edX

Course > Week 7... > Proble... > Proble...

Problem Set 7

Problems 1-4 correspond to "Kernels I: basis expansion"

Problem 1

1/1 point (graded)

Which of the following are quadratic functions of $x = (x_1, x_2, x_3)$? Select all that apply.

☐ $x_1^2 + x_2^2 + x_3^2 + x_1 x_2 x_3$

☑ $x_1 x_2 + x_1 x_3 + x_2 x_3$

☐ $(x_1 + x_2 + x_1 x_2)^2$

☑ $1 + x_1 + x_2 + x_3 + 10 x_2^2$

✔

Submit

Problem 2

1/1 point (graded)

A decision boundary in $\mathbb{R}^2$ is given by the equation

$$x_1 + 3x_1 x_2 = 6x_2^2 + 8.$$

This can be written in form $w \cdot \Phi(x) + b = 0$, where:

Processing math: 30%

$x = (x_1, x_2)$

$\Phi(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$ is a basis expansion

$b = -8$

What is w?

○ $w = (1, 3, -6, 0, 8)$

○ $w = (1, 0, 0, 3, 0)$

◉ $w = (1, 0, 0, -6, 3)$

○ $w = (1, -6, 0, 0, 3)$

✔

Submit

Problem 3

2/2 points (graded)
Data vectors $x = (x_1, ..., x_4)$ are augmented to give expanded features

$$\Phi(x) = (x_1, ..., x_4, x_1^2, ..., x_4^2, x_1 x_2, ..., x_3 x_4).$$

a) What is the dimension of $\Phi(x)$?

14   ✔

b) The Perceptron algorithm is run using the basis expansion $\Phi(x)$ and returns w, b. What is the dimension of w?

14   ✔

Processing math: 30%

Submit

Problem 4

1/1 point (graded)

We have a data set, of d-dimensional points and their labels, that is linearly separable. However, we use a basis expansion

$$\Phi(x) = (x_1, \ldots, x_d, x_1^2, \ldots, x_d^2, x_1 x_2, \ldots, x_{d-1} x_d),$$

and run the Perceptron algorithm with these expanded features. Which of the following outcomes necessarily occur? Select all that apply.

- ☑ The algorithm converges.

- ☐ The algorithm does not converge.

- ☐ The algorithm returns a vector w in which the entries corresponding to quadratic terms in $\Phi(x)$ (such as x_1^2) are zero.

✔

Submit

Problems 5-7 correspond to "Kernels II: the kernel trick"

Problem 5

1/1 point (graded)

For d-dimensional vectors x, let $\Phi(x)$ denote the basis expansion used to produce quadratic decision boundaries. If vectors $x, z \in \mathbb{R}^d$ are orthogonal to each other, what is $\Phi(x) \cdot \Phi(z)$?

1   ✔

Processing math: 30%

Submit

Problem 6

3/3 points (graded)

For d-dimensional vectors x, let $\Phi(x)$ denote the basis expansion used to produce quadratic decision boundaries. In what follows, suppose we have a data set of n labeled points.

a) What is the dimension of $\Phi(x)$? Pick the best option below.

- $O(d)$

- ● $O(d^2)$

- $O(n)$

✔

b) Suppose we learn a quadratic decision boundary by using the Perceptron algorithm with the kernel trick. The algorithm performs k updates and then converges, returning a vector α. What is the dimension of α? Pick the best option below.

- $O(d)$

- $O(d^2)$

- ● $O(n)$

✔

c) Continuing from part (b), what is the time complexity of classifying a new data point using α? Pick the best option below.

- $O(k)$

- ● $O(kd)$

Processing math: 30%

◯ O(kd^2)

◯ O(n)

✔

Submit

Problem 7

1/1 point (graded)

A data set consists of just four points in $\mathbb{R}$^2:

Label 1: points x^{(1)} = (2,3) and x^{(2)} = (3,6)

Label -1: points x^{(3)} = (1,2) and x^{(4)} = (0,7)

The kernel Perceptron algorithm (in dual form) is run on this data set, with a basis expansion \Phi(\cdot) that produces a quadratic boundary. The algorithm converges after just five update steps: two updates on point x^{(2)}, two updates on point x^{(3)} and one update on point x^{(4)}.

What vector \alpha and value of b are returned?

◉ \alpha = (0,2,2,1), \ b = -1

◯ \alpha = (0,1,1,1), \ b = 2

◯ \alpha = (0,2,2,1), \ b = 1

◯ \alpha = (0,1,1,1), \ b = 0

✔

Submit

Problems 8-11 correspond to "Kernels III: kernel SVM"

Processing math: 30%

Problem 8

1/1 point (graded)
To use the kernel trick with support vector machines, which version of the SVM optimization problem do we use?

◯ The primal form, which returns w and b

◉ The dual form, which returns \alpha and b

✔

Submit

Problem 9

1/1 point (graded)
In order to solve the kernel SVM optimization problem, what information about the data set \{(x^{(i)}, y^{(i)})\} do we need to provide to the optimization procedure?

◯ The labels y^{(i)} and the basis expansions \Phi(x^{(i)})

◯ The labels y^{(i)} and the squared norms \Phi(x^{(i)}) \cdot \Phi(x^{(i)})

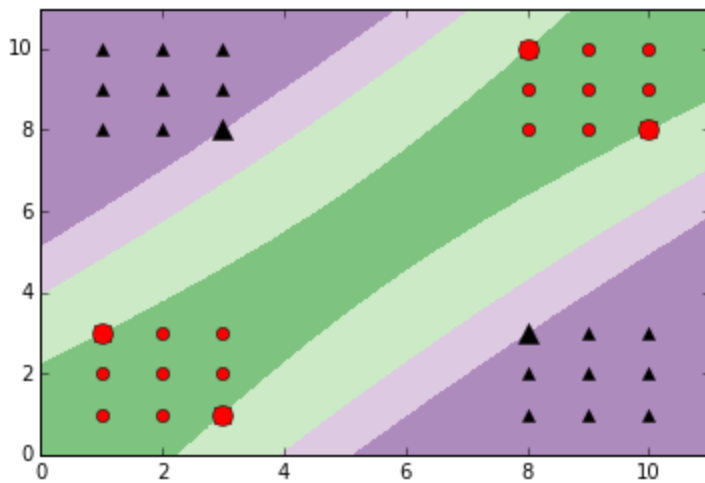◉ The labels y^{(i)} and the pairwise dot products \Phi(x^{(i)}) \cdot \Phi(x^{(j)})

✔

Submit

Problem 10

3/3 points (graded)
Consider the following example from lecture, which shows the decision boundary resulting from applying kernel SVM (with quadratic kernel) to a small two-dimensional data set.

Processing math: 30%

a) What is the dimension of \alpha in this case?

36 ✔

b) How many entries in \alpha are \gt 0?

6 ✔

c) How many entries in \alpha are \lt 0?

0 ✔

d) For you to think about: why does the margin on the green side appear to be larger than the margin on the purple side?

Submit

Problem 11

1/1 point (graded)
We have a data set of n labeled points and we wish to use kernel SVM to fit a quadratic boundary to it. To do this, we need to supply the labels of the points as well as the n

Processing math: 30%   pairwise dot products between the expanded feature vectors. Call this matrix K.

Now suppose that we change our mind and instead want a boundary that is a polynomial of degree 4. Again, we supply the labels of the data points and a matrix L. How does L compare to K? Select all that apply.

☐ K is n \times n while L is n^2 \times n^2

☑ L is also n \times n.

☐ L = K^2

☑ Each entry of L is the square of the corresponding entry of K

✔

Submit

Problems 12-15 correspond to "Kernels IV: the kernel function"

Problem 12

1/1 point (graded)

In order to get a nonlinear decision boundary, we can use a basis expansion \Phi(x). The corresponding kernel function is then k(x,z) = \Phi(x) \cdot \Phi(z). Do we need knowledge of both k(\cdot) and \Phi(\cdot) to learn a classifier and subsequently use it, or is it enough to know just k(\cdot)?

○ In order to learn a classifier, we need to know \Phi(\cdot) in addition to k(\cdot). In order to use the classifier, it is sufficient to just know k(\cdot).

○ In order to learn a classifier, it is sufficient to just know k(\cdot). In order to use the classifier, we need to know \Phi(\cdot) in addition to k(\cdot).

◉ It is sufficient to just know k(\cdot) for both learning a classifier and subsequently using it.

✔

Processing math: 30%

Submit

Problem 13

1/1 point (graded)

When using the RBF kernel, the final classifier assigns a label to a new point x by taking a weighted vote over the labels of the training points $x^{(i)}$. What do these weights depend upon? Select all that apply.

- ☑ The \alpha_i found by the learning algorithm: larger \alpha_i means a larger weight for $x^{(i)}$.

- ☑ The distance between x and $x^{(i)}$: closer points $x^{(i)}$ get a larger weight.

- ☐ The distance between x and $x^{(i)}$: farther points $x^{(i)}$ get a larger weight.

- ☐ The dot product x \cdot $x^{(i)}$: larger dot product implies a larger weight for $x^{(i)}$

- ☐ The dot product x \cdot $x^{(i)}$: larger dot product implies a smaller weight for $x^{(i)}$

✔

Submit

Problem 14

1/1 point (graded)

We decide to define k(x,z) = x \cdot z. What can be said about this?

- ○ This is not a valid kernel function.

- ○ This is a valid kernel function and yields a quadratic decision boundary.

- ● This is a valid kernel function and yields a linear decision boundary.

Processing math: 30%

✔

Submit

Problem 15

1/1 point (graded)
We decide to define k(x,z) to be 1 if x=z and 0 otherwise. What can be said about this?
Select all that apply.

[ ] This is not a valid kernel function.

[✔] This is a valid kernel function.

[✔] Assuming the training points are distinct, we can achieve zero training error using a classifier based on k(\cdot).

[✔] A classifier based on k(\cdot) is unlikely to generalize well beyond the training set.

✔

Submit

Problems 16-19 correspond to "Decision Trees"

Problem 16

1/1 point (graded)
Decision trees have great expressive power: they can represent any decision boundary, by carving up the space to a fine enough granularity. Which of the other methods we've studied are similarly expressive? Select all that apply.

[ ] Linear classifiers.

[ ] Support vector machines with a quadratic kernel.

Processing math: 30%

> ✔️ Nearest neighbor classifiers.

> ☐ Classifiers based on Gaussian generative models.

✔

Submit

Problem 17

1/1 point (graded)

Let's say our training set consists of n data points in d-dimensional space. At the top node of the tree, we have to pick a split, which involves choosing (i) a feature and (ii) a split value along that feature. How many possibilities do we need to try out, roughly?

> ⚪ n

> ⚪ d

> ⚪ Infinitely many

> ⦿ nd

✔

Submit

Problem 18

1/1 point (graded)

In order to grow a decision tree, we need a measure of how pure or impure (in terms of labels) a node is. A popular measure of this is the "Gini impurity index". If there are two labels, and a node has p fraction of one label and 1-p fraction of the other, the Gini impurity index is 2p(1-p). Notice that this is maximized when p=1/2.

What is the Gini impurity index for a node in which 20\% of the points have one label while 80\% have the other label?

Processing math: 30%

⊙ 0.2

⊙ 0.16

⊙ **0.32**

⊙ 0.8

✔

Submit

Problem 19

1/1 point (graded)

We have talked about a measure of impurity in the case of binary classification. If there are k possible classes, and a node contains these in proportions $p_1, p_2, \ldots, p_k$ (summing to 1), which of the following is a suitable measure of impurity? That is, the higher the value, the more impure the node. Select all that apply.

☐ The maximum of $p_1, p_2, \ldots, p_k$

☐ $p_1^2 + p_2^2 + \cdots + p_k^2$

☐ The minimum of $p_1, p_2, \ldots, p_k$

☑ $1 - (p_1^2 + p_2^2 \cdots + p_k^2)$

✔

Submit

Problems 20-21 correspond to "Boosting"

Processing math: 30%

1/1 point (graded)

Boosting requires the weak learning algorithm to work with training data in which each point $(x^{(i)}, y^{(i)})$ has a positive weight $w_i \gt 0$. Intuitively, a weight of two would be equivalent to having two copies of that data point.

Here's something for you to think about: how would you incorporate weights into the following learning algorithms without explicitly making copies of data points?

\bullet Decision trees.

\bullet Gaussian generative models.

\bullet Support vector machines.

Now select one of the following (you will get marked correct either way).

- ◉ I have figured this out.

- ○ I am having some trouble with this.

✔

Submit

Problem 21

1/1 point (graded)

Suppose we run boosting using weak classifiers from some class H (such as decision stumps), and suppose that on each round, the weak learner always returns a weak classifier whose error rate on the weighted data (for that round) is at most $1/2 - \epsilon$, for some $\epsilon \gt 0$. Which of the following is true? Select all that apply.

- ☐ Boosting converges to a classifier with zero test error.

- ☑ Boosting converges to a classifier with zero training error.

- ☐ Boosting converges to the classifier that has the smallest possible test error out of any classifier in H.

Processing math: 30%

Submit

Problems 22-23 correspond to "Random forests"

Problem 22

1/1 point (graded)

Which of the following is an obvious benefit of random forests over boosted decision trees? Select all that apply.

- ☑ The trees can be trained in parallel.

- ☐ Each individual tree is more highly optimized.

- ☐ Each individual tree has better accuracy.

✔

Submit

Problem 23

1/1 point (graded)

When learning a tree in a random forest, the split at each node is chosen from only a subset of the full feature set. What is the main reason for this?

- ○ It speeds up the training process.

- ◉ It creates trees with greater variability.

- ○ It improves the accuracy of each tree.

- ○ It ignores irrelevant features.

✔

Processing math: 30%

Submit

Processing math: 30%