COS-511: Learning Theory **Spring 2017**

Lecture 15: Kernel Methods

Lecturer: Roi Livni

15.1 Kernel Methods

So far we've discussed learning linear classes, while many problems may be solved using a linear classifier we would like to somehow elevate the tools developed so far to non–linear problems. One way to reduce non–linear problems into a linear formalization is through *high feature space embeddings*. For example, assume we wish to learn a polynomial over our data. i.e. we want to learn the optimal two degree multivariate polynomial over the domain $\mathcal{X}$. The optimization problem then becomes:

$$\begin{aligned} \text{minimize} \quad & \ell(p(\mathbf{x}), y) \\ \text{s.t.} \quad & p \in \{f : f(\mathbf{x}) = \sum_{|I|=2, I\subseteq[n]} \alpha_I \Pi_{i\in I} \mathbf{x}^{(i)}\} \end{aligned}$$

p is no longer linear in $\mathbf{x}$ and we need to parametrize it correctly in order to correctly learn it. One way to do that is by considering $p(x)$ is a linear functional over an embedding of $\mathbf{x}$ in a high dimensional feature space. Namely consider the embeding $\phi : \mathbf{x} \to \mathbb{R}^{1+n+\binom{n}{2}} = \mathbb{R}^{O(d^2)}$ by

$$(\phi(\mathbf{x}))_I = \Pi_{i\in I} \mathbf{x}^{(i)}$$

(Where I is a multi–index, indexing the elements of $\mathbb{R}^{O(d^2)}$. We can now rewrite the original problem via

$$\begin{aligned} \text{minimize} \quad & \ell(\langle \alpha, \phi(\mathbf{x})\rangle, y) \\ \text{s.t.} \quad & \alpha \in \mathbb{R}^{O(d^2)} \end{aligned}$$

To obtain generalization bounds for the learning problem we can regulate α and consider the problems

$$\begin{aligned} \text{minimize} \quad & \ell(\langle \alpha, \phi(\mathbf{x})\rangle, y) \\ \text{s.t.} \quad & \alpha \in K \end{aligned}$$

Where K is some class with bounded complexity (e.g. ℓ_1 bounded, ℓ_2 bounded vectors etc.). These problems are then linear in α yet allow us to learn much richer classes of functions: Ofcourse the main caveat is that with higher expressivity we will also see deterioration in sample complexity /computational complexity. We next focus on the special case ℓ_2 regularization where some of these obstacles may be overcome (under proper assumptions):

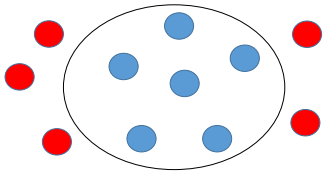


Figure 15.1: By embedding (x_1, x_2) into the feature space of monomials $(x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_1)$, and learning a linear classifier, we can learn a target function of the form (for example): $p(\mathbf{x}) = x_1^2 + x_2^2 - 1$. This target function will assign negative sign to every point in the circle, and positive point to every point outside of the circle. The target function is linear in the ambient space but translates to a non-linear classifier in the original space.

15.1.1 Learning ℓ_2 regulerized classes

Definition 15.1. *A Hilbert Space H is a linear space (i.e. a vector space equipped with addition and multiplication by scalar) equipped with a dot product which is a function $\langle \cdot, \cdot \rangle : H \times H \to \mathbb{R}$ s.t.:*

1. $\langle x, y_1 + y_2 \rangle = \langle x, y_1 \rangle + \langle x, y_2 \rangle$
2. $\langle c \cdot x; y \rangle = c \langle x; y \rangle$
3. $\langle x; y \rangle = \langle y; x \rangle$.

Every dot product in a Hilbert space defines a norm $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}; \mathbf{x} \rangle}$, which satisfies the Cauchy Schwartz inequality:

$$|\langle \mathbf{x}; \mathbf{y} \rangle| \leq \|\mathbf{x}\| \|\mathbf{y}\| \tag{15.1}$$

Definition 15.2. *A Hilbert space is called an RKHS if it is a space of continuous functions $f : \mathcal{X} \to \mathbb{R}$ from a domain $\mathcal{X}$ to $\mathbb{R}$ such that: There exists a continous mapping $\phi : \mathcal{X} \to H$ for which every $f \in H$:*

$$\langle \phi(\mathbf{x}); f \rangle = f(\mathbf{x})$$

The kernel function of the RKHS is defined as $k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \langle \phi(\mathbf{x}^{(1)}); \phi(\mathbf{x}^{(2)}) \rangle$

Note that an RKHS is completely characterized by the kernel function (as far as dot products between points in the domain $\mathcal{X}$). The following criterion is helpfull in identifying kernel functions and RKHS

Theorem 15.3 (Mercer Condition)**.** *A kernel function k defines an RKHS iff and only iff for every finite sample $\{\mathbf{x}^{(i)}\}_{i=1}^m$ the kernel matrix*

$$K_{i,j} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

is psd (positive semidefinite).

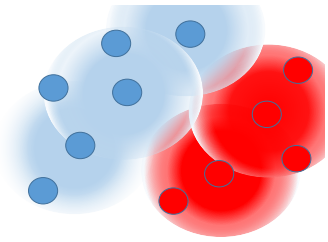
Example 15.1. *The following kernel functions define a kernel space*

1. $k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = (\langle \mathbf{x}^{(1)}; \mathbf{x}^{(2)} \rangle)^d$, *Homogenous Polynomial Kernel*
2. $k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = (1 + \langle \mathbf{x}^{(1)}; \mathbf{x}^{(2)} \rangle)^d$, *Polynomial Kernel*

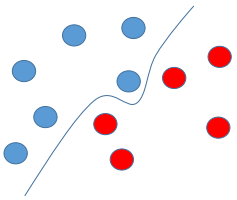
3. $k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \exp^{-\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|^2}$ *Gaussian Kernel*

4. $k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \exp^{-\langle \mathbf{x}^{(1)}; \mathbf{x}^{(2)} \rangle}$ *Exponential Kernel*

There are many other kernels in the literature: What makes many of them helpfull is the fact that we can efficiently compute the kernel: As we will see this allows us to efficiently implement SGD in the Hilbert Space. In fact, it was observed that many algorithm, may be implemented given just an oracle for dot product: SGD is one example but also PCA and other types of optimization algorithm. This observation has led to succefull kernelization of many other algorithms: not just classification.



(a) RBF Kernel: Using the RBF Kernel each point is mapped to a function $k(\mathbf{x}^{(i)}, \mathbf{x}) = e^{-\|\mathbf{x}^{(i)} - \mathbf{x}\|^2/\sigma}$. This mapping assign 1 to points close to $\mathbf{x}^{(i)}$ and zero for far away points. A linear function in the ambient space, translates into a linear combinations of such Radial basis functions.



(b) A polynomial kernel maps a point $\mathbf{x}^{(i)}$ to the function $p_{\mathbf{x}^{(i)}}(\mathbf{x}) = (\gamma + \mathbf{x}^{(i)} \cdot \mathbf{x})^d$ which is a polynomial, thus a linear separator in the RKHS translates into a linear combination of such polynomials, and allows learning an optimal polynomial for classification.

Figure 15.2: Different target functions expressible using kernel methods

The first question that comes to mind is the sample complexity of learning in an RKHS. The following result is an immediate corollary of Lem. 12.3:

Lemma 15.4. *Let H be an RKHS such that $k(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}) \leq 1$. Let*

$$\mathcal{F} = \{\mathbf{w} : \mathcal{X} \to \langle \mathbf{w}; \phi(\mathbf{x}) \rangle : \ \|\mathbf{w}\|_H \leq B\}, \ \}$$

then

$$\mathfrak{R}_m(\mathcal{F}) \leq \frac{B}{\sqrt{m}}$$

As a corollary, for every convex Lipschitz loss function we have that

$$\mathcal{L}_D(\mathbf{w}) \leq \mathcal{L}_S(\mathbf{w}) + O(\frac{L \cdot B \log 1/\delta}{\sqrt{m}})$$

A crucial point is that the result is *dimension – independent*, i.e. the sample complexity of $\mathcal{F}$ is independent of the dimension of H. For the case of polynomial kernels, we get to learn in a feature space of monomials that is order of magnitude of $O(n^d)$, exponential in the degree. For the case of RBF kernel, H is even infinite dimensional – yet we can still obtain tractable sample complexity.

15.1.2 The Expressive power of Kernel Methods

At a first glance kernel methods seem like a very powerfull tool. For example, it is known that any target function may be approximated by a multivariate polynomial of large enough degree: The polynomial kernel

allows one to embed our point in the space of polynomials and thus learn a polynomial of degree d: When one considers the space of all functions in the RKHS, these could have full expressive power (meaning we can approximate any function we might want).

The crucial point to consider is that when learning a linear classifier in an RKHS we allways learn *norm bounded* elements: In contrast with the space of all polynomials, the polynomials that may be expressed using a vector $\mathbf{w} : \mathcal{X} \to \langle \mathbf{w}; \phi(\mathbf{x}) \rangle$ such that $\|\mathbf{w}\|_H \leq B$ is a significantly smaller set then all polynomial of degree d.

15.1.3 The computational cost of learning in an RKHS

At a first glance, learning in a high dimensional feature space might seem prohibitive, especially since we need to deal with high dimensional vector. The next theorem gives a slight insight as to why the problem need not be so prohibitive, as it show that even if the problem is embedded in a very large dimension space: the solution may be searched in a low dimensioal space:

Theorem 15.5 (Representer Theorem)**.** *Let H be some hilbert space and consider a problem of the form:*

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \sum_{i=1}^{m} \ell(\langle \mathbf{x}_i; \mathbf{w} \rangle, y_i) \\ \text{s.t.} \|\mathbf{w}\| \leq B \end{aligned}$$

Then there exists a solution of the form $\mathbf{w}^* = \sum \alpha_i \mathbf{x}^{(i)}$ the minimizes the problem.

Proof. Let $\bar{\mathbf{w}}$ be some solution. Next let $H_S = \text{span}\{\mathbf{x}^{(i)}\}$ and set $H^\perp = \{\mathbf{w} : \langle \mathbf{w}; \mathbf{x}^{(i)} \rangle = 0, i = 1, \dots m\}$. Every element in H $\mathbf{w}$ can be written as $\mathbf{w} = \mathbf{w}_S + \mathbf{w}^\perp$ where $\mathbf{w}_S \in H_S$ and $\mathbf{w}^\perp \in H^\perp$. Indeed set

$$\mathbf{w}_S = \arg \min_{\mathbf{u} \in H_S} \|\mathbf{w} - \mathbf{u}\|^2$$

Then we claim that $\mathbf{w} - \mathbf{w}_S \in H^\perp$. Assume, otherwise then there exists $\mathbf{x}^{(i)}$ s.t. $\langle \mathbf{x}^{(i)}; \mathbf{w} \rangle \neq 0$. Assume that $\langle \mathbf{x}^{(i)}; \mathbf{w} \rangle > 0$, otherwise take $-\mathbf{x}^{(i)}$ in what follows:

Set

$$\bar{\mathbf{w}}_S = \mathbf{w}_S - \langle \mathbf{x}^{(i)}; \mathbf{w} - \mathbf{w}_S \rangle \frac{\mathbf{x}^{(i)}}{\|\mathbf{x}^{(i)}\|^2} \in H_S.$$

Then, setting $\eta_S = \langle \mathbf{x}^{(i)}; \mathbf{w} - \mathbf{w}_S \rangle$:

$$\begin{aligned} \|\mathbf{w} - \bar{\mathbf{w}}_S\|^2 = \|\mathbf{w} - \mathbf{w}_S\|^2 - 2\frac{\eta_S}{\|\mathbf{x}^{(i)}\|^2} \langle \mathbf{w} - \mathbf{w}_S; \mathbf{x}_i \rangle + \frac{\eta_S^2}{\|\mathbf{x}^{(i)}\|^4} \|\mathbf{x}^{(i)}\|^2 = \\ \|\mathbf{w} - \mathbf{w}_S\| - \frac{\eta_S^2}{\|\mathbf{x}^{(i)}\|^2} < \|\mathbf{w} - \mathbf{w}_S\|^2 \end{aligned}$$

Which contradicts minimality. It follows that $\mathbf{w} - \mathbf{w}_S - \in H^\perp$ and $\mathbf{w} = \mathbf{w}_S + \mathbf{w} - \mathbf{w}_S$.

Returning to the original problem let us write $\bar{\mathbf{w}} = \mathbf{w}_S + \mathbf{w}^\perp$ as discussed. then note that

$$\langle \bar{\mathbf{w}}; \mathbf{x}^{(i)} \rangle = \langle \mathbf{w}_S; \mathbf{x}^{(i)} \rangle + \langle \mathbf{w}^\perp; \mathbf{x}^{(i)} \rangle = \langle \mathbf{w}_S; \mathbf{x}^{(i)} \rangle$$

Hence

$$\sum \ell(\langle \bar{\mathbf{w}}; \mathbf{x}^{(i)} \rangle, y_i) = \sum \ell(\langle \mathbf{w}_S; \mathbf{x}^{(i)} \rangle, y_i)$$

and

$$\|\bar{\mathbf{w}}\|^2 = \|\mathbf{w}_S\|^2 + \|\mathbf{w}^\perp\|^2 \geq \|\mathbf{w}_S\|$$

Taken together if $\bar{\mathbf{w}}$ minimizes the problem and $\|\bar{\mathbf{w}}\| \leq B$, then $\mathbf{w}_S$ attains the same value over the objective and also $\|\mathbf{w}_S\| \leq B$. Further $\mathbf{w}_S \in \text{span}(\mathbf{x}^{(i)})$. □

Remark 1. *Using the same proof, note that we can derive a representer theorem also for an objective of the form:*

$$\lambda\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \ell(\langle \mathbf{w}; \mathbf{x}^{(i)} \rangle, y_i)$$

Which is the more standard objective one usually tries to solve when learning convex problems.

The Kernel Trick: The idea behind the kernel trick is to employ the representer theorem to allow efficient computations. Given the Empirical Risk Minimization problem and the representer theorem we know that we can replace the problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{m}\sum \ell(\langle \phi(\mathbf{x}^{(i)}); \mathbf{w} \rangle, y_i) \\ \text{s.t.} \quad & \|\mathbf{w}\| \leq B \end{aligned}$$

By restricting our attention to $\mathbf{w}$ of the form $\mathbf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}^{(i)})$. The norm of $\mathbf{w}$ is then calculated using

$$\|\mathbf{w}\|^2 = \sum \alpha_i \alpha_j \langle \phi(\mathbf{x}^{(i)}); \phi(\mathbf{x}^{(j)}) \rangle = \sum \alpha_i \alpha_j k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \alpha^\top K \alpha$$

Where K is the kernel matrix over the sample i.e. $K_{i,j} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$. Taken together, we can find an efficient solution be solving

$$\begin{aligned} \text{minimize} \quad & \frac{1}{m}\sum \ell(\sum \alpha_j k(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}), y_i) \\ \text{s.t.} \quad & \alpha^\top K \alpha \leq B \end{aligned}$$

Which is a convex problem over m variables: Therefore its complexity depends on the sample size.

15.1.4 Optimization and Representer Theorem, from SGD

We next will see how to optimize the original problem as well as derive the representer theorem, directly from SGD. The idea relies on the following observations

1. the first observation is, as before, given an RKHS with efficiently computable kernel function k, we can always calculate dot products between two points efficiently: without actually having to explicitly embed them in a high dimensional space.

2. The second observation, is that the vectors maintaing by SGD: $\mathbf{w}_t$ can allways be represented as a linear combination of past examples. Indeed if $\mathbf{w}_t$ is the vector at the t-th iteration, the update rule is given by:

$$\mathbf{w}_{t+1} \propto \mathbf{w}_t - \eta_{t+1} \nabla_{t+1}$$

Where $\nabla_{t+1} = \ell'(\langle \mathbf{w}_t, \phi(\mathbf{x}^{(t+1)})\rangle)\phi(\mathbf{x}^{(t+1)})$. Therefore if $\mathbf{w}_t = \sum_{i=1}^{t} \alpha_i \phi(\mathbf{x}^{(i)})$, then $\mathbf{w}_{t+1} \propto \sum_{i=1}^{t} \alpha_i \phi(\mathbf{x}^{(i)}) + \alpha_{t+1}\phi(\mathbf{x}^{(t+1)}$ where

$$\alpha_{t+1} = -\eta_{t+1}\ell'(\langle \mathbf{w}_t, \phi(\mathbf{x}^{(t+1)})\rangle, y_{t+1}).$$

To maintain a representation of $\mathbf{w}_{t+1}$ as a linear sum of $\{\phi(\mathbf{x}^{(i)})\}$ we need to calculate $\langle \mathbf{w}_t, \phi(\mathbf{x}^{(t+1)})\rangle$ (we assume we know how to calculate ℓ'). But given the representation and the kernel function we have:

$$\langle \mathbf{w}_t; \phi(\mathbf{x}^{(t+1)})\rangle = \langle \sum \alpha_i \phi(\mathbf{x}^{(i)}); \mathbf{x}^{(t+1)}\rangle = \sum \alpha_i \langle \mathbf{x}^{(i)}; \mathbf{x}^{(t+1)}\rangle = \sum \alpha_i k(\mathbf{x}^{(i)}, \mathbf{x}^{(t+1)}). \tag{15.2}$$

3. Similarly we can project by calculating the norm given by

$$\|\mathbf{w}_t\| = \sqrt{\langle \sum \alpha_i \mathbf{x}^{(i)}; \sum \alpha_i \mathbf{x}^{(i)}\rangle} = \sqrt{\sum \alpha_i \alpha_j k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})} \tag{15.3}$$

4. Taken together, we can implement the various steps of SGD in the high dimensional space using *implicit calculation of the dot product*

We summarize the kernelized version of SGD in Alg. 1 together with the guarantees in Thm. 15.6 (whose proof is along the lines above to show that indeed the algoirthm implements SGD in a high dimensional space).

$$\langle \mathbf{w}_t; \mathbf{x}^{(t)}\rangle = \sum_{i=1}^{T} \alpha_i k(\mathbf{x}^{(i)}, \mathbf{x}^{(t)})$$

Thus, by calculating the scalar product, we can avoid all calculations in the high dimensional space and maintain a tractable algorithm!!!

Algorithm 1 Kernelized SGD for Learning

0: **Input:** Stochastic sample $\{\mathbf{x}^{(t)}, y_t\}_{t=1}^{T}$ drawn IID, an kernel function k defining an RKHS and a sequence of learning rates $\{\eta_t\}$ $\mathbf{w}_1 = 0$.
for $t = 1, 2 \dots T$ **do**
Let $\alpha_t = \ell_t'(\langle \mathbf{w}_t; \phi(\mathbf{x}^{(t)})\rangle, y_t)$ $\quad \%\langle \mathbf{w}_t; \mathbf{x}^{(t)}\rangle = \sum \alpha_t k(\mathbf{x}^{(t)}, \mathbf{x})$
Update and Project:

$$\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t - \eta_t \alpha_t \cdot \phi(\mathbf{x}^{(t)})$$
$$\mathbf{w}_{t+1} = \frac{B}{\|\mathbf{w}_{t+\frac{1}{2}}\|}\mathbf{w}_{t+\frac{1}{2}} : \quad \%\|\mathbf{w}_{t+\frac{1}{2}}\| = \sum_{i,j \le t} \alpha_i \alpha_j k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

end for
return $\bar{\mathbf{w}}_T = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t$.

Theorem 15.6. *Let k be a kernel and $\phi : \mathcal{X} \to H$ the corresponding embedding in the RKHS.*

Consider the problem $(\mathcal{X}, \mathcal{F}, \ell)$ where ℓ is L-Lipschitz convex loss $\mathcal{X} \subseteq \mathbb{R}^d$ and

$$\mathcal{F} = \{\mathbf{w} \in H : \mathbf{w} : \mathbf{x} \to \langle \phi(\mathbf{x}); \mathbf{w}\rangle, \ \|\mathbf{w}\| \le B\}$$

. Given a access to $\{\mathbf{x}^{(t)}, y_t\}_{t=1}^{T} \sim D$ *pairs distributed according to* D*, Run Alg. 1. After* $T = O\left(\left(\frac{LB}{\epsilon}\right)^2\right)$ *iterations we obtain in expectation:*

$$\mathbb{E}\left[\mathcal{L}_D\left(\bar{\mathbf{w}}_T\right)\right] \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\ell(\mathbf{w}_t \cdot \mathbf{x}^{(t)}, y)\right] \leq \min_{\|\mathbf{w}^*\| \leq B} \mathcal{L}_D(\mathbf{w}^*) + \epsilon$$