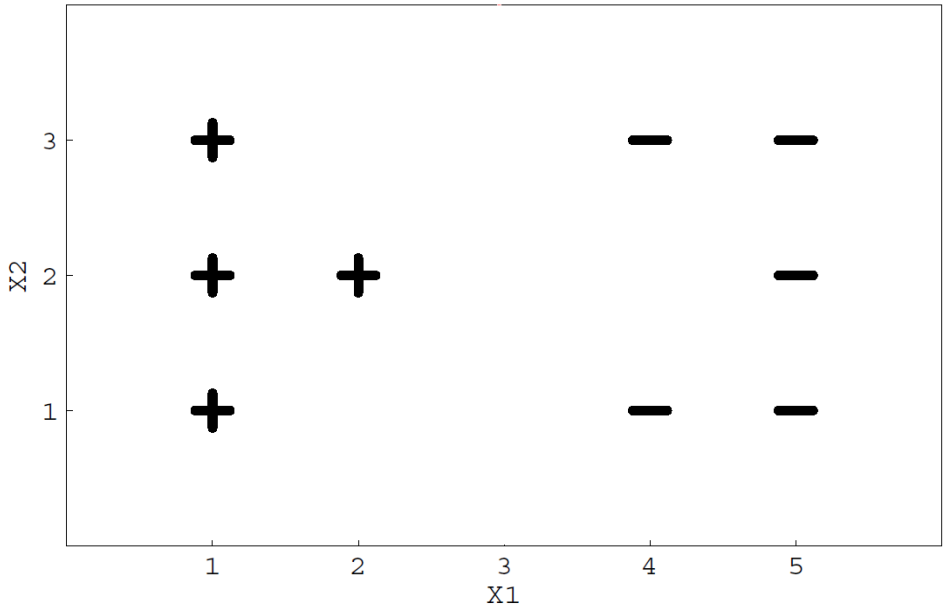


5: Linear SVM

(c) Suppose instead of SVM, we use regularized logistic regression to learn the classifier. That is,

$$(w, b) = \arg \min_{w \in \mathbb{R}^2, b \in \mathbb{R}} \frac{\|w\|^2}{2} - \sum_i \mathbb{1}[y^{(i)} = 0] \ln \frac{1}{1 + e^{(w \cdot x^{(i)} + b)}} + \mathbb{1}[y^{(i)} = 1] \ln \frac{e^{(w \cdot x^{(i)} + b)}}{1 + e^{(w \cdot x^{(i)} + b)}}$$

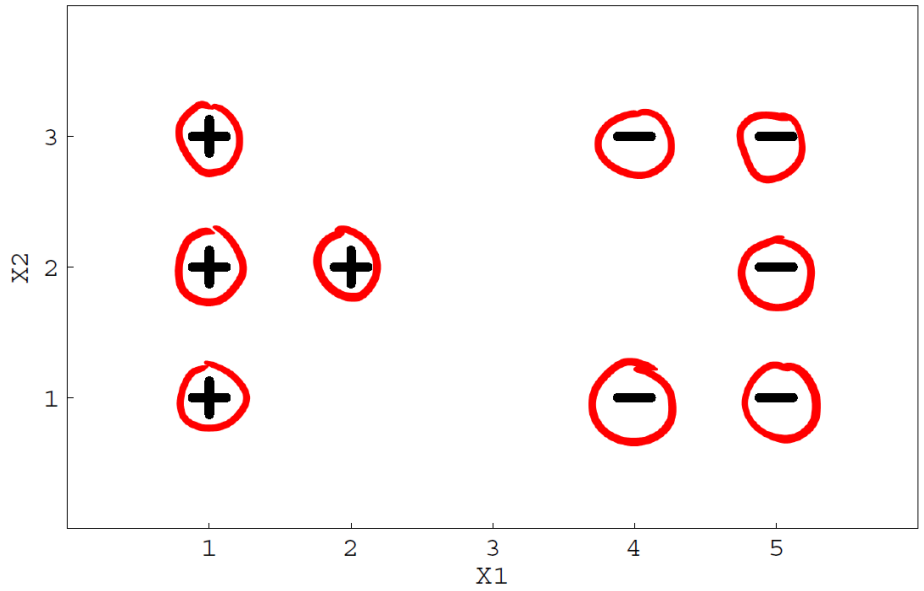
- Which examples when removed from the training set will result to a different decision boundary, than running regularized logistic regression on the full dataset?



5: Solution

(c) Because of the regularization, the weights will not diverge to infinity, and thus the probabilities at the solution are not at 0 and 1

- Because of this, every example contributes to the loss function, and thus has an influence on the solution



5: Linear SVM

(d)Suppose we have a kernel *K(. , .)*, such that there is an implicit high-dimensional feature map ϕ: R^d -> R^D that sa $\forall x, z \in \mathbb{R}^d, \; K(x,z) = \phi(x) \cdot \phi(z)$, where $\phi(x) \cdot \phi(z) = \sum_{i=1}^{D} \phi(x)_i \phi(z)_i$ is the dot product in the D-dimensional space

- Show how to calculate the Euclidean distance in the D-dimensional space without explicitly calculating the values in the D-dimensional vectors:

$$\|\phi(x) - \phi(z)\| = \sqrt{\sum_{i=1}^{D} (\phi(x)_i - \phi(z)_i)^2}$$

5: Solution

(d)

$$\|\phi(x) - \phi(z)\| = \sqrt{\sum_{i=1}^{D} (\phi(x)_i - \phi(z)_i)^2}$$

$$= \sqrt{\sum_{i=1}^{D} \phi(x)_i^2 + \phi(z)_i^2 - 2\phi(x)_i \phi(z)_i}$$

$$= \sqrt{\left(\sum_{i=1}^{D} \phi(x)_i^2\right) + \left(\sum_{i=1}^{D} \phi(z)_i^2\right) - \left(\sum_{i=1}^{D} 2\phi(x)_i \phi(z)_i\right)}$$

$$= \sqrt{\phi(x) \cdot \phi(x) + \phi(z) \cdot \phi(z) - 2\phi(x) \cdot \phi(z)}$$

$$= \sqrt{K(x,x) + K(z,z) - 2K(x,z)}$$

6 Bonus: Quadratic SVM (35 pts)

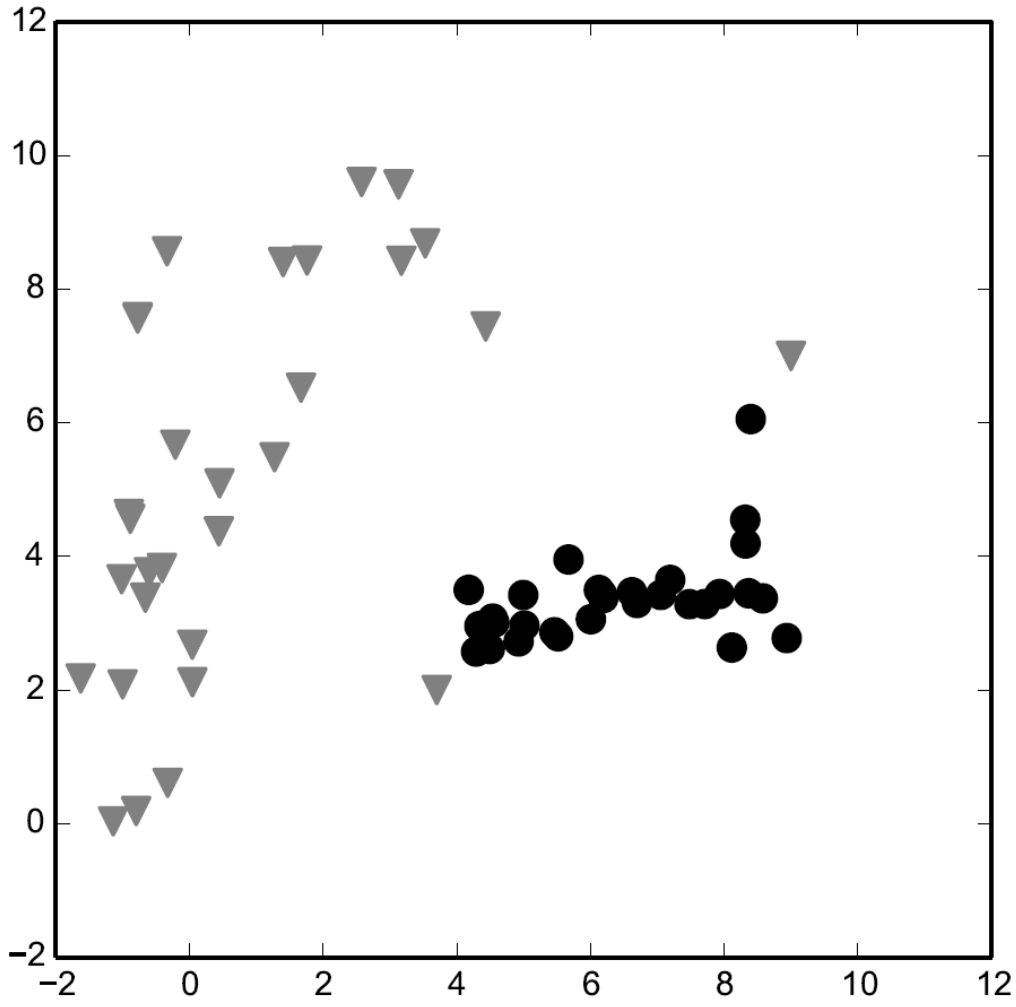
- We are given the 2-D training dataset D shown in the next Figure for a binary classification problem (circles vs. triangles)
- Assume we are using an SVM with a quadratic kernel – that is, our kernel function is a polynomial kernel of degree 2
  - Let C be the cost parameter of the SVM
- Assuming $D = \{x^i, y^i\}_{i=1,\ldots,n}$ with x in R^2 and y in {-1, +1}, recall that the SVM is solving the following optimization problem:

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^p, b\in\mathbb{R}} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{n}\xi_i \quad \text{such that}$$

$$y^i\left(\langle \boldsymbol{w}, \phi(\boldsymbol{x}^i)\rangle + b\right) \geq 1 - \xi_i \quad \text{for all } i = 1, \ldots, n$$

$$\xi_i \geq 0 \quad \text{for all } i = 1, \ldots, n$$

where ϕ is such that $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}')\rangle = (\langle \boldsymbol{x}, \boldsymbol{x}'\rangle + 1)^2$

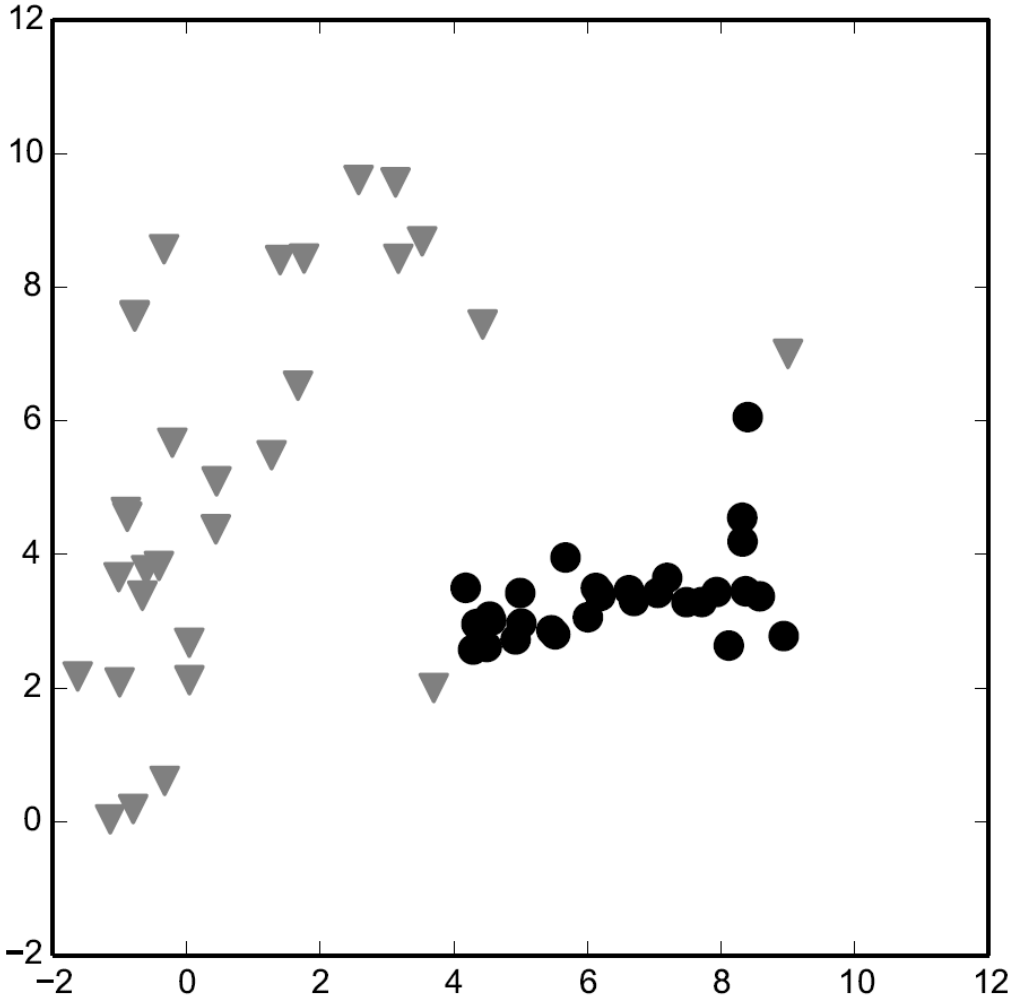


6 Bonus: Quadratic SVM (35 pts)

(a) (5 pts) Draw the decision boundary on the Figure for a very large value of C

(b) (5 pts) Draw the decision boundary on the Figure for a very small value of C

(c) (5 pts) Which of the two (large C or small C) do you expect to generalize better and why?

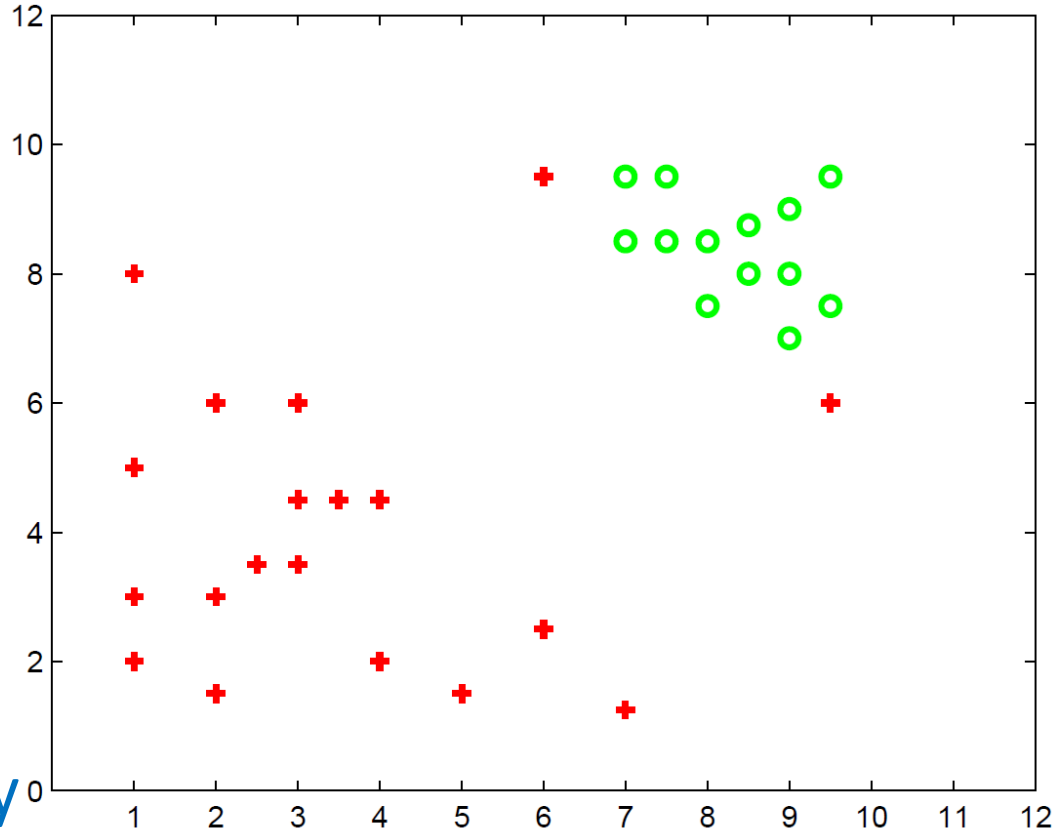


6 Bonus: Quadratic SVM (35 pts)

(d) (20 points) You are given the data set presented in next Figure

- You have been warned, however, that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific example too much

- You are interested in training an SVM with a quadratic kernel
  - The slack penalty C will determine the location of the separating hyperplane



i. Where would the decision boundary be for very large values of C (i.e., C → ∞)?

- Draw on the figure and justify your answer

6 Bonus: Quadratic SVM (35 pts)

(d)

ii. For C ≈ 0, indicate in the Figure, where you would expect the decision boundary to be?

iii. Which of the two previous cases would you expect to work better in the classification task? Why?

iv. Draw an example which will not change the decision boundary learned for very large values of C

v. Draw an example which will significantly change the decision boundary learned for very large values of C