CS 596: Theoretical Machine Learning

Lecturer: Pranjal Awasthi
Scribe: Malihe Alikhani

Lecture 11
October 26, 2016

1 Recap

In the last lecture we looked at SVMs and soft SVMs. In soft margin SVMs we optimize

$$\min_x \frac{1}{m}\sum_{i=1}^{m}(1 - y_i(w \cdot X_i))_+ + \lambda||w||^2$$

when $(x)_+ = \max(0, x)$.

We also proved a theorem that if w_S is the output of soft margin SVM on S, then with probability at least $1 - \delta$,

$$err(w_S) \leq margin_{err}(w_S) + 2\beta + (4\beta + 2)\sqrt{\frac{log\frac{1}{\delta}}{2m}}$$

where $\beta = \frac{1}{2\lambda m}$.

2 Kernel Methods

In this lecture we will see how to extend the ideas from SVMs in non-linear settings. Consider the following example. How can we separate this data?

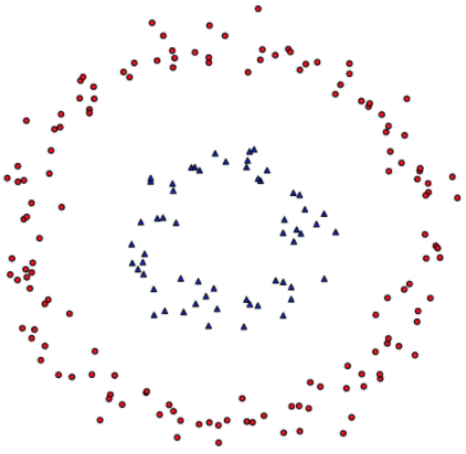

Figure 1: The equation of the red circle is $X^2 + Y^2 = 4$ and the equation of the blue circle is $X^2 + Y^2 = 1$

Linear functions won't help us here. Instead we want to learn a degree 2 polynomial. This can b viewed as a linear function in a higher dimensional feature space given by the mapping Φ

$$\Phi(X_1, X_2) \rightarrow (1, \sqrt{2}X_1, \sqrt{2}X_2, X_1^2, X_2^2, \sqrt{2}X_1X_2).$$

We can also extend the idea in a natural way to induce mappings to a feature space that captures degree d polynomials. We would like to understand if one can learn linear functions in this higher dimensional space in polynomial time. The obvious approach to explicitly map every example and run soft SVM in the new space will be too expensive. Let's consider the soft SVM objective again. We want to find w in the new space that minimizes

$$\frac{1}{m}\sum_{i=1}^{m}(1 - y_i(w \cdot \Phi(X_i))_+ + \lambda||w||^2$$

. Similar to soft SVMs, we know that the optimal w will be a linear combination of the examples, i.e., $w^* = \sum_{j=1}^{m} \alpha_j y_j \Phi(X_j)$. Hence we can convert this into a minimization problem over the α variables.

$$\min_{\alpha_1,\alpha_2,\ldots\alpha_m \geq 0} \frac{1}{m}\sum_{i=1}^{m}(1 - y_i \sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \Phi(X_i) \cdot \Phi(X_j))_+ + \lambda \sum_{i,j} \alpha_i\alpha_j y_i y_j \Phi(X_i) \cdot \Phi(X_j).$$

Notice that the above minimization problem only needs to compute dot products $\Phi(X_i) \cdot \Phi(X_j)$. The following claim shows that this can be done without explicitly maintaining the feature mapping

Claim 1.

$$\Phi(X_i) \cdot \Phi(X_j) = (1 + X_i \cdot X_j)^d$$

Similarly, after computing w^*, the prediction on a new example can be obtained efficiently as

$$w^* \cdot \Phi(X) = \sum_{j=1}^{m} \alpha_j y_j \Phi(X_j) \cdot \Phi(X) = \sum_{j=1}^{m} \alpha_j y_j (1 + X_j \cdot X)^d$$

Next we want to understand if the above approach can be extended to an arbitrary class of functions, i..e when can one minimize the criteria below efficiently

$$\min_{f \in H} \frac{1}{m}\sum_{i=1}^{m} l(f(X_i), y_i) + \lambda||f||^2$$

We will see that the above minimization can be done in polynomial time if $l()$ is a convex function in its first argument and H is a *Reproducing Kernel Hilbert Space(RKHS)*.

Definition 1. (Hilbert Space) A Hilbert space is a vector space of functions $f : X \to R$ that:

1. has a dot product.
2. is complete.

Definition 2. (Reproducing Kernel Hilbert Space(RKHS))
A Hilbert space H is an RKHS if :

1. For every $\forall x \in X$, there exists a corresponding function $f_x \in H$.
2. For any $x \in X$ and $g \in H$, $g(x) = g \cdot f_x$.

Definition 3 (Kernel Function)**.** Associated with an RKHS is a unique kernel function $K : X \times X \mapsto \Re$, such that $K(x, y) = f_x \cdot f_y$. This kernel is a positive semi-definite kernel. In other words, for each $m \geq 1$ and $x_1, x_2, \ldots x_m \subseteq X$, the $m \times m$ matrix M such that $M_{i,j} = K(x_i, x_j)$ has non-negative eigenvalues.

It is also true that any positive semi-definite kernel induces a unique RKHS. Now we present the main theorem that helps us optimize efficiently over an RKHS.

Theorem 2. (Representer Theorem) Suppose that we want to solve the following

$$f^* = arg\min_{f\in H} \frac{1}{m}\sum_{i=1}^{m} l(f(X_i), y_i) + \lambda||f||^2$$

where H is an RKHs with the associated kernel function $K()$. Then f^* is of the form

$$f^* = \sum_{j=1}^{m} \alpha_j f_{X_j}()$$

and

$$f^*(X_i) = \sum_{j=1}^{m} \alpha_j f_{X_j}(X_i) = \sum_{j=1}^{m} \alpha_j k(X_i X_j)$$

Proof. Let $H_0 = \{f_{X_i}() : X_i \in S\}$ and $H_1 = \{g \in H : g(g(X_i) = 0, \forall i \in S\}$.
Notice that H_0 and H_1 are orthogonal. Hence, we can decompose the optimal function f^* as

$$f^* = f_0 + f_1$$

, where $f_0 \in span(H_0)$ and $f_1 \in span(H_1)$. Then we have that

$$\frac{1}{m}\sum_{i=1}^{m} l(y_i, f_0(X_i) + f_1(X_i)) + \lambda||f_0||^2 + \lambda||f_1||^2$$

$$= \frac{1}{m}\sum_{i=1}^{m} l(y_i, f_0(X_i)) + \lambda||f_0||^2 + \lambda||f_1||^2 \geq \frac{1}{m}\sum_{i=1}^{m} l(y_i, f_0(X_i)) + \lambda||f_0||^2$$

We've shown that the cost of f_0 is less than the cost of f^* and that contradicts the optimality of f^*.

□

Example 1. A popular kernel function to use is the Gaussian kernel defined as

$$k(x, x') = e^{\frac{-||x-x'||^2}{2\sigma^2}}$$

Finally, as in the last lecture we can provide bounds on the generalization ability of any minimizer over an RKHS using algorithmic stability

Theorem 3. If A operates as follows :

$$A(s) = arg\min_{f\in H} \frac{1}{m}\sum_{i=1}^{m} l(f(X_i), y_i) + \lambda||f||^2$$

and

1. H is an RKHS with an associated kernel $K()$.
2. $l(f(X_i), y_i)$ is convex in the first argument.
3. l is σ -lipschitz that is $|l(a, y) - l(b, y)| \leq \sigma|a - b|$.

Theorem 4. A is β - stable for $\beta = \frac{\sigma R^2}{2\lambda m}$, and $R = \max_{x\in X} k(x, x)$.

2.1 Additional Reading

- Chapter 16 of the book: `http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf`.
- More detailed survey by Hofmann, Schölkopf and Smola: `https://arxiv.org/pdf/math/0701907.pdf`.