Kernels for dummies

1 Determining whether two distributions are different

Suppose we have samples from two distributions, $p(x)$ and $q(x)$. The simplest estimate for $p(x)$ are $q(x)$ is one of the form

$$\hat{p}(x) = \frac{1}{n_p}\sum_{i=1}^{n_p}\delta(x - x_i^p) \tag{1a}$$ {pq}

$$\hat{q}(x) = \frac{1}{n_q}\sum_{i=1}^{n_q}\delta(x - x_i^q) \tag{1b}$$

where x_i^p andx_i^q are samples from $p(x)$ and $q(x)$, respectively.

Based on the samples, we want to know if the true distributions, $p(x)$ and $q(x)$, are different. To determine that we define a difference via a witness function, $f(x)$, as

$$\Delta_f \equiv \int dx f(x)\delta\hat{p}(x)\,. \tag{2}$$ {deltaf}

where

$$\delta\hat{p}(x) \equiv \hat{p}(x) - \hat{q}(x)\,. \tag{3}$$

We want to find the $f(x)$ that maximizes Δ_f. We can't, of course, let $f(x)$ be arbitrary, because then we could make Δ_f infinite. So we put a kernel regularizer on it: we maximize Δ_f^2 with respect to $f(x)$ subject to the constraint that

$$\int dxdy\, f(x)K^{-1}(x,y)f(y) = 1\,. \tag{4}$$ {deltap}

Here $K^{-1}(x,y)$ is defined via

$$\int dy\, K^{-1}(x,y)K(y,z) = \delta(x-z) \tag{5}$$ {kinv_def}

where $\delta(\cdot)$ is the Dirac delta function. For why this is a regularizer, see the last section, Sec. 4.

This gives us a standard Lagrange multiplier problem. Before writing it down, though, to emphasize the connection to linear algebra, we define a giant dot product, {giant_dot}

$$\mathbf{f}\bullet\mathbf{g} \equiv \int dx f(x)g(x) \tag{6a}$$

$$\mathbf{f}\bullet\mathbf{D}(y) \equiv \int dx f(x)D(x,y)\,. \tag{6b}$$

Here bold indicates generalized (meaning uncountably infinite dimensional) vectors or matrices: the y^{th} component of $\mathbf{f}$ is $f(y)$ and the xy^{th} component of $\mathbf{D}$ is $D(x,y)$. Note that

when taking the giant dot product between a vector and a matrix, we will rarely need to specify the dependence, as we did in Eq. (6b). Typically we'll just write $\mathbf{f} \bullet \mathbf{D}$, a quantity whose y^{th} component is $\int dx f(x) D(x,y)$.

With this notation, our problem can be written

$$\mathbf{f}^* = \arg\max_{\mathbf{f}} \left\{ [\mathbf{f} \bullet (\hat{\mathbf{p}} - \hat{\mathbf{q}})]^2 + \lambda [\mathbf{f} \bullet \mathbf{K}^{-1} \bullet \mathbf{f} - 1] \right\} \tag{7}$$

where λ is a Lagrange multiplier. Minimizing with respect to $\mathbf{f}$, and enforcing the constraint $\mathbf{f} \bullet \mathbf{K}^{-1} \bullet \mathbf{f} = 1$, leads to

$$\mathbf{f}^* = \frac{\mathbf{K} \bullet \delta\hat{\mathbf{p}}}{[\delta\hat{\mathbf{p}} \bullet \mathbf{K} \bullet \delta\hat{\mathbf{p}}]^{1/2}} \,. \tag{8}$$

Inserting this into Eq. (2), we find that

$$\Delta^2_{f^*} = \delta\hat{\mathbf{p}} \bullet \mathbf{K} \bullet \delta\hat{\mathbf{p}} = \int dx \, \delta\hat{p}(x) K(x,y) \delta\hat{p}(y) \,. \tag{9}$$

Or, using Eq. (4) for $\delta\hat{p}(x)$ and Eq. (1a) for $p(x)$ and $q(x)$, we arrive at our final expression,

$$\Delta^2_{f^*} = \frac{1}{n_p^2} \sum_{i,j=1}^{n_p} K(x_i^p, x_j^p) - \frac{2}{n_p n_q} \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} K(x_i^p, x_j^q) + \frac{1}{n_q^2} \sum_{i,j=1}^{n_q} K(x_i^q, x_j^q) \,. \tag{10}$$

Computing $\Delta^2_{f^*}$ from data is the easy part; the hard part is determining whether or not it's statistically significantly different from zero. But we won't do that here.

2 Kernel ridge regression

Suppose we want to minimize the following least square distance,

$$\Delta^2 = \sum_i (\mathbf{f} \bullet \hat{\mathbf{p}}_i - y_i)^2 \tag{11}$$

with respect to $f(x)$. Here $\hat{p}_i(x)$ is a sample distribution,

$$\hat{p}_i(x) = \frac{1}{n_i} \sum_k \delta(x - x_k^i) \tag{12} \quad \{\texttt{pihat}\}$$

where x_k^i is the k^{th} sample from the true distribution, $p_i(x)$, and n_i is the number of samples. We use ridge regression, which means

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} \left[\Delta^2 + \lambda \mathbf{f} \bullet \mathbf{K}^{-1} \bullet \mathbf{f} \right] . \tag{13} \quad \{\texttt{fstar\_regr}$$

Note that this is equivalent to computing a map estimate with a log likelihood of $-\Delta^2/2$ and a Gaussian process prior.

Differentiating the right hand side of Eq. (13) gives us

$$\mathbf{f}^* = \frac{1}{\lambda}\sum_i (y_i - \mathbf{f}^* \bullet \hat{\mathbf{p}}_i)\mathbf{K} \bullet \hat{\mathbf{p}}_i \tag{14}$$

Consequently,

$$\mathbf{f}^* = \sum_i \alpha_i \mathbf{K} \bullet \hat{\mathbf{p}}_i \tag{15}$$ {fstar_kerne

where

$$\alpha_i = \frac{y_i - \mathbf{f}^* \bullet \hat{\mathbf{p}}_i}{\lambda}\,. \tag{16}$$ {alpha}

Inserting Eq. (15) into (16), and assuming a symmetric kernel, we have

$$\lambda\alpha_i = y_i - \sum_j \hat{\mathbf{p}}_i \bullet \mathbf{K} \bullet \hat{\mathbf{p}}_j + y_i\,. \tag{17}$$ {eig}

It is convenient to define the matrix

$$B_{ij} \equiv \hat{\mathbf{p}}_i \bullet \mathbf{K} \bullet \hat{\mathbf{p}}_j = \int dxdy\, \hat{p}_i(x) K(x,y) \hat{p}_j(y)\,. \tag{18}$$ {adef}

Note that B_{ij} is easy to compute: inserting Eq. (12) into (18) gives us

$$B_{ij} = \frac{1}{n_i n_j}\sum_{kl} K(x_k^i, x_l^j)\,. \tag{19}$$

In any case, inserting Eq. (18) into (17) and using vector notation, we have

$$\boldsymbol{\alpha} = (\mathbf{B} + \lambda\mathbf{I})^{-1} \cdot \mathbf{y} \tag{20}$$

where $\mathbf{B}$ and $\mathbf{I}$ are plain old matrices, and $\mathbf{I}$ is the identity matrix. This is standard ridge regression.

3 Kernel PCA

Suppose we have a set of points, x_i, $i = 1, ..., n$ (the x_i could be vectors, but that won't affect anything we do), and we want to project them into a higher dimensional space and do PCA in that space. Since we're going to a higher dimensional space, might as well go all the way to an uncountably infinite dimensional space, and map the x_i to functions. We thus define

$$f_i(y) = K(x_i, y)\,. \tag{21}$$ {fdef}

We now want to do PCA in function space. If this were standard old PCA, we would minimize the cost function

$$\Delta = \sum_{i=1}^n \int dy \left(f_i(y) - \sum_k A_{ik} v_k(y)\right)^2 \tag{22}$$ {delta}

with respect to the A_{ik} and $v_k(y)$. This, however, corresponds to minimizing the L2 norm. But there are other choices of norm. Here we consider one class of norms, which is to insert $Q^{-1}(x,y)$ into the square in Eq. (22). This leads to the cost function

$$\Delta_Q = \sum_{i=1}^n \int dy\, dz \left(f_i(y) - \sum_k A_{ik} v_k(y)\right) Q^{-1}(y,z) \left(f_i(z) - \sum_l A_{il} v_l(z)\right). \tag{23}$$

If $Q(y,z)$ were a standard kernel-like object, say $Q(y,z) = \exp(-(y-z)^2)$, then the effect of adding $Q^{-1}(y,z)$ would be to emphasize smoothness. Without loss of generality we may assume that $Q(x,y)$ is symmetric: $Q(x,y) = Q(y,x)$.

To emphasize the relationship to linear algebra, we replace the integrals with giant dot products,

$$\Delta_Q = \sum_{i=1}^n \left(\mathbf{f}_i - \sum_k A_{ik} \mathbf{v}_k\right) \bullet \mathbf{Q}^{-1} \bullet \left(\mathbf{f}_i - \sum_l A_{il} \mathbf{v}_l\right). \tag{24}$$ {delta_q}

Finding the A_{ik} and $v_k(y)$ that minimize Δ_Q is reasonably straightforward (algebra below), leading to

$$\mathbf{v} = \sum_i \alpha_i \mathbf{f} \tag{25}$$

where the α_i obey the eigenvalue equation

$$\sum_j C_{ij} \alpha_j = \lambda_k \alpha_i \tag{26}$$ {eigen_c}

with

$$C_{ij} = \mathbf{f}_i \bullet \mathbf{Q}^{-1} \bullet \mathbf{f}_j. \tag{27}$$ {cov_c}

Using Eq. (21), C_{ij} becomes (assuming a symmetric kernel, K),

$$C_{ij} = \int dydz\, K(x_i,y) Q^{-1}(y,z) K(z,x_j). \tag{28}$$

An especially convenient choice for Q is $Q = K$, in which case

$$C_{ij} = K(x_i,x_j). \tag{29}$$

3.1 Algebra for kernel PCA

Probably all this is standard, but just in case, here we derive the above equations. Our starting point is to make Eq. (24) look more like standard PCA. To do that, we make the definitions {transform}

$$\mathbf{g}_i \equiv \mathbf{f}_i \bullet \mathbf{Q}^{-1/2} \tag{30a}$$

$$\mathbf{u}_k \equiv \mathbf{v}_k \bullet \mathbf{Q}^{-1/2}, \tag{30b}$$

So that

$$\Delta_Q = \sum_{i=1}^{n} \left(\mathbf{g}_i - \sum_k A_{ik} \mathbf{u}_k \right) \bullet \left(\mathbf{g}_i - \sum_l A_{il} \mathbf{u}_l \right) . \tag{31}$$ {pca_standa

We want to minimize this expression with respect to A_{ik} and u_k. Setting $d\Delta_q/du_k$ and $d\Delta_Q/dA_{ik}$ to zero yields

$$\sum_i A^T_{ki} \mathbf{g}_i = \sum_i A^T_{ki} \sum_l A_{il} \mathbf{u}_l \tag{32a}$$ {du}

$$\mathbf{g}_i \bullet \mathbf{u}_k = \sum_l A_{il} \mathbf{u}_l \bullet \mathbf{u}_k \tag{32b}$$ {dA}

where T denotes transpose. Solving Eq. (32b) for A^T_{ki} gives us

$$A^T_{ki} = \sum_l (\mathbf{u}_k \bullet \mathbf{u}_l)^{-1} \mathbf{u}_l \bullet \mathbf{g}_i \tag{33}$$

where $\mathbf{u}_k \bullet \mathbf{u}_l$ is treated as a matrix with components k and l. Inserting this into Eq. (32a), and applying a small amount of algebra, we have

$$\sum_l (\mathbf{u}_k \bullet \mathbf{u}_l)^{-1} \mathbf{u}_l \bullet \sum_i \mathbf{g}_i \mathbf{g}_i = \sum_l (\mathbf{u}_k \bullet \mathbf{u}_l)^{-1} \mathbf{u}_l \bullet \sum_i \mathbf{g}_i \mathbf{g}_i \bullet \sum_m \mathbf{u}_m (\mathbf{u}_m \bullet \mathbf{u}_l)^{-1} \mathbf{u}_l . \tag{34}$$

Getting rid of the inverse on both sides leads to

$$\sum_i \mathbf{g}_i \mathbf{g}_i \bullet \mathbf{u}_k = \mathbf{u}_k \bullet \sum_i \mathbf{g}_i \mathbf{g}_i \bullet \sum_m \mathbf{u}_m (\mathbf{u}_m \bullet \mathbf{u}_l)^{-1} \mathbf{u}_l . \tag{35}$$

As is easy to verify, this equation is satisfied if the $\mathbf{u}_k$ are eigenvectors of $\sum_i \mathbf{g}_i \mathbf{g}_i$. That is, the $\mathbf{u}_k$ obey

$$\sum_i \mathbf{g}_i \mathbf{g}_i \bullet \mathbf{u}_k = \lambda_k \mathbf{u}_k \tag{36}$$ {eigen_u}

It's not totally clear that this is the global minimum, but we'll assume it is. Once the $\mathbf{u}_k$ are known, it's easy to show that the A_{ik} are given by

$$A_{ik} = \mathbf{g}_i \bullet \mathbf{u}_k . \tag{37}$$

The next step is to find $\mathbf{v}_k$ in terms of $\mathbf{u}_k$. Inserting Eq. (30) into (36), we have

$$\mathbf{R} \bullet \mathbf{Q}^{-1} \bullet \mathbf{v}_k = \lambda_k \mathbf{v}_k \tag{38a}$$ {eigen_a}

$$A_{ik} = f_i \bullet \mathbf{v}_k \tag{38b}$$ {eigen_b}

where

$$R(x,y) \equiv \sum_{i=1}^{n} f_i(x) f_i(y) . \tag{39}$$ {rdef}

Because there are a finite number of samples (n) and the $\mathbf{g}_i$ are infinite dimensional (they're functions), the eigenvectors, $\mathbf{u}_k$, are linear combinations of the $\mathbf{g}_i$,

$$\mathbf{u}_k = \sum_i \alpha_i \mathbf{g}_i \,. \tag{40}$$

This implies, using Eq. (30b), that $\mathbf{v}_k$ is given by

$$\mathbf{v}_k = \sum_i \alpha_i \mathbf{f}_i \,. \tag{41}$$

Inserting this into Eq. (38a), we have

$$\mathbf{R} \bullet \mathbf{Q}^{-1} \bullet \sum_i \alpha_i \mathbf{f}_i = \lambda_k \sum_i \alpha_i \mathbf{f}_i \,. \tag{42}$$

Using Eq. (39) for R, this becomes

$$\sum_i \mathbf{f}_i \mathbf{f}_i \bullet \mathbf{Q}^{-1} \bullet \sum_j \mathbf{f}_j \alpha_j = \lambda_k \sum_i \mathbf{f}_i \alpha_i \,. \tag{43}$$

Assuming the $\mathbf{f}_i$ are linearly independent, we can drop the sum over i. This produces Eq. (26), with the covariance matrix, C_{ij}, given by Eq. (27).

4 Why $\mathbf{K}^{-1}$ is a regularizer

{kinv}

To see why $\mathbf{K}^{-1}$ is a regularizer, we'll restrict ourselves to translation invariant kernels: $K(x,y) = K(x-y)$. We start by defining

$$D(\mathbf{f},\mathbf{g}) \equiv \int dxdy\, f(x) K^{-1}(x-y) g(y) \,. \tag{44}$$ {D}

Then we'll Fourier transform. We'll use tildes to denote a Fourier transform; for example,

$$\tilde{f}(k) \equiv \int dx\, e^{ikx} f(x) \,, \tag{45}$$ {ftinv}

which implies that

$$f(x) = \int \frac{dk}{2\pi}\, e^{-ikx} \tilde{f}(k) \,. \tag{46}$$ {ft}

Using Eq. (46), Eq. (44) becomes, after rearranging terms slightly,

$$D(\mathbf{f},\mathbf{g}) = \int \frac{dk_x dk_y}{(2\pi)^2}\, \tilde{f}(k_x)\tilde{g}(k_y) \int dxdy\, e^{-ik_x x} K^{-1}(x-y) e^{-ik_y y} \,. \tag{47}$$

Letting $e^{-ik_x x} = e^{-ik_x(x-y)} e^{-ik_x y}$, making the change of variables $z = x - y$, and performing the integral over z gives us

$$D(\mathbf{f},\mathbf{g}) = \int dk_x dk_y\, \tilde{f}(k_x)\tilde{g}(k_y) \int dy\, \tilde{K}^{-1}(k_x) e^{-i(k_y+k_x)y} \,. \tag{48}$$

The integral over y is $2\pi\delta(k_x + k_y)$. Assuming f and g are real, so that $\tilde{g}(-k) = \tilde{g}^*(k)$ where * denotes complex conjugate, we have

$$D(\mathbf{f}, \mathbf{g}) = \int dk\, \tilde{f}(k)\tilde{g}^*(k)\tilde{K}^{-1}(k)\,. \tag{49}$$ {Dfg}

So what's $\tilde{K}^{-1}$? To answer that, we use Eq. (5) to write

$$1 = \int dx\, e^{ik(x-z)} \int dy\, K(x-y)K^{-1}(y-z)\,. \tag{50}$$

Making the change of variable $x = u + y$, and then $y = u + w$, this becomes

$$1 = \int du\, e^{iku} K(u) \int dw\, e^{ik(w)} K^{-1}(w)\,. \tag{51}$$

The first integral is $\tilde{K}(k)$; the second is $\tilde{K}^{-1}(k)$. Thus,

$$\tilde{K}^{-1}(k) = \frac{1}{\tilde{K}(k)}\,. \tag{52}$$

Inserting this into Eq. (49) gives us

$$D(\mathbf{f}, \mathbf{g}) = \int dk\, \frac{\tilde{f}(k)\tilde{g}^*(k)}{\tilde{K}(k)}\,. \tag{53}$$

For smooth kernels, $\tilde{K}(k)$ falls off rapidly with k; for instance, if $K(x-y) = e^{-(x-y)^2/2\sigma^2}$, then $\tilde{K}(k) \propto e^{-\sigma^2 k^2/2}$. Thus, $D(\mathbf{f}, \mathbf{g})$ is large for functions with large Fourier components – that is, it's large for non-smooth functions.