Support vector machines

This reading material is based on Chapters in Friedman et al. (2001).

1 How to solve the optimisation problem for the maximal margin classifier

For the maximal margin classifier, we solve the following optimisation problem to get the maximal margin hyperplane:

$$
\begin{aligned}
&\max_{\beta_0,\beta,M} \; M \\
&\text{s.t.} ||\boldsymbol{\beta}|| = 1 \\
&\quad y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) \geqslant M, \;\; i = 1, \ldots, N.
\end{aligned}
$$

We can drop the $||\boldsymbol{\beta}|| = 1$ condition by written the last condition as

$$\frac{1}{||\boldsymbol{\beta}||} y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) \geqslant M,$$

or

$$y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) \geqslant M||\boldsymbol{\beta}||.$$

Since for an $\boldsymbol{\beta}$ and β_0 satisfying these inequalities, any positively scaled multiple satisfies them too, we can arbitrarily set $||\boldsymbol{\beta}|| = 1/M$. Thus the optimisation problem is equivalent to

$$
\begin{aligned}
&\min_{\beta_0,\boldsymbol{\beta}} \; \frac{1}{2}||\boldsymbol{\beta}||^2 \\
&\text{s.t.} \;\; y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) \geqslant 1, \;\; i = 1, \ldots, N.
\end{aligned}
$$

This is a convex optimisation problem, which can be solved based on the Lagrange function:

$$L_P = \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^{N} \alpha_i[y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - 1].$$

Taking the first-orider derivatives with respect to $\boldsymbol{\beta}$ and β_0 and setting them to zeros, we have

$$\boldsymbol{\beta} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i, \tag{1.1}$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i. \tag{1.2}$$

Substitutting the above two equations back to the Lagrange function, we maximise

$$
\begin{aligned}
L_D = &\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \mathbf{x}_i^T \mathbf{x}_k \\
&\text{s.t.} \;\; \alpha_i \geqslant 0 \;\; \text{and} \;\; \sum_{i=1}^{N} \alpha_i y_i = 0.
\end{aligned} \tag{1.3}
$$

This simpler convex optimisation problem can be solved by standard software. The solution must satisfy the Karush-Kuhn-Tucker conditions, which include (1.1), (1.2) and (1.3) and

$$\alpha_i[y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) - 1] = 0 \;\; \forall i.$$

2 How to solve the optimisation problem for the support vector classifier

For the support vector classifier, we solve the following optimisation problem

$$
\begin{aligned}
\max_{\beta_0, \boldsymbol{\beta}, \xi_1, \ldots, \xi_N, M} \quad & M \\
\text{s.t.} \;\; & ||\boldsymbol{\beta}|| = 1 \\
& y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) \geqslant M(1 - \xi_i), \\
& \xi_i \geqslant 0, \sum_{i=1}^{N} \xi_i \leqslant C, \;\; i = 1, \ldots, N.
\end{aligned}
$$

We can drop the $|\boldsymbol{\beta}|| = 1$ condition by using $||\boldsymbol{\beta}|| = 1/M$. Thus the above problem is equivalent to

$$
\begin{aligned}
\min_{\beta_0, \boldsymbol{\beta}, \xi_1, \ldots, \xi_N} \quad & \frac{1}{2}||\boldsymbol{\beta}||^2 \\
\text{s.t.} \;\; & y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) \geqslant (1 - \xi_i), \\
& \xi_i \geqslant 0, \sum_{i=1}^{N} \xi_i \leqslant C \;\; \forall i.
\end{aligned}
$$

We can further re-express the above problem as

$$
\begin{aligned}
\min_{\beta_0, \boldsymbol{\beta}, \xi_1, \ldots, \xi_N} \quad & \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{N} \xi_i \\
\text{s.t.} \;\; & y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) \geqslant (1 - \xi_i), \\
& \xi_i \geqslant 0 \;\; \forall i.
\end{aligned}
$$

for computational convenience.

The Lagrange function is

$$L_P = \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i.$$

Taking the first-order derivatives with respect to $\boldsymbol{\beta}$, β_0 and ξ_i and setting them to zeros, we obtain

$$\boldsymbol{\beta} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i, \tag{2.1}$$

$$0 = \sum_{i=1}^{N}\alpha_i y_i, \tag{2.2}$$

$$\alpha_i = C - \mu_i, \forall i, \tag{2.3}$$

as well as $\alpha_i,\ \mu_i,\ \xi_i\ \geqslant 0\ \forall i$. Substituting the above constraints to the Lagrange function, we maximise

$$\begin{aligned} L_D = &\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N}\alpha_i\alpha_k y_i y_k \mathbf{x}_i^T\mathbf{x}_k \\ &\text{s.t. } 0 \leqslant \alpha_i \leqslant C, \\ &\sum_{i=1}^{N}\alpha_i y_i = 0. \end{aligned} \tag{2.4}$$

The solution should satisfy the KKT conditions, which include

$$\alpha_i[y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1 - \xi_i)] = 0, \tag{2.5}$$

$$\mu_i\xi_i = 0, \tag{2.6}$$

$$y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1 - \xi_i) \geqslant 0, \forall i, \tag{2.7}$$

as well as (2.1), (2.2) and (2.3). This simpler convex optimisation problem can be solved by standard techniques.

3 Kernels in SVM

From (2.4), we can see where to involve kernel. The Lagrange function can be written as

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N}\alpha_i\alpha_k y_i y_k \langle\mathbf{x}_i, \mathbf{x}_k\rangle.$$

If we transform the original features using a function $\phi(\cdot)$, then

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_k)\rangle.$$

The solution can be written as

$$f(\mathbf{x}) = h(\mathbf{x})^T \boldsymbol{\beta} + \beta_0 = \sum_{i=1}^{N} \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_k)\rangle + \beta_0.$$

We need not specify the transformation h(x) at all, but require only knowledge of the kernel function

$$K(\mathbf{x}_i, \phi(\mathbf{x}_k)) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_k)\rangle$$

that computes inner products in the transformed space. The solution can then be written based on the kernel function

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_k) + \beta_0.$$

4 Support vector regression

For a linear regression model

$$f(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\beta} + \beta_0,$$

we minimise the following objective function with a ‘loss function+penalisation’ form

$$H(\boldsymbol{\beta}, \beta_0) = \sum_{i=1}^{N} V(y_i - f(\mathbf{x}_i)) + \frac{\lambda}{2}||\boldsymbol{\beta}||^2,$$

where

$$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon \\ |r| - \epsilon, & \text{otherwise.} \end{cases}$$

We can see that $V_\epsilon(r)$ depends on the value of ϵ and that’s why SVR is called ‘eps-regression’ in the e1071 package. $V_\epsilon(r)$ has a similar property as the hinge loss in SVM: the points with small residuals, $y_i - f(\mathbf{x}_i)$, are ignored in the optimisation.

If $\hat{\boldsymbol{\beta}}$ and $\hat{\beta}_0$ are the solutions by minimising H, the solution function can be shown to have the form

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i)\mathbf{x}_i,$$

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i)\langle \mathbf{x}, \mathbf{x}_i \rangle + \beta_0.$$

The above solutions only depend on the inner products, thus we can generalise the method to richer spaces by using the kernel trick.

References

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The elements of statistical learning*, Volume 1. Springer series in statistics New York.