

DSO530 Statistical Learning Methods

Lecture 8 : Support Vector Machines (SVMs)

Dr. Xin Tong

Department of Data Sciences and Operations

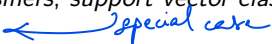
Marshall School of Business

University of Southern California

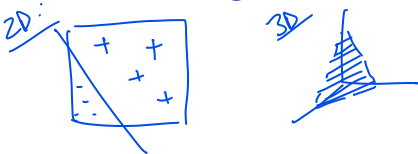
Introduction

special
case



- Topics include *maximal margin classifiers*, *support vector classifiers* and *support vector machine (SVM)* 
- SVMs were invented in the 1990s; it had some dominance until the resurgence of neural networks.
- SVMs are still considered one of the best off-the-shelf classifiers
- People often loosely refer to the maximal margin classifier, the support vector classifier, and the support vector machine as “support vector machines”
- In SVM, people label the two classes by +1 and -1
- This lecture covers the methodology. For implementation, use `svm` in `sklearn`: <https://scikit-learn.org/stable/modules/svm.html>

Maximal margin classifier



A generic equation in 2D
 $a + bx_1 + cx_2 = 0$

- **hyperplane**: In a p -dimensional space, a hyperplane is a flat *affine* subspace of dimension $p - 1$
- *Affine* subspace is a fancy math term. It is like a linear subspace, but the origin is not necessarily in it
- Mathematically, a hyperplane is

$$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p = 0$$

- If $X = (X_1, \dots, X_p)^T$ leads to $>$, it lies on one side of the hyperplane; if $<$, it lies on the other side

-ve on one side
of hyperplane
& +ve on the other
otherwise
hyperplane
would be useless

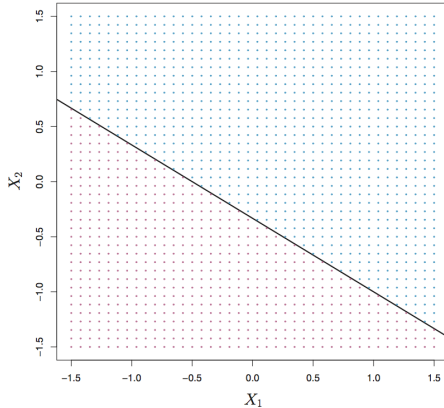
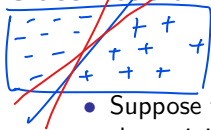


FIGURE 9.1. The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.

Classification Using a Separating Hyperplane



multiple ideal hyperplanes exist but blue generalizes well.

- Suppose that it is possible to construct a hyperplane that separates the training observations perfectly according to their class labels
- A separating hyperplane has the property that

$$y_i(\underbrace{\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}}_{\text{+ve or -ve}}) > 0,$$

Therefore +ve x +ve
&
-ve x -ve
is
+ve

for all $i = 1, \dots, n$

- A test observation is assigned a class depending on which side of the hyperplane it is located
- That is, we classify the test observation x^* based on the sign of

$$f(x^*) = \beta_0 + \beta_1 x_1^* + \dots + \beta_p x_p^*$$

- If $f(x^*) > 0$, we assign to class 1, and if $f(x^*) < 0$, we assign to class -1
- In addition to the sign, how should you interpret $|f(x^*)|$?

the bigger the abs value, farther thept from hyperplane

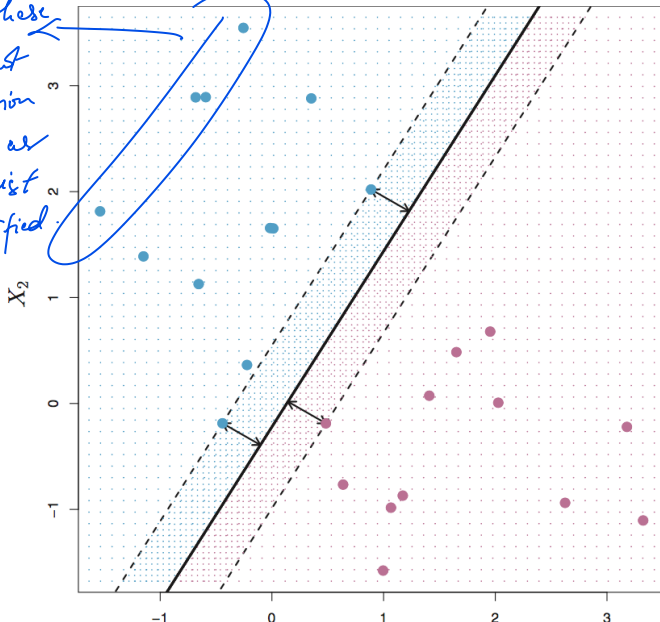
The Maximal Margin Classifier

↓
minimal
distance.

- If data can be perfectly separated by a hyperplane, then there exist an infinite number of such hyperplanes
- A natural question: which one do we want to choose?
- A natural choice is the **maximal margin hyperplane** (also known as the *optimal separating hyperplane*), which is the separating hyperplane that is farthest from the training observations.
- The minimal distance from the training observations to the hyperplane is known as the **margin**
- **Maximal margin classifier**: classify a test observation based on which side of the maximal margin hyperplane it lies

- There are three **support vectors** in the plot
- If you move other training data, what happens to the maximal margin classifier?

moving these
pts does not
move decision
boundary as
minimal dist
is still satisfied



Construction of the Maximal Margin Classifier

$$\underbrace{\text{maximize } M}_{\beta_0, \beta_1, \dots, \beta_p, M} \quad (9.9)$$

$$\text{subject to } \sum_{j=1}^p \beta_j^2 = 1, \quad (9.10)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M \quad \forall i = 1, \dots, n. \quad (9.11)$$

- Equation(9.11) guarantees that each observation will be on the correct side of the hyperplane, provided that M is positive
- Under equation (9.10), the perpendicular distance from the i th observation to the hyperplane is given by

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip})$$

- M represents the margin of the hyperplane

for every training
obs dist has
be at least M
↑ from decision
boundary.

Non-separable Case

special case of
of maximal margin.

- What if no separating hyperplane exists? A common scenario
- In other words, optimization problem (9.9-9.11) has no solution for $M > 0$
- A fix: develop a hyperplane that *almost* separates the classes, using a so-called soft margin \rightarrow for non separable class.
- **Support vector classifier** (a.k.a. soft margin classifier): the generalization of the maximal margin classifier to the non-separable case
- The margin is *soft* because it can be violated by some of the training observations

- Even if a separating hyperplane does exist, there are instances in which a classifier based on a separating hyperplane might not be desirable

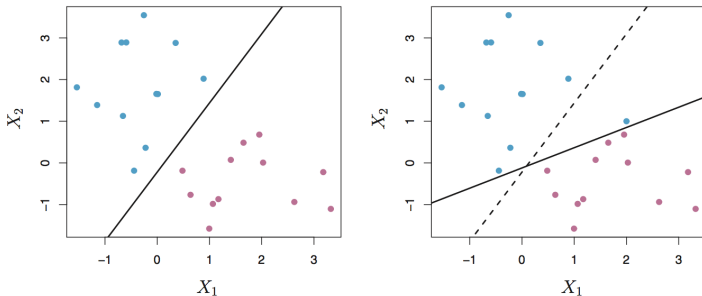


FIGURE 9.5. Left: Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane. Right: An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.

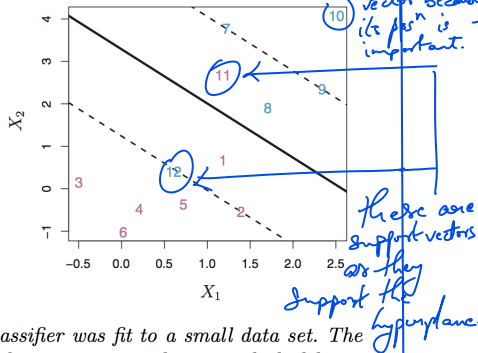
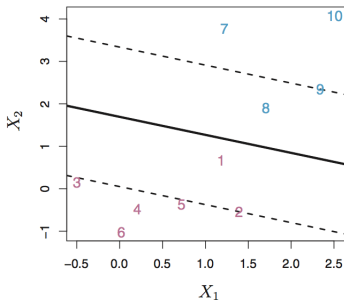


FIGURE 9.6. Left: A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines. Purple observations: Observations 3, 4, 5, and 6 are on the correct side of the margin, observation 2 is on the margin, and observation 1 is on the wrong side of the margin. Blue observations: Observations 7 and 10 are on the correct side of the margin, observation 9 is on the margin, and observation 8 is on the wrong side of the margin. No observations are on the wrong side of the hyperplane. Right: Same as left panel with two additional points, 11 and 12. These two observations are on the wrong side of the hyperplane and the wrong side of the margin.

w/o soft margins, maximal margin classifier is not robust enough.

Optimization Program for Support Vector Classifier

If $0 < \epsilon_i < 1$,
pts allowed on
wrong side of margin.

$$\text{maximize}_{\beta_0, \beta_1, \dots, \beta_p, \epsilon_1, \dots, \epsilon_n, M} M \quad (9.12)$$

smaller than n

$$\text{subject to } \sum_{j=1}^p \beta_j^2 = 1, \quad (9.13)$$

If $\epsilon_i > 1$,
pts are allowed
on wrong side of
hyperplane.

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \quad (9.14)$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^n \epsilon_i \leq C, \quad (9.15)$$

difference
from nominal
margin classifier.

- C is a nonnegative tuning parameter; We can think of C as a budget for the amount that the margin can be violated by the n observations
- **Q:** as C increases, will the margin widen? How about the bias-variance trade-off?
 ↳ bias ↳ variance as small C changes model
- $\epsilon_1, \dots, \epsilon_n$ are slack variables that allow individual observations to be on the wrong side of the margin or the hyperplane
- Once we solve (9.12 – 9.15), we classify a test point x^* as before
- **Q:** what does it mean for some ϵ_i to be positive or bigger than 1?

Classification with Non-linear Decision Boundaries

- Sometimes no linear decision boundary does well
- How to adapt the support vector classifiers to accommodate these cases? *→ from linear regression, x_1^2, x_1^3*
- One solution is to introduce interaction terms, higher-order terms as new features, as we did for linear regression
- But this approach is often ad-hoc and incurs heavy computation burden if we introduce too many new features
- **support vector machine (SVM)**: an extension of the support vector classifier that results from enlarging the feature space using **kernels**
- The detail is beyond the scope of the course, but we can take a look at some common kernels
- Interested readers can refer to Prof. Andrew Ng's CS 229 notes
- **Kernel**: function that quantifies the similarity of two observations

linear kernel: $K(x_i, x_{i'}) = \sum_{j=1}^p x_{ij} x_{i'j}$

non linear decision boundaries

- Other kernels

$$\text{polynomial kernel: } K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^p x_{ij} x_{i'j}\right)^d$$

$$\text{radial kernel: } K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2\right)$$

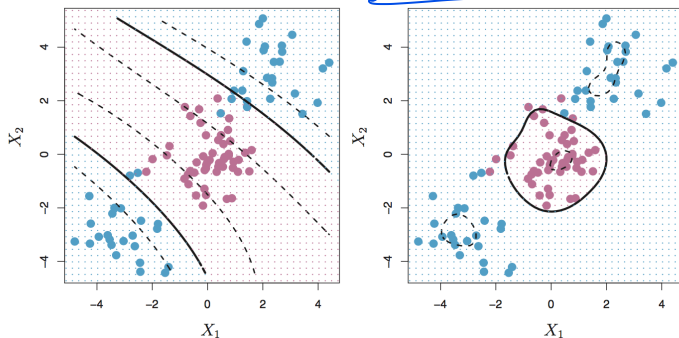


FIGURE 9.9. Left: An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule. Right: An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.