Kernelizations

David Rosenberg

New York University

February 18, 2015

Linear SVM

- The SVM prediction function is the solution to

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n}\left(1 - y_i\left[w^T x_i + b\right]\right)_+ .$$

- Found it's equivalent to solve the dual problem to get α^*:i

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_j^T x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{c}{n}\right] \; i = 1, \ldots, n. \end{aligned}$$

- Notice: x's only show up as inner products with other x's.

Kernelization

Definition

We say a machine learning method is **kernelized** if all references to inputs $x \in \mathcal{X}$ are through an inner product between pairs of points $\langle x, y \rangle$ for $x, y \in \mathbf{R}^d$.

So far, we've only partially kernelized SVM

We've shown that the training portion is kernelized. Later we'll show the prediction portion is also kernelized.

SVM Dual Problem

- x's only show up in pairs of inner products: $x_j^T x_i = \langle x_j, x_i \rangle$:

$$\sup_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_j, x_i \rangle$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i \in \left[0, \frac{c}{n}\right] \; i = 1, \ldots, n.$$

- Then primal optimal solution is given as:

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

and for any $\alpha_i \in \left(0, \frac{c}{n}\right)$,

$$b^* = y_i - x_i^T w^*.$$

SVM: Kernelizing b

- We found that for any j with $\alpha_j \in \left(0, \frac{c}{n}\right)$:

$$\begin{aligned} b^* &= y_j - x_j^T w^* \\ &= y_j - x_j^T \left(\sum_{i=1}^n \alpha_i^* y_i x_i \right). \\ &= y_j - \sum_{i=1}^n \alpha_i^* y_i \langle x_j, x_i \rangle . \end{aligned}$$

- What about kernelizing w^*?

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i$$

- Not obvious...
- But we really only care about kernelizing the predictions $f^*(x)$.

SVM: Kernelizing Predictions $f^*(x)$

- For any j with $\alpha_j \in \left(0, \frac{c}{n}\right)$:

$$\begin{aligned} f^*(x) &= x^T w^* + b^* \\ &= x^T \left(\sum_{i=1}^n \alpha_i^* y_i x_i \right) + b^* \\ &= \sum_{i=1}^n \alpha_i^* y_i \langle x_i, x \rangle + \left(y_j - \sum_{i=1}^n \alpha_i^* y_i \langle x_j, x_i \rangle \right) \end{aligned}$$

- We now have a fully kernelized version of SVM.
- Can we kernelize the primal version of the SVM?

Kernelizing the SVM Primal Problem

- Primal SVM

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum_{i=1}^{n} \left(1 - y_i\left[w^T x_i + b\right]\right)_+ .$$

- From our study of the dual, found that

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i .$$

- So w^* is a linear combination of the input vectors.
- Restrict to optimization to w of the form

$$w = \sum_{i=1}^{n} \beta_i x_i .$$

Some Vectorization

- **Design matrix** $X \in \mathbf{R}^{n \times d}$ has input vectors as rows:

$$X = \begin{pmatrix} -x_1- \\ \vdots \\ -x_n- \end{pmatrix}.$$

- The contraint on w looks like

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix} = \begin{pmatrix} | & \cdots & | \\ x_1 & \cdots & x_n \\ | & \cdots & | \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} = X^T \beta.$$

- So replace all w with $X^T \beta$, with $\beta \in \mathbf{R}^n$ unrestricted.

The Kernel Matrix (or the Gram Matrix)

Definition

For a set of $\{x_1, \ldots, x_n\}$ and an inner product $\langle \cdot, \cdot \rangle$on the set, the **kernel matrix** or the **Gram matrix** is defined as

$$K = \left(\langle x_i, x_j \rangle \right)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}.$$

Then for the standard Euclidean inner product $\langle x_i, x_j \rangle = x_i^T x_j$, we have

$$K = XX^T$$

Some Vectorization

- Regularization Term:

$$\|w\|^2 = w^T w = \beta^T X X^T \beta = \beta^T K \beta$$

- Prediction on training point x_i:

$$\begin{aligned} f(x_i) &= b + x_i^T w \\ &= b + x_i^T \left(\sum_{j=1}^{n} \beta_j x_j \right) \\ &= b + \sum_{j=1}^{n} \beta_j K_{ij} \end{aligned}$$

Kernelized Primal SVM

- Putting it together, kernelized primal SVM is

$$\min_{\beta \in \mathbf{R}^n, b \in \mathbf{R}} \frac{1}{2}\beta^T K \beta + \frac{c}{n} \sum_{i=1}^{n} \left(1 - y_i \left[b + \sum_{j=1}^{n} \beta_j K_{ij} \right] \right)_+ .$$

- We can write this as a differentiable, constrained optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\beta^T K \beta + \frac{c}{n}\mathbf{1}^T \xi \\ \text{subject to} \quad & \xi \succeq 0 \\ & \xi \succeq (\mathbf{1} - Y[b + K\beta]), \end{aligned}$$

where $Y = \text{diag}(y_1, \ldots, y_n)$, $\mathbf{1}$ is a column vector of 1's, and $\succeq$ represent element-wise vector inequality.

Kernelized Primal SVM: Kernel Trick

- Kernelized primal SVM is

$$\min_{\beta \in \mathbf{R}^n, b \in \mathbf{R}} \frac{1}{2} \beta^T K \beta + \frac{C}{n} \sum_{i=1}^{n} \left(1 - y_i \left[b + \sum_{j=1}^{n} \beta_j K_{ij} \right] \right)_+ .$$

- We derived this with $K = XX^T$, which corresponds to the linear kernel.
- We can now swap out K for any other kernel matrix K' on the same points.
- For example, can use RBF kernel function.