PROJECT 1 TASK 1 MULTI-CLASS SUPPORT VECTOR MACHINE

ADAM LINDHE

A multi-class support vector machine. This section provides some additional details on the multi-class support vector machine encountered in the project.

We consider a classification problem with C classes. Suppose we have training data in the form of $\{(x_i, y_i)\}_{i=1}^n$ that are outcomes of IID pairs $\{(X_i, Y_i)\}_{i=1}^n$ where X_i represents the explanatory variables and Y_i the corresponding label. We also have $X_{n+1} = x_{n+1}$ and want to predict the label Y_{n+1}.

We consider a classifier that is represented using a matrix

$$\Theta = [\theta_1 \ \theta_2 \ \cdots \ \theta_C] \in \mathbb{R}^{d \times C},$$

where θ_j is a d-dimensional vector that aims to identify those x that have label j. Given Θ the predicted class for a data vector (image, say) $x \in \mathbb{R}^d$ is

$$\text{argmax}_{j=1,\dots,C} \left(\sum_{i=1}^d x_i \Theta_{ij} \right) = \text{argmax}_{j=1,\dots,C} [\Theta^T x]_j.$$

A way to train the classifier is to minimize the empirical hinge-loss given by

$$\frac{1}{n} \sum_{k=1}^n \max_{j \neq y_k} \left[1 + \sum_{i=1}^d x_{ki} (\Theta_{ij} - \Theta_{iy}) \right]_+,$$

where $t_+ = \max(t, 0)$ is the positive part and we write $x_k = (x_{k1}, \dots, x_{kd})^{\mathrm{T}}$.

We can interpret this as a decision problem in the following way. Let the action space be $\aleph = \mathbb{R}^C$ and let the decision rules be parametrized by Θ with $\delta(x) = \Theta^{\mathrm{T}} x$. That is, $\delta(x)$ is a C-dimensional vector, whose components are $\theta_j^{\mathrm{T}} x$ can be interpreted as the level of support for class j. The loss function is the hinge-loss $L : \{1, \dots, C\} \times \mathbb{R}^C \to [0, \infty)$ given by

$$L(y, a) = \max_{j \neq y} [1 + a_j - a_y]_+.$$

Note that this loss function takes into account the level of support of all classes and $L(y, a) = 0$ if and only if $a_y \geq a_j + 1$ for all $j \neq y$. That is, the loss is zero if and only if the level of support for class y has a sufficiently high margin in relation to the other classes.

The Bayes risk of a decision rule δ is given by

$$R(\delta) = \mathrm{E}[L(Y_{n+1}, \delta(X_{n+1}))] = \mathrm{E}[\max_{j \neq Y_{n+1}} [1 + \sum_{i=1}^d X_{(n+1)i} (\Theta_{ij} - \Theta_{iy})]_+.$$

Since the pairs $\{(X_i, Y_i)\}_{i=1}^n$ are IID the Bayes risk can be approximated by the empirical risk

$$R^{\text{emp}}(\delta) = \frac{1}{n} \sum_{k=1}^{n} \max_{j \neq y_k} \left[1 + \sum_{i=1}^{d} x_{ki} (\Theta_{ij} - \Theta_{iy}) \right]_+ .$$

Since the empirical risk is viewed as a function of Θ, we write $R^{\text{emp}}(\Theta)$ instead of $R^{\text{emp}}(\delta)$.

In the project assignment you will show that $R^{\text{emp}}(\Theta)$ is convex, calculate a subgradient and implement a stochastic subgradient algorithm for minimizing the empirical risk.