Kernels

MIT 15.097 Course Notes

Cynthia Rudin

Credits: Bartlett, Schölkopf and Smola, Cristianini and Shawe-Taylor

The *kernel trick* that I'm going to show you applies much more broadly than SVM, but we'll use it for SVM's.

Warning: This lecture is technical. Try to get the basic idea even if you don't catch all the details.

Basic idea: If you can't separate positives from negatives in a low-dimensional space using a hyperplane, then map everything to a higher dimensional space where you can separate them.

The picture looks like this but it's going to be a little confusing at this point:

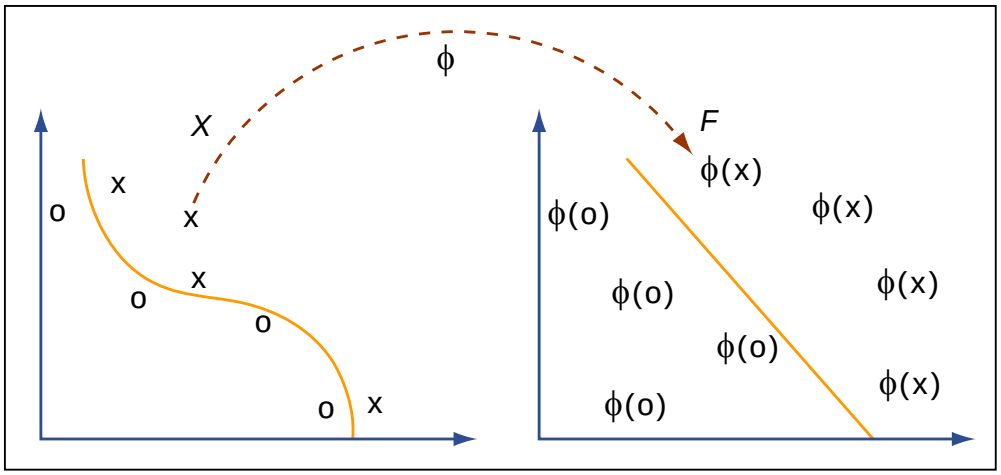


Image by MIT OpenCourseWare.

or it might look like this:

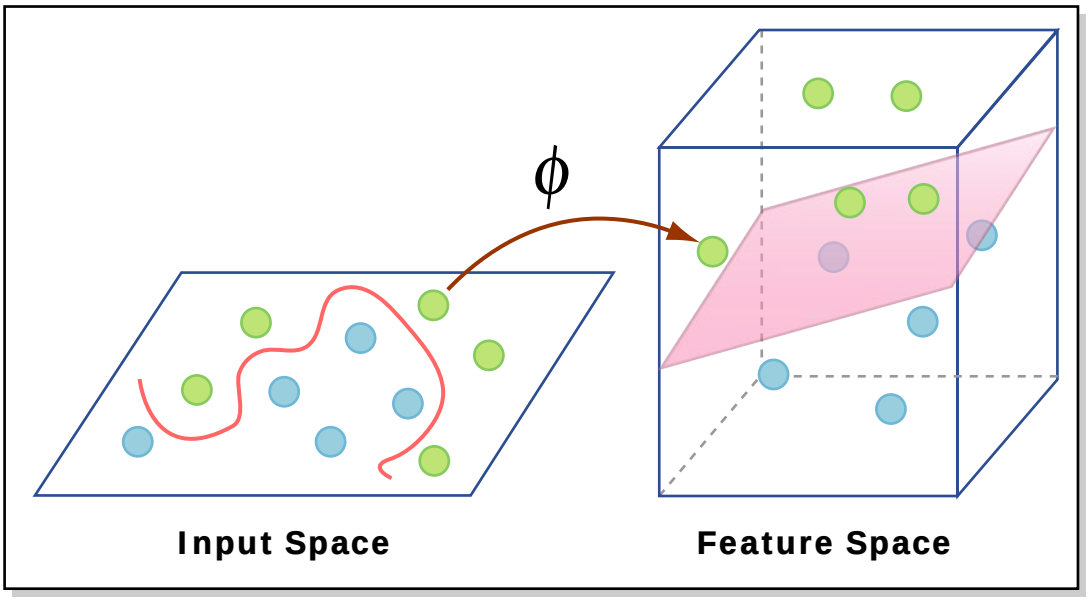


Image by MIT OpenCourseWare.

Say I want to predict whether a house on the real-estate market will sell today or not:

$$\mathbf{x} = \left[\underbrace{x^{(1)}}_{\text{house's list price}}, \underbrace{x^{(2)}}_{\text{estimated worth}}, \underbrace{x^{(3)}}_{\text{length of time on market}}, \underbrace{x^{(4)}}_{\text{in a good area}}, \ldots \right].$$

We might want to consider something more complicated than a linear model:

Example 1: $[x^{(1)}, x^{(2)}] \rightarrow \mathbf{\Phi}\left([x^{(1)}, x^{(2)}]\right) = [x^{(1)2}, x^{(2)2}, x^{(1)}x^{(2)}]$

The 2d space gets mapped to a 3d space. We could have the inner product in the 3d space:

$$\mathbf{\Phi}(\mathbf{x})^T\mathbf{\Phi}(\mathbf{z}) = x^{(1)2}z^{(1)2} + x^{(2)2}z^{(2)2} + x^{(1)}x^{(2)}z^{(1)}z^{(2)}.$$

Example 2:

$$\begin{aligned}[x^{(1)}, x^{(2)}, x^{(3)}] \rightarrow \mathbf{\Phi}&\left([x^{(1)}, x^{(2)}, x^{(3)}]\right) \\ &= [x^{(1)2}, x^{(1)}x^{(2)}, x^{(1)}x^{(3)}, x^{(2)}x^{(1)}, x^{(2)2}, x^{(2)}x^{(3)}, x^{(3)}x^{(1)}, x^{(3)}x^{(2)}, x^{(3)2}]\end{aligned}$$

and we can take inner products in the 9d space, similarly to the last example.

Rather than apply SVM to the original $\mathbf{x}$'s, apply it to the $\mathbf{\Phi}(\mathbf{x})$'s instead.

The kernel trick is that if you have an algorithm (like SVM) where the examples appear only in inner products, you can freely replace the inner product with a different one. (And it works just as well as if you designed the SVM with some map $\mathbf{\Phi}(\mathbf{x})$ in mind in the first place, and took the inner product in $\mathbf{\Phi}$'s space instead.)

Remember the optimization problem for SVM?

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,k=1}^{m} \alpha_i \alpha_j y_i y_k \underbrace{\mathbf{x}_i^T \mathbf{x}_k}_{\leftarrow \text{ inner product}}$$

$$\text{s.t. } 0 \leq \alpha_i \leq C, i = 1, ..., m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0$$

You can replace this inner product with another one, *without even knowing* $\mathbf{\Phi}$. In fact there can be many different feature maps that correspond to the same inner product.

In other words, we'll replace $\mathbf{x}^T\mathbf{z}$ (i.e., $\langle \mathbf{x}, \mathbf{z} \rangle_{\mathbf{R}^n}$) with $k(\mathbf{x}_i, \mathbf{x}_j)$, where k happens to be an inner product in some feature space, $\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{x}) \rangle_{\mathcal{H}_k}$. Note that k is also called the kernel (you'll see why later).

Example 3: We could make $k(\mathbf{x}, \mathbf{z})$ the square of the usual inner product:

$$k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle_{\mathbf{R}^n}^2 = \left(\sum_{j=1}^{n} x^{(j)} z^{(j)} \right)^2 = \sum_{j=1}^{n} \sum_{k=1}^{n} x^{(j)} x^{(k)} z^{(j)} z^{(k)}.$$

But how do I know that the square of the inner product is itself an inner product in some feature space? We'll show a general way to check this later, but for now, let's see if we can find such a feature space.

Well, for $n = 2$, if we used $\mathbf{\Phi}\left([x^{(1)}, x^{(2)}]\right) = \left[x^{(1)2}, x^{(2)2}, x^{(1)}x^{(2)}, x^{(2)}x^{(1)}\right]$ to map into a 4d feature space, then the inner product would be:

$$\mathbf{\Phi}(\mathbf{x})^T \mathbf{\Phi}(\mathbf{z}) = x^{(1)2} z^{(1)2} + x^{(2)2} z^{(2)2} + 2x^{(1)} x^{(2)} z^{(1)} z^{(2)} = \langle \mathbf{x}, \mathbf{z} \rangle_{\mathbf{R}^2}^2.$$

So we showed that k is an inner product for $n = 2$ because we found a feature space corresponding to it.

For $n = 3$ we can also find a feature space, namely the 9d feature space from Example 2 would give us the inner product k.
That is,

$$\mathbf{\Phi}(\mathbf{x}) = (x^{(1)2}, x^{(1)}x^{(2)}, ..., x^{(3)2}), \text{ and } \mathbf{\Phi}(\mathbf{z}) = (z^{(1)2}, z^{(1)}z^{(2)}, ..., z^{(3)2}),$$

$$\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{z}) \rangle_{\mathbf{R}^9} = \langle \mathbf{x}, \mathbf{z} \rangle^2_{\mathbf{R}^3}.$$

That's nice.

We can even add a constant, so that k is the inner product plus a constant squared.

Example 4:

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^T\mathbf{z} + c)^2 = \left(\sum_{j=1}^{n} x^{(j)}z^{(j)} + c\right)\left(\sum_{\ell=1}^{n} x^{(\ell)}z^{(\ell)} + c\right) \\ &= \sum_{j=1}^{n}\sum_{\ell=1}^{n} x^{(j)}x^{(\ell)}z^{(j)}z^{(\ell)} + 2c\sum_{j=1}^{n} x^{(j)}z^{(j)} + c^2 \\ &= \sum_{j,\ell=1}^{n} (x^{(j)}x^{(\ell)})(z^{(j)}z^{(\ell)}) + \sum_{j=1}^{n} (\sqrt{2c}x^{(j)})(\sqrt{2c}z^{(j)}) + c^2, \end{aligned}$$

and in $n = 3$ dimensions, one possible feature map is:

$$\mathbf{\Phi}(\mathbf{x}) = [x^{(1)2}, x^{(1)}x^{(2)}, ..., x^{(3)2}, \sqrt{2c}x^{(1)}, \sqrt{2c}x^{(2)}, \sqrt{2c}x^{(3)}, c]$$

and c controls the relative weight of the linear and quadratic terms in the inner product.

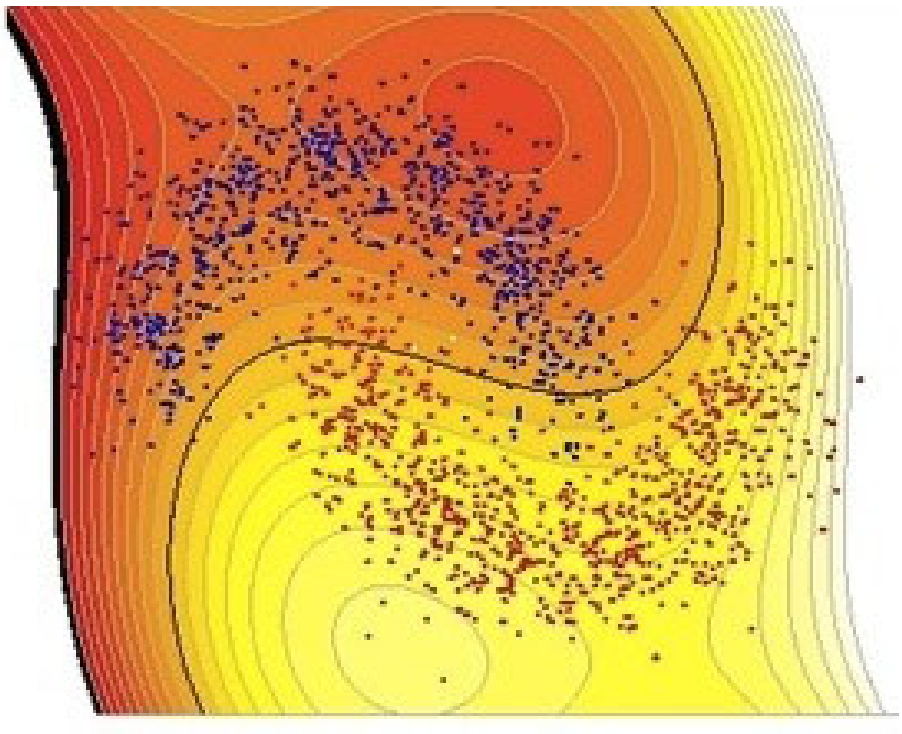
Even more generally, if you wanted to, you could choose the kernel to be any higher power of the regular inner product.

Example 5: For any integer $d \geq 2$

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + c)^d,$$

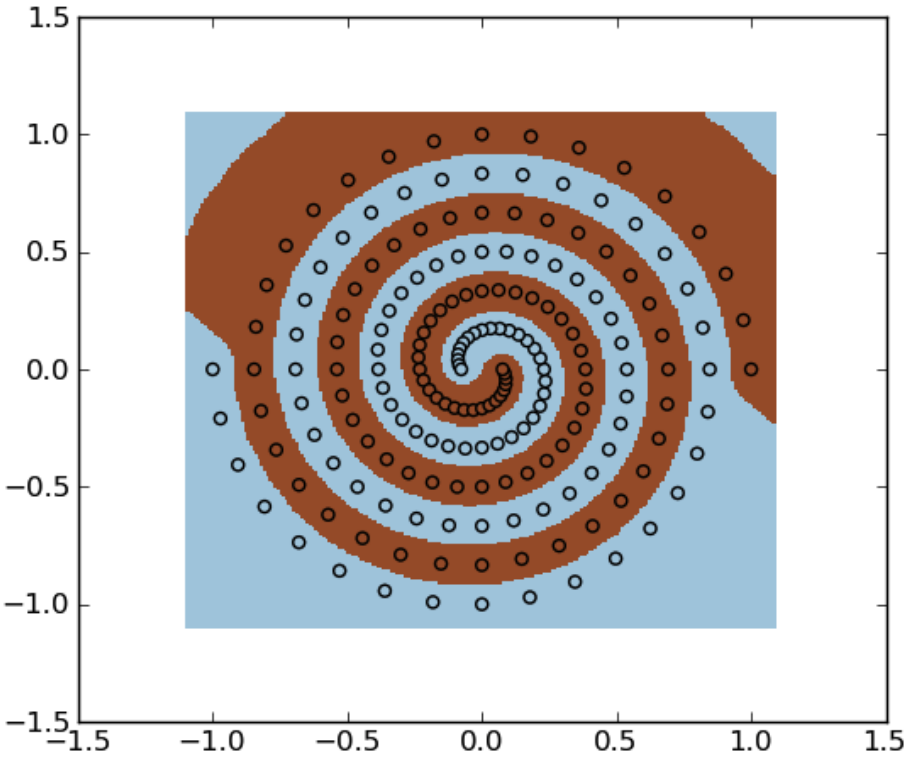
where the feature space $\mathbf{\Phi}(\mathbf{x})$ will be of degree $\binom{n+d}{d}$, with terms for all monomials up to and including degree d. The decision boundary in the feature space (of course) is a hyperplane, whereas in the input space it's a polynomial of degree d. Now do you understand that figure at the beginning of the lecture?

Because these kernels give rise to polynomial decision boundaries, they are called *polynomial kernels*. They are very popular.



Courtesy of Dr. Hagen Knaf. Used with permission.

Beyond these examples, it is possible to construct very complicated kernels, even ones that have infinite dimensional feature spaces, that provide a lot of modeling power:



SVMs with these kinds of fancy kernels are among the most powerful ML algorithms currently (a lot of people say the most powerful).

But the solution to the optimization problem is still a simple linear combination, even if the feature space is very high dimensional.

How do I evaluate $f(\mathbf{x})$ for a test example $\mathbf{x}$ then?

If we're going to replace $\mathbf{x}_i^T \mathbf{x}_k$ everywhere with some function of $\mathbf{x}_i$ and $\mathbf{x}_k$ that is hopefully an inner product from some other space (a kernel), we need to ensure that it really is an inner product. More generally, we'd like to know how to construct functions that are guaranteed to be inner products in some space. We need to know some functional analysis to do that.

Roadmap

1. make some definitions (inner product, Hilbert space, kernel)
2. give some intuition by doing a calculation in a space with a finite number of states
3. design a general Hilbert space whose inner product is the kernel
4. show it has a reproducing property - now it's a Reproducing Kernel Hilbert space
5. create a totally different representation of the space, which is a more intuitive to express the kernel (similar to the finite state one)
6. prove a cool representer theorem for SVM-like algorithms
7. show you some nice properties of kernels, and how you might construct them

Definitions

An *inner product* takes two elements of a vector space $\mathcal{X}$ and outputs a number. An inner product could be a usual dot product: $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}'\mathbf{v} = \sum_i u^{(i)} v^{(i)}$, or it could be something fancier. An inner product $\langle \cdot, \cdot \rangle$ must satisfy the following conditions:

1. Symmetry
$$\langle u, v \rangle = \langle v, u \rangle \quad \forall u, v \in \mathcal{X}$$
2. Bilinearity
$$\langle \alpha u + \beta v, w \rangle = \alpha \langle u, w \rangle + \beta \langle v, w \rangle \quad \forall u, v, w \in \mathcal{X}, \forall \alpha, \beta \in \mathbf{R}$$
3. Strict Positive Definiteness
$$\langle u, u \rangle \geq 0 \quad \forall x \in \mathcal{X}$$
$$\langle u, u \rangle = 0 \iff u = 0.$$

An *inner product space* (or pre-Hilbert space) is a vector space together with an inner product.

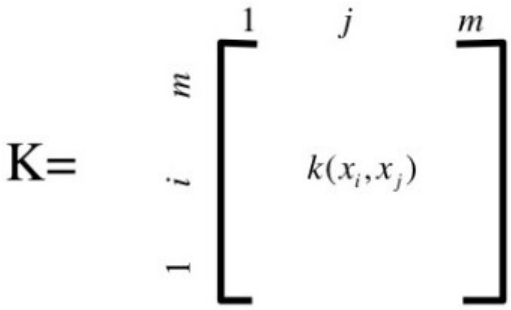
A *Hilbert space* is a complete inner product space. ('Complete' means sequences converge to elements of the space - there aren't any "holes" in the space.)

Examples of Hilbert spaces include:

- The vector space $\mathbf{R}^n$ with $\langle u, v \rangle_{\mathbf{R}^n} = u^T v$, the vector dot product of u and v.
- The space ℓ_2 of square summable sequences, with inner product $\langle u, v \rangle_{\ell_2} = \sum_{i=1}^{\infty} u_i v_i$
- The space $L_2(\mathcal{X}, \mu)$ of square integrable functions, that is, functions f such that $\int f(x)^2 d\mu(x) < \infty$, with inner product $\langle f, g \rangle_{L_2(\mathcal{X},\mu)} = \int f(x) g(x) d\mu(x)$.

Finite States

Say we have a finite input space $\{x_1, ..., x_m\}$. So there's only m possible states for the x_i's. (Think of the bag-of-words example where there are $2^{(\#\text{words})}$ possible states.) I want to be able to take inner products between any two of them using my function k as the inner product. Inner products by definition are symmetric, so $k(x_i, x_j) = k(x_j, x_i)$. In other words, in order for us to even consider k as a valid kernel function, the matrix:



needs to be symmetric, and this means we can diagonalize it, and the eigendecomposition takes this form:

$$\mathbf{K} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'$$

where $\mathbf{V}$ is an orthogonal matrix where the columns of $\mathbf{V}$ are eigenvectors, $\mathbf{v_t}$, and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues λ_t on the diagonal. This fact (that

real symmetric matrices can always be diagonalized) isn't difficult to prove, but requires some linear algebra.

For now, assume all λ_t's are nonnegative, and consider this feature map:

$$\mathbf{\Phi}(x_i) = [\sqrt{\lambda_1}v_1^{(i)}, ..., \sqrt{\lambda_t}v_t^{(i)}, ..., \sqrt{\lambda_m}v_m^{(i)}].$$

(writing it for x_j too):

$$\mathbf{\Phi}(x_j) = [\sqrt{\lambda_1}v_1^{(j)}, ..., \sqrt{\lambda_t}v_t^{(j)}, ..., \sqrt{\lambda_m}v_m^{(j)}].$$

With this choice, k is just a dot product in $\mathbf{R}^m$:

$$\langle \mathbf{\Phi}(x_i), \mathbf{\Phi}(x_j) \rangle_{\mathbf{R}^m} = \sum_{t=1}^{m} \lambda_t v_t^{(i)} v_t^{(j)} = (\mathbf{V}\mathbf{\Lambda}\mathbf{V}')_{ij} = K_{ij} = k(x_i, x_j).$$

Why did we need the λ_t's to be nonnegative? Say $\lambda_s < 0$. Form a point in feature space that is a special linear combination of the $\mathbf{\Phi}(x_i)$'s:

$$\mathbf{z} = \sum_{i=1}^{m} v_s^{(i)} \mathbf{\Phi}(x_i). \quad \text{(coeffs are elements of } \mathbf{v}_s)$$

Then calculate

$$\begin{aligned} \|\mathbf{z}\|_2^2 &= \langle \mathbf{z}, \mathbf{z} \rangle_{\mathbf{R}^m} = \sum_i \sum_j v_s^{(i)} \mathbf{\Phi}(x_i)^T \mathbf{\Phi}(x_j)\ v_s^{(j)} = \sum_i \sum_j v_s^{(i)} K_{ij} v_s^{(j)} \\ &= \mathbf{v}_s^T \mathbf{K} \mathbf{v}_s = \lambda_s < 0 \end{aligned}$$

which conflicts with the geometry of the feature space.

We just showed that if k has any chance of being an inner product in a feature space, then matrix $\mathbf{K}$ needs to be positive semi-definite (needs to have non-negative eigenvalues).

In fact, we'll just define kernels in the first place to be positive semi-definite, since they can't be inner products without that. In the infinite state case, we can't write out a Gram matrix (like $\mathbf{K}$ in the finite state case) for the whole space, because the x's can take infinitely many values. We'll just get to pick m examples - we want to make sure the Gram matrix for those examples is positive semi-definite, no matter what examples we get!

Let us officially define a kernel. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ is a *kernel* if

- k is symmetric: $k(x, y) = k(y, x)$.
- k gives rise to a positive semi-definite "Gram matrix," i.e., for any $m \in \mathbf{N}$ and any $x_1, ..., x_m$ chosen from $\mathcal{X}$, the Gram matrix $\mathbf{K}$ defined by $K_{ij} = k(x_i, x_j)$ is positive semidefinite.

Another way to show that a matrix $\mathbf{K}$ is positive semi-definite is to show that

$$\forall \mathbf{c} \in \mathbf{R}^m, \mathbf{c}^T \mathbf{K} \mathbf{c} \geq 0. \tag{1}$$

(This is equivalent to all the eigenvalues being nonnegative, which again is not hard to show but requires some calculations.)

Here are some nice properties of k:

- $k(u, u) \geq 0$ (Think about the Gram matrix of $m = 1$.)
- $k(u, v) \leq \sqrt{k(u, u)k(v, v)}$ (This is the Cauchy-Schwarz inequality.)

The second property is not hard to show for $m = 2$. The Gram matrix

$$\mathbf{K} = \begin{pmatrix} k(u, u) & k(u, v) \\ k(v, u) & k(v, v) \end{pmatrix}$$

is positive semi-definite whenever $\mathbf{c^T K c} \geq 0 \ \ \forall \mathbf{c}$. Choose in particular

$$\mathbf{c} = \begin{bmatrix} k(v, v) \\ -k(u, v) \end{bmatrix}.$$

Then since $\mathbf{K}$ is positive semi-definite,

$$0 \leq \mathbf{c^T K c} = [k(v, v)k(u, u) - k(u, v)^2]k(v, v)$$

(where I skipped a little bit of simplifying in the equality) so we must then have $k(v, v)k(u, u) \geq k(u, v)^2$. That's it!

Building a Rather Special Hilbert Space

Define $\mathbf{R}^{\mathcal{X}} := \{f : \mathcal{X} \to \mathbf{R}\}$, the space of functions that map $\mathcal{X}$ to $\mathbf{R}$. Let's define the feature map $\mathbf{\Phi} : \mathcal{X} \to \mathbf{R}^{\mathcal{X}}$ so that it maps x to $k(\cdot, x)$:

$$\mathbf{\Phi} : x \longmapsto k(\cdot, x).$$

So, $\mathbf{\Phi}(x)$ is a function, and for a point $z \in \mathcal{X}$, the function assigns $k(z, x)$ to it.

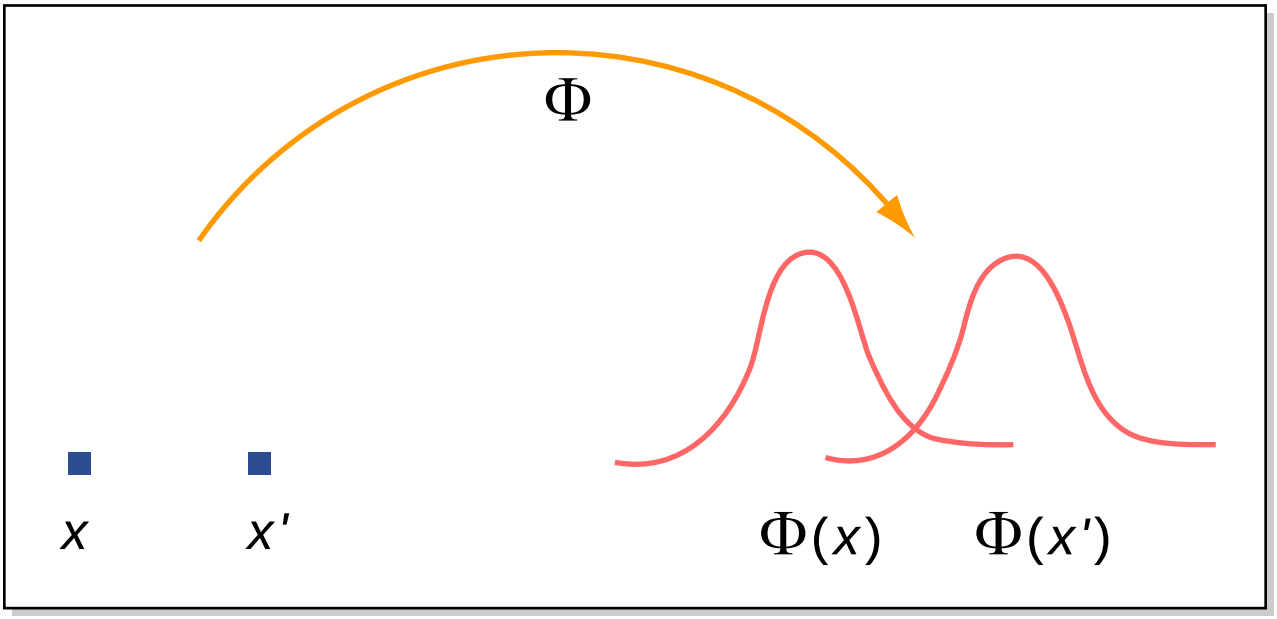


Image by MIT OpenCourseWare.

So we turned each x into a function on the domain $\mathcal{X}$. Those functions could be thought of as infinite dimensional vectors. These functions will be elements of our Hilbert space.

We want to have k be an inner product in feature space. To do this we need to:

1. Define feature map $\mathbf{\Phi} : \mathcal{X} \to \mathbf{R}^{\mathcal{X}}$, which we've done already. Then we need to turn the image of $\mathbf{\Phi}$ into a vector space.
2. Define an inner product $\langle , \rangle_{H_k}$.
3. Show that the inner product satisfies:

$$k(x, z) = \langle \mathbf{\Phi}(x), \mathbf{\Phi}(z) \rangle_{H_k}.$$

That'll make sure that the feature space is a pre-Hilbert space.

Let's do the rest of step 1. Elements in the vector space will look like this, they'll be in the span of the $\mathbf{\Phi}(x_i)$'s:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \leftarrow \text{“vectors”}$$

where m, α_i and $x_1 ... x_m \in \mathcal{X}$ can be anything. (We have addition and multiplication, so it's a vector space.) The vector space is:

$$\text{span}\left(\{\mathbf{\Phi}(x) : x \in \mathcal{X}\}\right) = \left\{ f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) : m \in \mathbf{N}, x_i \in \mathcal{X}, \alpha_i \in \mathbf{R} \right\}.$$

For step 2, let's grab functions $f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$ and $g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j)$, and define the inner product:

$$\langle f, g \rangle_{H_k} = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j).$$

Is it well-defined?

Well, it's symmetric, since k is symmetric:

$$\langle g, f \rangle_{H_k} = \sum_{j=1}^{m'} \sum_{i=1}^{m} \beta_j \alpha_i k(x'_j, x_i) = \langle f, g \rangle_{H_k}.$$

It's also bilinear, look at this:

$$\langle f, g \rangle_{H_k} = \sum_{j=1}^{m'} \beta_j \sum_{i=1}^{m} \alpha_i k(x_i, x'_j) = \sum_{j=1}^{m'} \beta_j f(x'_j)$$

so we have

$$\begin{aligned} \langle f_1 + f_2, g \rangle_{H_k} &= \sum_{j=1}^{m'} \beta_j \left(f_1(x'_j) + f_2(x'_j) \right) \\ &= \sum_{j=1}^{m'} \beta_j f_1(x'_j) + \sum_{j=1}^{m'} \beta_j f_2(x'_j) \\ &= \langle f_1, g \rangle_{H_k} + \langle f_2, g \rangle_{H_k} \text{ (look just above)} \end{aligned}$$

(We can do the same calculation to show $\langle f, g_1 + g_2 \rangle_{H_k} = \langle f, g_1 \rangle_{H_k} + \langle f, g_2 \rangle_{H_k}$.) So it's bilinear.

It's also positive semi-definite, since k gives rise to positive semi-definite Gram matrices. To see this, for function f,

$$\langle f, f \rangle_{H_k} = \sum_{ij=1}^{m} \alpha_i \alpha_j k(x_i, x_j) = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$$

and we said earlier in (1) that since $\mathbf{K}$ is positive semi-definite, it means that for any α_i's we choose, the sum above is ≥ 0.

So, k is almost a inner product! (What's missing?) We'll get to that in a minute.

For step 3, something totally cool happens, namely that the inner product of $k(\cdot, x)$ and f is just the evaluation of f at x.

$$\langle k(\cdot, x), f \rangle_{H_k} = f(x).$$

How did that happen?

And then this is a special case:

$$\langle k(\cdot, x), k(\cdot, x') \rangle_{H_k} = k(x, x').$$

This is why k is called a reproducing kernel, and the Hilbert space is a RKHS.

Oops! We forgot to show that one last thing, which is that $\langle , \rangle_{H_k}$ is strictly positive definite, so that it's an inner product. Let's use the reproducing property. For any x,

$$|f(x)|^2 = |\langle k(\cdot, x), f \rangle_{H_k}|^2 \leq \langle k(\cdot, x), k(\cdot, x) \rangle_{H_k} \cdot \langle f, f \rangle_{H_k} = k(x, x) \langle f, f \rangle_{H_k}$$

which means that $\langle f, f \rangle_{H_k} = 0 \Rightarrow f = 0$ for all x. Ok, it's positive definite, and thus, an inner product.

Now we have an inner product space. In order to make it a Hilbert space, we need to make it complete. The completion is actually not a big deal, we just create a norm:

$$\|f\|_{H_k} = \sqrt{\langle f, f \rangle_{H_k}}$$

and just add to H_k the limit points of sequences that converge in that norm. Once we do that,

$$H_k = \overline{\{f : f = \sum_i \alpha_i k(\cdot, x_i)\}}$$

is a Reproducing Kernel Hilbert space (RKHS).

Here's a formal (simplified) definition of RKHS:

We are given a (compact) $\mathcal{X} \subseteq \mathbf{R}^d$ and a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbf{R}$. We say $\mathcal{H}$ is a *Reproducing Kernel Hilbert Space* if there exists a $k : \mathcal{X} \to \mathbf{R}$, such that

1. k has the reproducing property, i.e., $f(x) = \langle f(\cdot), k(\cdot, x)\rangle_{\mathcal{H}}$
2. k spans $\mathcal{H}$, that is, $\mathcal{H} = \overline{\text{span}\{k(\cdot, x) : x \in \mathcal{X}\}}$

So when we do the kernel trick, we could think that we're implicitly mapping x to $f = \sum_i \alpha_i k(\cdot, x)$ in the RKHS H_k. Neat, huh?

The RKHS we described is the one from the Moore-Aronszajn Theorem (1950) that states that for every positive definite function $k(\cdot, \cdot)$ there exists a unique RKHS.

We could stop there, but we won't. We're going to define *another* Hilbert space that has a one-to-one mapping (an isometric isomorphism) to the first one.

Mercer's Theorem

The inspiration of the name "kernel" comes from the study of integral operators, studied by Hilbert and others. Function k which gives rise to an operator T_k via:

$$(T_k f)(x) = \int_{\mathcal{X}} k(x, x') f(x') dx'$$

is called the *kernel* of T_k.

Think about an operator $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$. If you don't know this notation, don't worry about it. Just think about T_k eating a function and spitting out another one. Think of T_k as an infinite matrix, which maps infinite dimensional vectors to other infinite dimensional vectors. Remember earlier we analyzed the finite state case? It's just like that.[1]

And, just like in the finite state case, because T_k is going to be positive semi-definite, it has an eigen-decomposition into eigenfunctions and eigenvalues that

[1] If you know the notation, you'll see that I'm missing the measure on L_2 in the notation. Feel free to put it back in as you like.

are nonnegative.

The following theorem from functional analysis is very similar to what we proved in the finite state case. This theorem is important - it helps people construct kernels (though I admit myself I find the other representation more helpful).

Basically, the theorem says that if we have a kernel k that is positive (defined somehow), we can expand k in terms of eigenfunctions and eigenvalues of a positive operator that comes from k.

Mercer's Theorem *(Simplified) Let $\mathcal{X}$ be a compact subset of $\mathbf{R}^n$. Suppose k is a continuous symmetric function such that the integral operator $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ defined by*

$$(T_k f)(\cdot) = \int_{\mathcal{X}} k(\cdot, \mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

is positive; which here means $\forall f \in L_2(\mathcal{X})$,

$$\int_{\mathcal{X}\times\mathcal{X}} k(\mathbf{x}, \mathbf{z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geq 0,$$

then we can expand $k(\mathbf{x}, \mathbf{z})$ in a uniformly convergent series in terms of T_k's eigenfunctions $\psi_j \in L_2(\mathcal{X})$, normalized so that $\|\psi\|_{L_2} = 1$, and positive associated eigenvalues $\lambda_j \geq 0$,

$$k(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty} \lambda_j \psi_j(\mathbf{x}) \psi_j(\mathbf{z}).$$

The definition of positive semi-definite here is equivalent to the ones we gave earlier.[2]

So this looks just like what we did in the finite case. So we could define a feature map as in the finite case this way:

$$\mathbf{\Phi}(\mathbf{x}) = [\sqrt{\lambda_1}\psi_1(\mathbf{x}), ..., \sqrt{\lambda_j}\psi_j(\mathbf{x}), ...].$$

[2]To see the equivalence, for this direction $\Leftarrow$, choose f as a weighted sum of δ functions at each x. For this direction $\Rightarrow$, say that $\int_{\mathcal{X}\times\mathcal{X}} k(x,z)f(x)f(z)dxdz \geq 0$ doesn't hold for some f and show the contrapositive. Approximate the integral with a finite sum over a mesh of inputs $\{x_1, ..., x_m\}$ chosen sufficiently finely, then let $\mathbf{v}$ be the values of f on the mesh, and as long as the mesh is fine enough, we'll have $\mathbf{v}'\mathbf{K}\mathbf{v} < 0$, so $\mathbf{K}$ isn't positive semi-definite.

So that's a cool property of the RKHS that we can define a feature space according to Mercer's Theorem. Now for another cool property of SVM problems, having to do with RKHS.

Representer Theorem

Recall that the SVM optimization problem can be expressed as follows:

$$f^* = \text{argmin}_{f \in H_k} R^{\text{train}}(f)$$

where

$$R^{\text{train}}(f) := \sum_{i=1}^{m} \text{hingeloss}(f(x_i), y_i) + C\|f\|_{H_k}^2.$$

Or, you could even think about using a generic loss function:

$$R^{\text{train}}(f) := \sum_{i=1}^{m} \ell(f(x_i), y_i) + C\|f\|_{H_k}^2.$$

The following theorem is kind of a big surprise. It says that the solutions to any problem of this type - no matter what the loss function is! - all have a solution in a rather nice form.

Representer Theorem *(Kimeldorf and Wahba, 1971, Simplified) Fix a set $\mathcal{X}$, a kernel k, and let H_k be the corresponding RKHS. For any function $\ell : \mathbf{R}^2 \to \mathbf{R}$, the solutions of the optimization problem:*

$$f^* \in \text{argmin}_{f \in H_k} \sum_{i=1}^{m} \ell(f(x_i), y_i) + \|f\|_{H_k}^2$$

can all be expressed in the form:

$$f^* = \sum_{i=1}^{m} \alpha_i k(x_i, \cdot).$$

This shows that to solve the SVM optimization problem, we only need to solve for the α_i, which agrees with the solution from the Lagrangian formulation for SVM. It says that even if we're trying to solve an optimization problem in an infinite dimensional space H_k containing linear combinations of kernels centered

on *arbitrary* $x_i's$, then the solution lies in the span of the m kernels centered on the x_i's.

I hope you grasp how cool this is!

Proof. Suppose we project f onto the subspace:

$$\text{span}\{k(x_i, \cdot) : 1 \leq i \leq m\}$$

obtaining f_s (the component along the subspace) and $f_\perp$ (the component perpendicular to the subspace). We have:

$$f = f_s + f_\perp \Rightarrow \|f\|_{H_k}^2 = \|f_s\|_{H_k}^2 + \|f_\perp\|_{H_k}^2 \geq \|f_s\|_{H_k}^2.$$

This implies that $\|f\|_{H_k}$ is minimized if f lies in the subspace. Furthermore, since the kernel k has the reproducing property, we have for each i:

$$f(x_i) = \langle f, k(x_i, \cdot)\rangle_{H_k} = \langle f_s, k(x_i, \cdot)\rangle_{H_k} + \langle f_\perp, k(x_i, \cdot)\rangle_{H_k} = \langle f_s, k(x_i, \cdot)\rangle_{H_k} = f_s(x_i).$$

So,

$$\sum_{i=1}^{m} \ell(f(x_i), y_i) = \sum_{i=1}^{m} \ell(f_s(x_i), y_i).$$

In other words, the loss depends only on the component of f lying in the subspace. To minimize, we can just let $\|f_\perp\|_{H_k}$ be 0 and we can express the minimizer as:

$$f^*(\cdot) = \sum_{i=1}^{m} \alpha_i k(x_i, \cdot).$$

That's it! ■

Draw some bumps on x_i's

Constructing Kernels

Let's construct new kernels from previously defined kernels. Suppose we have k_1 and k_2. Then the following are also valid kernels:

1. $k(x, z) = \alpha k_1(x, z) + \beta k_2(x, z)$ for $\alpha, \beta \geq 0$

Proof. k_1 has its feature map $\mathbf{\Phi}_1$ and inner product $\langle\rangle_{H_{k_1}}$ and k_2 has its feature map $\mathbf{\Phi}_2$ and inner product $\langle\rangle_{H_{k_2}}$. By linearity, we can have:

$$\alpha k_1(x,z) = \langle\sqrt{\alpha}\mathbf{\Phi}_1(x), \sqrt{\alpha}\mathbf{\Phi}_1(z)\rangle_{H_{k_1}} \text{ and } \beta k_2(x,z) = \langle\sqrt{\beta}\mathbf{\Phi}_2(x), \sqrt{\beta}\mathbf{\Phi}_2(z)\rangle_{H_{k_2}}$$

Then:

$$\begin{aligned} k(x,z) &= \alpha k_1(x,z) + \beta k_2(x,z) \\ &= \langle\sqrt{\alpha}\mathbf{\Phi}_1(x), \sqrt{\alpha}\mathbf{\Phi}_1(z)\rangle_{H_{k_1}} + \langle\sqrt{\beta}\mathbf{\Phi}_2(x), \sqrt{\beta}\mathbf{\Phi}_1(z)\rangle_{H_{k_2}} \\ &=: \langle[\sqrt{\alpha}\mathbf{\Phi}_1(x), \sqrt{\beta}\mathbf{\Phi}_2(x)], [\sqrt{\alpha}\mathbf{\Phi}_1(z), \sqrt{\beta}\mathbf{\Phi}_2(z)]\rangle_{H_{\text{new}}} \end{aligned}$$

and that means that $k(x,z)$ can be expressed as an inner product.

2. $k(x,z) = k_1(x,z)k_2(x,z)$
   Proof omitted - it's a little fiddly.

3. $k(x,z) = k_1(h(x), h(z))$, where $h : \mathcal{X} \to \mathcal{X}$.

   Proof. Since h is a transformation in the same domain, k is simply a different kernel in that domain:

   $$k(x,z) = k_1(h(x), h(z)) = \langle\mathbf{\Phi}(h(x)), \mathbf{\Phi}(h(z))\rangle_{H_{k_1}} =: \langle\mathbf{\Phi}_h(x), \mathbf{\Phi}_h(z)\rangle_{H_{\text{new}}}$$

4. $k(x,z) = g(x)g(z)$ for $g : \mathcal{X} \to \mathbf{R}$.
   Proof omitted, again this one is a little tricky.

5. $k(x,z) = h(k_1(x,z))$ where h is a polynomial with positive coefficients
   Proof. Since each polynomial term is a product of kernels with a positive coefficient, the proof follows by applying 1 and 2.

6. $k(x,z) = \exp(k_1(x,z))$
   Proof Since:
   $$\exp(x) = \lim_{i\to\infty}\left(1 + x + \cdots + \frac{x^i}{i!}\right)$$
   the proof basically follows from 5.

7. $k(\mathbf{x},\mathbf{z}) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{z}\|_{\ell_2}^2}{\sigma^2}\right)$ "Gaussian kernel"

Proof.

$$k(\mathbf{x},\mathbf{z}) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{z}\|_{\ell_2}^2}{\sigma^2}\right) = \exp\left(\frac{-\|\mathbf{x}\|_{\ell_2}^2 - \|\mathbf{z}\|_{\ell_2}^2 + 2\mathbf{x}^T\mathbf{z}}{\sigma^2}\right)$$
$$= \left(\exp\left(\frac{-\|\mathbf{x}\|_{\ell_2}}{\sigma^2}\right)\exp\left(\frac{-\|\mathbf{z}\|_{\ell_2}}{\sigma^2}\right)\right)\exp\left(\frac{2\mathbf{x}^T\mathbf{z}}{\sigma^2}\right)$$

$g(\mathbf{x})g(\mathbf{z})$ is a kernel according to 4, and $\exp(k_1(\mathbf{x},\mathbf{z}))$ is a kernel according to 6. According to 2, the product of two kernels is a valid kernel.

Note that the Gaussian kernel is translation invariant, since it only depends on $\mathbf{x}-\mathbf{z}$.

This picture helps with the intuition about Gaussian kernels if you think of the first representation of the RKHS I showed you:

$$\mathbf{\Phi}(\mathbf{x}) = k(\mathbf{x},\cdot) = \exp\left(\frac{-\|\mathbf{x}-\cdot\|_{\ell_2}^2}{\sigma^2}\right)$$

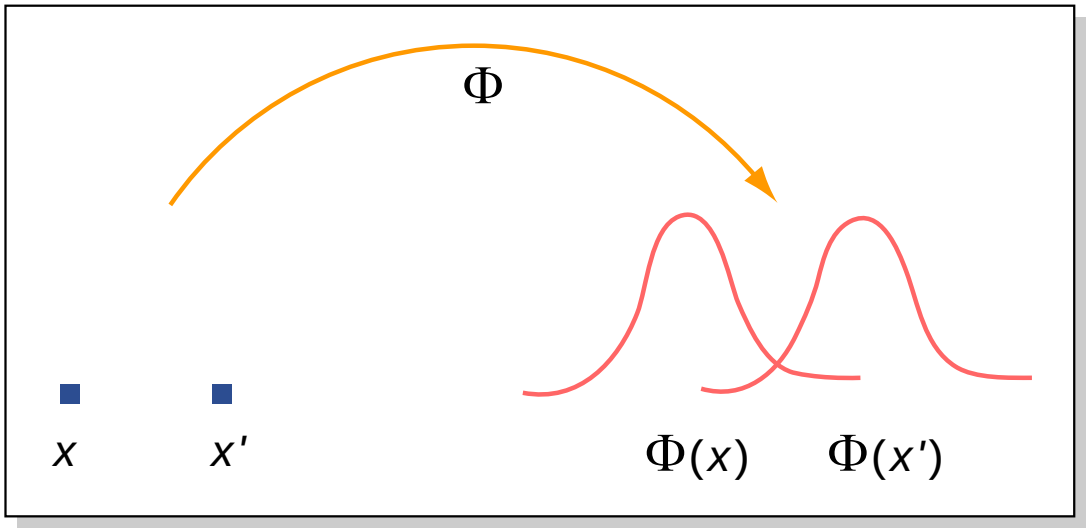
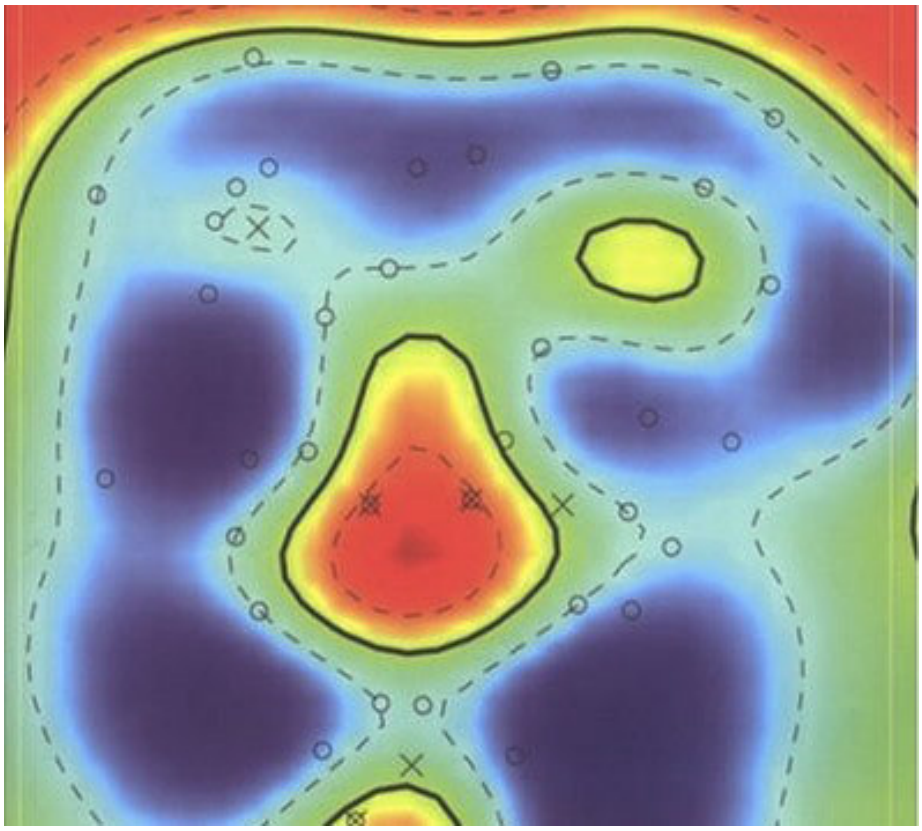


Image by MIT OpenCourseWare.

Gaussian kernels can be very powerful:



Bernhard Schölkopf, Christopher J.C. Burges, and Alexander J. Smola, ADVANCES IN KERNEL METHODS: SUPPORT VECTOR LEARNING, published by the MIT Press. Used with permission.

Don't make the σ^2 too small or you'll overfit!

15.097 Prediction: Machine Learning and Statistics
Spring 2012