ECON 950 — Winter 2020

Prof. James MacKinnon

12. Support Vector Machines

These notes are based on Chapter 9 of ISLR.

Support vector machines are a popular method for classification problems where there are two classes.

There are extensions for regression and multi-way classification, but we will not discuss them.

12.1. Separating Hyperplanes

Recall that a **hyperplane** in two dimensions is defined by

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0. \tag{1}$$

This is just a straight line.

More generally, when there are p dimensions, we can write

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 \ldots + \beta_p X_p = 0. \tag{2}$$

If we form the X_i into a vector $\boldsymbol{x}$, we can also write

$$\beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta} = 0. \tag{3}$$

Every hyperplane divides the space in which it lives into two parts, depending on whether $\beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta} > 0$ or $\beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta} \leq 0$.

In some cases, when we have data labelled with two classes, we can find a **separating hyperplane** such that all the points in one class lie on one side of it, and all the points in the other class lie on the other side.

Let the training observations be denoted y_i and $\boldsymbol{x}_i$, where y_i contains the class labels, which are -1 and 1.

If a separating hyperplane exists, it must have the property that

$$\begin{aligned} \beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} > 0 \quad & \text{if } y_i = 1 \\ \beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} < 0 \quad & \text{if } y_i = -1 \end{aligned} \tag{4}$$

for all observations. More compactly, we can write

$$y_i(\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta}) > 0 \quad \text{for all } i = 1, \ldots, N. \tag{5}$$

Notice that the values of β_0 and $\boldsymbol{\beta}$ are not unique. If (5) is true for any $(\beta_0, \boldsymbol{\beta})$ pair, then it is also true for $(\lambda\beta_0, \lambda\boldsymbol{\beta})$ for any positive λ.

If one separating hyperplane exists, then typically an infinite number of them exist. See ISLR-fig-9.02.pdf. This is true even if we impose a constraint like $\beta_0^2 + ||\boldsymbol{\beta}||^2 = 1$.

When a separating hyperplane exists, we have a **perfect classifier**. For every observation, we can classify y_i as -1 or 1 with certainty.

With other methods, such as logit and probit, having a perfect classifier is bad. It makes it impossible to obtain parameter estimates that are finite.

But for support vector machines, this is the ideal situation, albeit one that is rarely achieved with actual data.

The loglikelihood function for both logit and probit models can be written as

$$\ell(\boldsymbol{y}, \beta_0, \boldsymbol{\beta}) = \sum_{y_i=1} \log F(\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}) + \sum_{y_i=-1} \log\big(1 - F(\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta})\big). \tag{6}$$

When there exists a separating hyperplane, and we evaluate $F(\cdot)$ in (6) at values that define it, we have $\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta} > 0$ for every observation in the first summation, and $\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta} < 0$ for every observation in the second summation.

This implies that $F(\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}) > 0.5$ for every observation in the first summation, and $F(\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta}) < 0.5$ for every observation in the second summation.

If we multiply β_0 and $\boldsymbol{\beta}$ by a positive number $\lambda > 1$, we increase the value of every term in (6). The value of $F(\beta_0 + \boldsymbol{x}_i\boldsymbol{\beta})$ gets closer to 1 for terms in the first summation, and closer to 0 for terms in the second summation.

The maximum possible value of $\ell(\boldsymbol{y}, \beta_0, \boldsymbol{\beta})$ is 0. We can make it as close as we like to 0 by making λ big enough.

In terms of β_0 and $\boldsymbol{\beta}$, all values are going to plus or minus infinity as this happens. So any optimization algorithm will fail.

For support vector machines, in contrast, having a separating hyperplane, and hence a perfect classifier, is actually the ideal situation.

We simply classify a test observation, say $\boldsymbol{x}^*$, as 1 if $\beta_0 + \boldsymbol{x}^{*\top}\boldsymbol{\beta} > 0$ and as -1 if $\beta_0 + \boldsymbol{x}^{*\top}\boldsymbol{\beta} < 0$.

12.2. Maximal Margin Classifiers

As we saw in ISLR-fig-9.02.pdf, if there exists a separating hyperplane, there are typically an infinite number of them.

The **maximal margin hyperplane**, or **optimal separating hyperplane**, is the one that is farthest from the training observations.

The **margin** is simply the smallest perpendicular distance between any of the training observations $\boldsymbol{x}_i$ and the hyperplane.

The **maximal margin classifier** simply classifies each observation based on which side of the maximal margin hyperplane it is.

This is shown in ISLR-fig-9.03.pdf for the data in ISLR-fig-9.02.pdf.

In the figure, the maximal margin hyperplane depends on just three points, the three **support vectors**. Small changes in the location of other observations does not affect its location.

The maximal margin hyperplane can be obtained by solving a particular optimization problem. We need to maximize M with respect to M, β_0, and $\boldsymbol{\beta}$ subject to the

constraints

$$y_i(\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta}) \geq M, \quad \text{for all } i = 1, \ldots, N. \tag{7}$$

and

$$\beta_0^2 + \boldsymbol{\beta}^\top \boldsymbol{\beta} = 1. \tag{8}$$

The first constraint ensures that every point is on the right side of the maximal margin hyperplane, and indeed that it is distant from it by at least M, the margin. The second constraint is just a normalization.

Even when separating hyperplanes exist, the maximal margin hyperplane may be very sensitive to individual observations.

In ISLR-fig-9.05.pdf, adding just one observation dramatically changes the slope of the hyperplane.

The optimization problem above can be solved efficiently, but it is almost never of interest, because in practice separating hyperplanes almost never exist.

12.3. Support Vector Classifiers

In practice, a separating hyperplane rarely exists. For any possible hyperplane, there will be some observations on the wrong side.

The **support vector classifier** or **soft margin classifier** chooses a hyperplane where some observations are on the wrong side.

In some cases, there may exist a separating hyperplane, but it is better to put some observations on the wrong side of the margin.

Now we maximize M subject to the constraints

$$\beta_0^2 + \boldsymbol{\beta}^\top \boldsymbol{\beta} = 1, \tag{9}$$

$$y_i(\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta}) \geq M(1 - \varepsilon_i), \quad \text{for all } i = 1, \ldots, N, \tag{10}$$

where $\varepsilon_i \geq 0$ and

$$\sum_{i=1}^{N} \varepsilon_i \leq C. \tag{11}$$

We now have to choose the ε_i as well as M, β_0, and $\boldsymbol{\beta}$. The ε_i are called **slack variables**.

Equation (9) is the same as (8). It is just a normalization.

What has changed is that (10) allows points to be on the wrong side of the margin when $\varepsilon_i > 0$.

In (11), C is a nonnegative tuning parameter. Its value, not surprisingly, turns out to be very important.

If $\varepsilon_i = 0$, then observation i lies on the correct side of the margin.

If $\varepsilon_i > 0$, then observation i lies on the wrong side of the margin.

If $\varepsilon_i > 1$, then observation i lies on the wrong side of the hyperplane.

The value of C puts a limit on the extent to which the ε_i can collectively exceed zero. When $C = 0$, we are back to (7) and (8).

For $C > 0$, no more than C observations can be on the wrong side of the hyperplane, because we will have $\varepsilon_i > 1$ for every such observation.

Since every violation of the margin increases the sum of the ε_i, we can afford more violations when C is large than when it is small. Thus M will almost surely increase with C.

ISLR-fig-9.07.pdf illustrates what can happen as C changes. In it, the value of C decreases from upper left to lower right.

One important feature of the SV classifier is that only observations that lie on the margin or that violate the margin will affect the hyperplane.

For all other observations, the inequalities in (10) are satisfied with $\varepsilon_i = 0$. Moving them a little (or a lot) while keeping them on the correct side of the margin has no effect at all on the solution.

The observations that matter (the ones on the margin or on the wrong side of it) are called **support vectors**.

When the tuning parameter C is large, the margin is wide, many observations violate the margin, and so there are many support vectors. There will tend to be low variance but high bias.

When the tuning parameter C is small, the margin is narrow, few observations violate the margin, and so there are few support vectors. There will tend to be low bias but high variance.

The SV classifier is totally insensitive to observations on the correct side of the margin, and therefore (for a wide margin) on the correct side of the hyperplane by quite a bit.

For logistic regression, something similar but less extreme is true. The estimates are never totally insensitive to any observation, but they are not very sensitive to observations that are far from the hyperplane on the correct side.

12.4. Support Vector Machines

So far, we have only considered decision boundaries that are hyperplanes. But if the boundaries are actually nonlinear, hyperplanes won't work well.

See ISLR-fig-9.08.pdf.

We could just add powers and/or cross-products of the x_{ij}, increasing the number of parameters to be estimated.

The **support vector machine**, or **SVM**, is an extension of the support vector classifier that results from enlarging the feature space using kernels.

The solution to the support vector classifier problem in (9) and (10) involves only the inner products of the observations.

The linear support vector classifier for any point $\boldsymbol{x}$ can be represented as

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{N} \alpha_i \boldsymbol{x}^\top \boldsymbol{x}_i, \tag{12}$$

where there is one parameter α_i for each training observation.

To estimate the parameters β_0 and α_i, we need every inner product $\boldsymbol{x}_i^\top \boldsymbol{x}_{i'}$. There are $N(N-1)/2$ of these.

It turns out that $\alpha_i = 0$ if $\boldsymbol{x}_i$ is not a support vector.

Thus we can rewrite (12) as

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \boldsymbol{x}^\top \boldsymbol{x}_i, \tag{13}$$

where $\mathcal{S}$ is the set of support vectors. This makes it very inexpensive to classify a new observation.

Now suppose that, whenever the inner product $\boldsymbol{x}_i^\top \boldsymbol{x}_{i'}$ appears in calculations for the support vector classifier, we replace it by the **kernel**

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}), \tag{14}$$

where $K(\cdot)$ can be chosen in various ways. Now

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}). \tag{15}$$

The **linear kernel** is just $\boldsymbol{x}_i^\top \boldsymbol{x}_{i'}$, which gives us the support vector classifier.

The **polynomial kernel** of degree d is

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = (1 + \boldsymbol{x}_i^\top \boldsymbol{x}_{i'})^d. \tag{16}$$

In effect, we are now fitting a support vector classifier in a higher-dimensional space involving polynomials of degree d, rather than in the original feature space.

Another possibility is the **radial kernel**, which is

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \exp\big(-\gamma(\boldsymbol{x}_i - \boldsymbol{x}_{i'})'(\boldsymbol{x}_i - \boldsymbol{x}_{i'})\big). \tag{17}$$

The value of γ, which is a positive constant, is chosen in advance or by cross-validation.

The radial kernel has very local behavior. When a test observation $\boldsymbol{x}^*$ is far from the training observation $\boldsymbol{x}_i$, then

$$K(\boldsymbol{x}^*, \boldsymbol{x}_i) = \exp\big(-\gamma(\boldsymbol{x}^* - \boldsymbol{x}_i)'(\boldsymbol{x}^* - \boldsymbol{x}_i)\big) \tag{18}$$

will be very small unless γ is tiny. This means that $\boldsymbol{x}_i$ will play virtually no role in $f(\boldsymbol{x}^*)$. Thus training observations which are far from $\boldsymbol{x}^*$ will have very little impact on the predicted class label for $\boldsymbol{x}^*$.

In this sense, an SVM with a radial kernel is like kernel regression, where observations far from $\boldsymbol{x}^*$ have little or no effect on the fitted value for $\boldsymbol{x}^*$.

One advantage of using kernels instead of simply adding functions of the original features (regressors) is that the data continue to affect the results only through the $N(N-1)/2$ distinct values of $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$.

For any vector $\boldsymbol{x}$, we can compute $\hat{f}(\boldsymbol{x})$ and classify an observation based on the value of $\hat{f}(\boldsymbol{x}) - t$ for some cutoff value t.

Doing this for the heart disease data yields the ten ROC curves shown in four panels in ISLR-fig-9.10-11.pdf.

For the training data, the winner seems to be SVM with $\gamma = 0.1$. It beats LDA (linear discriminant analysis), the support vector classifier, and SVM with smaller values of γ.

However, for the test data, SVM with $\gamma = 0.1$ now seems to be the worst method. It is hard to tell what the best method is.

As ISLR discuss in Section 9.4, SVMs can be generalized to handle more than two classes, but the generalizations are not very natural, because we have to do everything for pairs of outcomes.

There is also something called **support vector regression**, where only residuals larger in absolute value than some positive constant contribute to the loss function.

12.5. SVMs and Logistic Regression

Despite good marketing, and a very different approach to computation, SVMs are not as different from other methods as they were originally claimed to be.

We can rewrite the criterion for estimating a support vector classifier — see (9) through (11) — as

$$\min_{\beta_0, \boldsymbol{\beta}} \left(\sum_{i=1}^{N} \max\bigl(0, y_i f(\boldsymbol{x}_i)\bigr) + \lambda \boldsymbol{\beta}' \boldsymbol{\beta} \right), \tag{19}$$

where $\lambda \geq 0$ is a tuning parameter, and $f(\boldsymbol{x})$ is the support vector classifier.

When λ is large, the β_j will tend to be small, many violations of the margin will be tolerated, and we obtain an estimator with low variance but high bias.

When λ is small, the β_j will tend to be large, fewer violations of the margin will be tolerated, and we obtain an estimator with high variance but low bias.

Thus, a small value of λ corresponds to a small value of C in (11).

Note also that $\lambda\boldsymbol{\beta}'\boldsymbol{\beta}$ looks just like the penalty term for ridge regression.

Thus, the form of the objective function for SVM is very similar to the one for ridge-regularized logistic regression.

The first term inside the summation in (19) is $\max\bigl(0, y_i f(\boldsymbol{x}_i)\bigr)$, which leads to what is called **hinge loss**.

Hinge loss looks like the loss for logistic regression; see ISLR-fig-9.12.pdf.

When an observation is on the correct side of the margin, the loss is zero. When it is on the wrong side, the loss is linear with a slope of 1.

For logistic regression, the loss is small when an observation is on the the correct side of the hyperplane by some distance. When it is on the wrong side of the hyperplane, the loss is approximately linear.

Looking at SVMs in terms of (19) (perhaps generalized to allow for a kernel), we see that the value of λ, and hence C, is really important.

Relative merits of SVM and logistic regression?