CPS331 Lecture: Resemblance-Based Learning last revised November 8, 2018

Objectives:

1. To introduce support vector machines
2. To introduce the notion of linear separability
3. To introduce the "kernel trick"

Materials:

1. Story from p. 177 of Domingos to read
2. Projectable of Komondor dog
3. Projectable of example data points
4. Projectable of the above with multiple separators
5. Projectable of the above showing notion of margin

I. Introduction

A. Pedro Domingos begins his chapter on resemblance-based learning with the following story

READ Story from p. 177

B. The story I just read suggested a general problem-solving strategy: from the set of descriptions of problems solved in the past, find the one that most-closely resembles the problem being solved, and use or adapt the solution for that piece of data.

C. This is a strategy that we use in everyday life. Consider the following picture:

PROJECT picture of Komondor

1. What kind of animal is this?

ASK

It's a Komondor - an unusual-looking dog.

2. I assume none of you knows how to recognize a Komondor. How did you recognize it as a dog?

   ASK

3. Presumably you recognized this as a dog because it is more similar to other dogs you have seen that it is to other creatures like cats or squirrels or ...

D. Resemblance-based reasoning is used in lots of places.

Examples:

1. Law
2. Medical diagnosis
3. Auto mechanics fixing a car

E. As a learning strategy, we use a set of training data representing previously-solved / classified problems, and learn an efficient way of identifying the one(s) that serve as precedents.

II. Support-Vector Machines

A. One class of approaches that is widely used for supervised learnng is called support-vector machines (or SVM's). In fact, there are "off-the-shelf" SVM packages that can be applied to a given data set without the user having to understand the underlying mathematics.

B. In its basic form, an SVM is used for two-way classification i.e. a given instance belongs to one or the other of two classes. Often, this takes the form of a decision problem - i.e. for any given description, decide whether or not it belongs to a particular category (a yes-no, boolean result). An SVM finds a function that has the following behavior:

Where the description of the instance is in the form of what is called a feature vector .

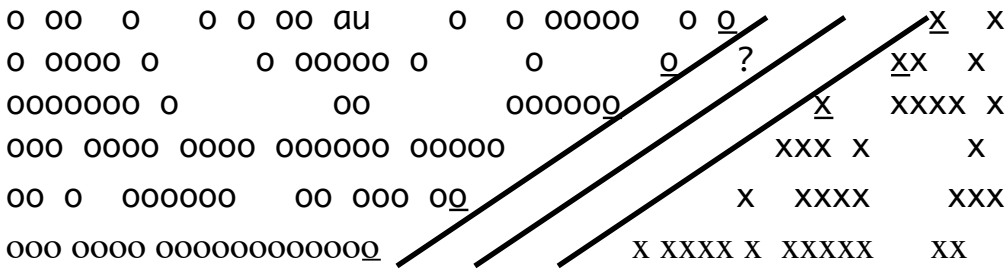
C. SVM's learn what is called a "maximum margin" mapping. To see what that is, consider the following: Suppose our training data consists of entities described by a feature vector with two values. Suppose, if we were to plot all the data on a single graph, we would get the following, where the x's represent data points for one classification and the o's data points for the other classification.

```
o  oo   o     o  o  oo  au      o   o  ooooo   oo  o            x   x
o  oooo  o        o  ooooo  o       o         o              xx   x
ooooooo  o           oo          oooooo             x     xxxx  x
ooo  oooo  oooo  oooooo  ooooo                   xxx  x       x
oo  o   oooooo     oo  ooo  oo                    x   xxxx      xxx
ooo oooo oooooooooooo                    x xxxx x  xxxxx     xx
```

PROJECT

1. This data is linearly-separable - i.e. it is possible to draw a single line that separates the two categories. (If the feature vector has more than two elements - as is generally the case - we would have a multi-dimension graph and the separator would be a hyper-plane.)

2. In fact, though, in a case like this there are multiple possibilities for the separator:



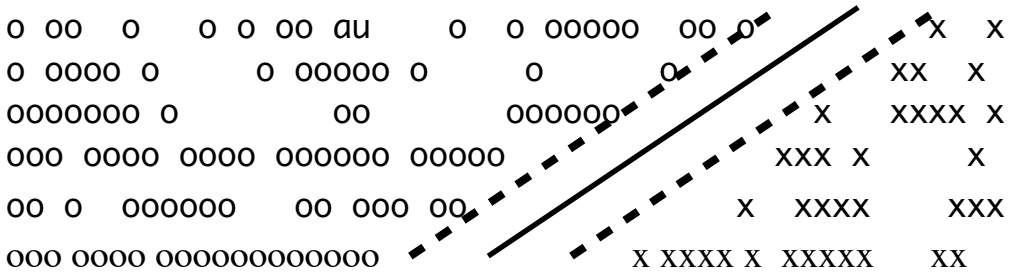
PROJECT

3. We call the descriptions of the data points that are nearest the separator line (the underlined ones) the support vectors. The mapping ultimately discovered will depend only on these - they support the mapping.

4. Which separator is the most desirable?

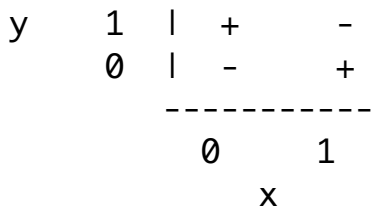
   a) In two cases, the separator is so close to some of the data points that a new data point that belongs to one of the categories might be misclassified (e.g. the point labeled ?, which probably should be classified as an o but will be classified instead as an x if the leftmost separator is used.

   b) The most desirable separator is the middle one. We call this the maximum-margin separator, because it has the largest margins between it and the actual data points - shown below as the spaces between the separator and the dashed lines.



PROJECT

5. An SVM can also be learned with data that is not linearly separable. Consider the following two-dimensional data:



Where the +'s are at (0, 1) and (1, 0), and the -'s are at (0, 0) and (1,1)

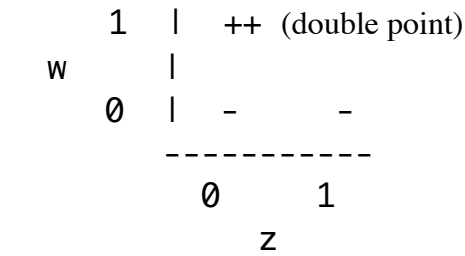
(Hopefully, you recognize that this is essentially the exclusive or function: something is classified as a "+" if x xor y is 1.)

6. Clearly, there is no way to draw a single line that separates the +'s and -'s. (This corresponds to the xor-problem we looked at with neural networks)

7. Suppose we replace the original variables with two new variables calculated from the original ones:

$w = (x-y)^2$
$z = xy$.

Then we could graph the points in terms of these new variables as follows



Clearly, this function is linearly separable.

8. Such a way of replacing variables (and perhaps adding additional ones) is called the kernel trick.

   a) It can be done ahead of time as a preprocessing step if it is known that the function is not linearly separable; there are several well-known transformations that can be used in cases like this.

   b) It can be used by SVM's to convert non-linearly separable ones into the equivalent of linearly separable ones.