Support Vector Machines

Mirko Birbaumer

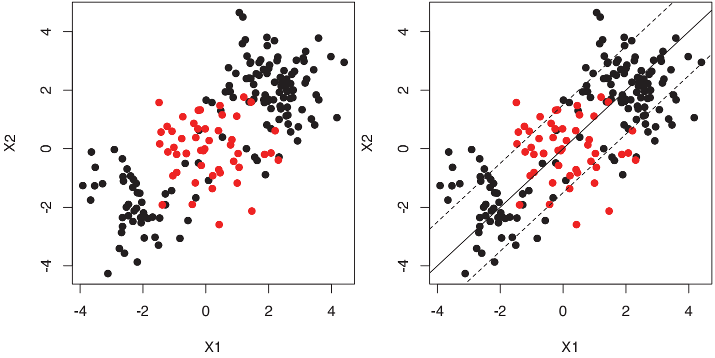
HSLU T&A

Predictive Modeling

Support Vector Machines

- Support vector classifier is a natural approach for classification in the two-class setting, if the boundary between the two classes is **linear**

- In practice, we are faced with **non-linear class boundaries**



Support Vector Machines

- **Support vector machine** (SVM) is an extension of the support vector classifier that results from enlarging the feature space

- **Main idea**: we may want to **enlarge our feature space** in order to accommodate a **non-linear boundary** between the classes

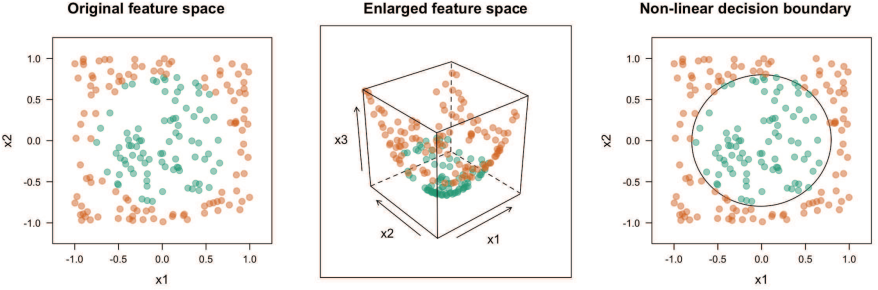
- Rather than fitting a support vector classifier using p features

$$X_1, X_2, \ldots, X_p$$

- We could instead fit a support vector classifier using $2p$ features

$$X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2$$

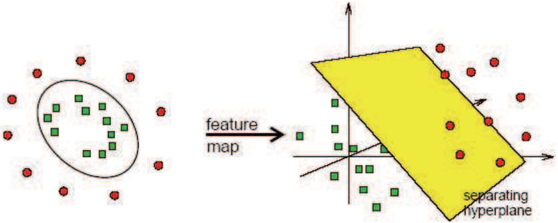
Enlarged Feature Space



We may enlarge the feature space by adding a third feature, say $X_3 = X_1^2 + X_2^2$

Enlarged Feature Space

- Why does enlarging the feature space lead to a **non-linear decision boundary**?
- In the **enlarged feature space**, the resulting **decision boundary** is in fact **linear** (seperating hyperplane)



- In the original feature space, the decision boundary is of the form $q(x) = 0$, where q is a quadratic polynomial, and its solutions are generally **non-linear**

Optimization Problem

1. Maximize M

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n} M \tag{1}$$

2. subject to

$$y_i\left(\beta_0+\sum_{j=1}^{p}\beta_{j1}x_{j1}+\sum_{j=1}^{p}\beta_{j2}x_{j2}^2\right)\geq M\left(1-\varepsilon_i\right) \tag{2}$$

for all $i = 1, 2, \ldots, n$

Optimization Problem

3. and subject to

$$\sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1 \tag{3}$$

4. and to

$$\varepsilon_i \geq 0; \qquad \sum_{i=1}^{n} \varepsilon_i \leq C \tag{4}$$

   where C is a nonnegative tuning parameter.

This optimization problem may become intractable for high dimensional feature space extension → **Kernel Trick**

Kernel Trick: Support Vector Classifiers Rewritten

- The **linear support vector classifier** can be represented as

$$f(x^*) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x^*, x_i \rangle \tag{5}$$

where x_i denotes *training observations* and x^* denotes a *new* observation we want to classify

- The inner product of two observations $x_i, x_{i'}$ is given by

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{r} x_{ij} \cdot x_{i'j} \tag{6}$$

Kernel Trick: Support Vector Classifiers Rewritten

- To estimate the parameters $\alpha_1, \dots, \alpha_n$ and β_0, all we need are the $\binom{n}{2}$ inner products $\langle x_i, x_{i'} \rangle$ between all pairs of training observations where $\binom{n}{2}$ means $n(n-1)/2$, and gives the number of pairs among a set of n items

- However, it turns out that α_i is **nonzero** only for the support vectors in the solution. If a training observation is not a support vector, then its α_i equals zero

- If $\mathcal{S}$ is the collection of indices of these support vectors, we can rewrite any support vector classifier function as

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x^*, x_i \rangle \tag{7}$$

Kernel Trick: Kernel Functions

- **Support vector machine** (SVM) is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**

- In representing the linear classifier

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x^*, x_i \rangle \tag{8}$$

  and in computing its coefficients, all we need are **inner products**

- Every time the inner $\langle x, x_i \rangle$ appears in the representation for the support vector classifier, we replace it with a generalization of the inner product of the form

$$K(x_i, x_{i'}) \tag{9}$$

Kernel Trick: Kernel Functions

- This results in the **support vector machine**

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x_i, x_{i'}) \tag{10}$$

where

$$K(x_i, x_{i'}) \tag{11}$$

is called **kernel function**

- **Kernel trick:** Using such a generalized kernel instead of the standard linear kernel in the support vector classifier algorithm amounts to fitting a support vector classifier in a higher-dimensional space involving extended features!

Linear Kernel

- If we choose in

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x^*, x_{i'}) \tag{12}$$

for $K(x_i, x_{i'})$ the **linear kernel function**

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{13}$$

this would just give us back the support vector classifier

- **Linear kernel** essentially quantifies the similarity of a pair of observations using Pearson (standard) correlation

Polynomial Kernel Function

- Now we choose in

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x^*, x_{i'}) \tag{14}$$

for $K(x_i, x_{i'})$ the **polynomial kernel function** of degree d

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d \tag{15}$$

where d is a positive polynomial integer

- Using such a kernel with $d > 1$, instead of the standard linear kernel, in the support vector classifier algorithm leads to a much **more flexible decision boundary**

- Please see example 3.1 of the chapter Support Vector Machines

Radial Kernel

- Now, we choose in

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x^*, x_{i'}) \tag{16}$$

for $K(x_i, x_{i'})$ the **radial kernel**

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right) \tag{17}$$

where γ is a positive constant

Radial Kernel

- If a given test observation $x^* = (x_1^* \ldots, x_p^*)$ is **far** from training observation x_i wrt Euclidean distance , then

$$\sum_{j=1}^{p}(x_j^* - x_{i'j})^2$$

  will be **large**

- Then

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2\right) \tag{18}$$

  will be **tiny**

Radial Kernel

- Recall that the predicted class label for the test observation x^* is based on the sign of

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x^*, x_i) \tag{19}$$

- In other words, training observations x_i that are far from x^* will play essentially **no** role in the predicted class label for x^*

- This means that the radial kernel has very **local** behavior, in the sense that only nearby training observations have an effect on the class label of a test observation

- Please see example `3.2` of the chapter `Support Vector Machines`

Advantages of Kernel Functions

- What is the advantage of using a kernel rather than simply enlarging the feature space using functions of the original features?
- One advantage is computational, and it amounts to the fact that using kernels, for training one need only compute

$$K(x_i, x_{i'})$$

for all $\binom{n}{2}$ distinct training observation pairs i, i'

Advantages of Kernel Functions

- Training a SVM by computing

$$K(x_i, x_{i'})$$

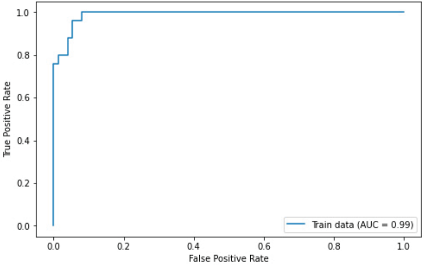
for all $\binom{n}{2}$ distinct training observation pairs i, i' can be done without explicitly working in the enlarged feature space

- This is important because in many applications of SVMs, the enlarged feature space is so large that computations are **intractable**

- For some kernels, such as the radial kernel, the feature space is **implicit** and **infinite-dimensional**, so we could never do the computations there anyway.

ROC Curves

- The **ROC curve** is a popular graphic for simultaneously displaying the two types of errors - **false positive** and **true positive rates** - for all possible thresholds.



- The name **ROC** is historic, and comes from communications theory. It is an acronym for **receiver operating characteristics**.

ROC Curves

- SVMs output class labels for each observation
- However, it is also possible to obtain **fitted values** for each observation, which are the numerical scores (distances to separating hyperplane) used to obtain the class labels
- For an SVM with a **non-linear kernel**, the equation that yields the **fitted value** is given in

$$f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}}^{n} \alpha_i K(x^*, x_i) \tag{20}$$

ROC Curves

- Relationship between the **fitted value** and the **class prediction** for a given observation is simple:
  - If the fitted value $f(x^*)$ **exceeds** a given **threshold**, then the observation is assigned to one class
  - If the fitted value $f(x^*)$ is **less** than this threshold, then it is assigned to the other class
- Threshold usually is zero, but we may choose a different value for the threshold

ROC Curves

- A ROC curve is constructed by:
  1. Choosing threshold value
  2. Computing fitted values $f(x^*)$ of observations
  3. Classifying observations with respect to the chosen threshold
  4. Computing the related true and false positive rates
  5. For each threshold, display the corresponding true and false positive rate as ROC curve
- See example 4.1 of the `Support Vector Machine` chapter

SVMs with More than Two Classes

- So far, our discussion has been limited to the case of **binary classification**: that is, classification in the **two-class setting**
- How can we extend SVMs to the more general case where we have some arbitrary number of classes?
- The two most popular are the **one-versus-one** and **one-versus-all** approaches

One-versus-One Classification

- A **one-versus-one** or **all-pairs** approach constructs $\binom{K}{2}$ SVMs, each of which compares a pair of classes
- **Example:** one such SVM might compare the kth class, coded as $+1$, to the k'th class, coded as -1
- We classify a test observation using each of the $\binom{K}{2}$ classifiers, and we tally the number of times that the test observation is assigned to each of the K classes
- The final classification is performed by assigning the test observation to **the class to which it was most frequently assigned** in these $\binom{K}{2}$ pairwise classifications

One-versus-All Classification

- We fit K SVMs, each time comparing **one** of all the K classes to the **remaining** $K-1$ classes
- Let $\beta_{0k}, \alpha_{1k}, \ldots, \alpha_{pk}$ denote the parameters that result from fitting an SVM comparing the kth class (coded as $+1$) to the others (coded as -1)
- Let x^* denote a test observation. We assign the observation to the class for which

$$f(x^*) = \beta_{0k} + \sum_{i \in \mathcal{S}} \alpha_{ik} K(x^*, x_i) \tag{21}$$

  is **largest**

- This amounts to a high level of confidence that the test observation belongs to the kth class rather than to any of the other classes.

Application to Gene Expression Data

- `Khan` data set consists of a number of tissue samples corresponding to four distinct types of small round blue cell tumors
- For each tissue sample, 2308 gene expression measurements are available
- **Goal:** By means of support vector machines to predict cancer subtype using gene expression measurements
- Please check example `5.3` of the `Support Vector Machine` chapter