CS 536 :What is the Gaussian Kernel even doing?

16:198:536

SVMs are frequently used in an 'off-the-shelf' way - take your data, pick your favorite kernel, throw the nearest SVM solver at the problem, and see what happens. A common choice for the kernel, before anything else is known, is the Gaussian Kernel

$$K(\underline{x},\underline{y}) = \exp\left(-\frac{1}{\sigma^2}||\underline{x}-\underline{y}||^2\right). \tag{1}$$

But what is this kernel doing? What does the 'Gaussian Kernel SVM' mean? What is the importance of σ^2?

To illustrate this by example, consider the usual XOR data:

$$\{A=((-1,+1),+1), B=((-1,-1),-1), C=((+1,-1),+1), D=((+1,+1),-1)\}.$$

Writing out the dual SVM in this case, we have

$$\begin{aligned}\max_{\alpha_A,\alpha_B,\alpha_C,\alpha_D}\ & [\alpha_A+\alpha_B+\alpha_C+\alpha_D] - \frac{1}{2}\left[\alpha_A^2K(A,A)+\alpha_B^2K(B,B)+\alpha_C^2K(C,C)+\alpha_D^2K(D,D)\right]\\ & -[-\alpha_AK(A,B)\alpha_B+\alpha_AK(A,C)\alpha_C-\alpha_AK(A,D)\alpha_D]\\ & -[-\alpha_BK(B,C)\alpha_C+\alpha_BK(B,D)\alpha_D-\alpha_CK(C,D)\alpha_D]\\ \text{(s.t.)}\ & \alpha_A-\alpha_B+\alpha_C-\alpha_D=0\\ & \alpha_A,\alpha_B,\alpha_C,\alpha_D\geq 0.\end{aligned} \tag{2}$$

To analyze the solution to the Gaussian Kernel SVM and the effect of σ^2, we need an explicit, general solution, so it's worth taking the time to solve out the above explicitly.

To simplify this, note that regardless of σ^2, we have that $K(A,A)=K(B,B)=K(C,C)=K(D,D)=1$. Additionally, $K(A,B)=\exp(-(1/\sigma^2)2^2)=\exp(-4/\sigma^2)$. Similarly, $K(A,B)=K(B,C)=K(C,D)=K(D,A)=\exp(-4/\sigma^2)$. Across the square, we have $K(A,C)=K(B,D)=\exp(-8/\sigma^2)$. Denote $\rho=\exp(-4/\sigma^2)$ for notational convenience. We wish to solve

$$\begin{aligned}\max_{\alpha_A,\alpha_B,\alpha_C,\alpha_D}\ & [\alpha_A+\alpha_B+\alpha_C+\alpha_D] - \frac{1}{2}\left[\alpha_A^2+\alpha_B^2+\alpha_C^2+\alpha_D^2\right]\\ & -\left[-\left(\alpha_A\alpha_B\right)\rho+\left(\alpha_A\alpha_C\right)\rho^2-\left(\alpha_A\alpha_D\right)\rho\right]\\ & -\left[-\left(\alpha_B\alpha_C\right)\rho+\left(\alpha_B\alpha_D\right)\rho^2-\left(\alpha_C\alpha_D\right)\rho\right]\\ \text{(s.t.)}\ & \alpha_A-\alpha_B+\alpha_C-\alpha_D=0\\ & \alpha_A,\alpha_B,\alpha_C,\alpha_D\geq 0.\end{aligned} \tag{3}$$

This simplifies further still:

$$\begin{aligned}\max_{\alpha_A,\alpha_B,\alpha_C,\alpha_D}\ & [\alpha_A+\alpha_B+\alpha_C+\alpha_D] - \frac{1}{2}\left[\alpha_A^2+\alpha_B^2+\alpha_C^2+\alpha_D^2\right] + \left[\alpha_A+\alpha_C\right]\left[\alpha_B+\alpha_D\right]\rho - \left[\alpha_A\alpha_C+\alpha_B\alpha_D\right]\rho^2\\ \text{(s.t.)}\ & \alpha_A+\alpha_C=\alpha_B+\alpha_D\\ & \alpha_A,\alpha_B,\alpha_C,\alpha_D\geq 0.\end{aligned} \tag{4}$$

Noting that $\alpha_A+\alpha_C=\alpha_B+\alpha_D$, it's convenient to introduce a new variable ϵ to represent this sum, so that $\alpha_C=\epsilon-\alpha_A$, $\alpha_D=\epsilon-\alpha_B$:

$$\begin{aligned}\max_{\alpha_A,\alpha_B,\epsilon}\ & [2\epsilon] - \frac{1}{2}\left[\alpha_A^2+\alpha_B^2+(\epsilon-\alpha_A)^2+(\epsilon-\alpha_B)^2\right] + \epsilon^2\rho - \left[\alpha_A(\epsilon-\alpha_A)+\alpha_B(\epsilon-\alpha_B)\right]\rho^2\\ \text{(s.t.)}\ & \alpha_A\leq\epsilon, \alpha_B\leq\epsilon\\ & \alpha_A,\alpha_B,\epsilon\geq 0,\end{aligned} \tag{5}$$

or (grouping like terms)

$$\begin{aligned}\max_{\alpha_A,\alpha_B,\epsilon}\ & \left[2\epsilon+\epsilon^2+\epsilon^2\rho\right]-\left(\left[\alpha_A^2-\epsilon\alpha_A\right]+\left[\alpha_A\epsilon-\alpha_A^2\right]\rho^2\right)-\left(\left[\alpha_B^2-\epsilon\alpha_B\right]+\left[\alpha_B\epsilon-\alpha_B^2\right]\rho^2\right)\\ \text{(s.t.)}\ & \alpha_A\leq\epsilon,\alpha_B\leq\epsilon\\ & \alpha_A,\alpha_B,\epsilon\geq 0.\end{aligned}\tag{6}$$

One observation to make here is that the above is completely separable and symmetric in terms of α_A and α_B - we can imagine trying to maximize with respect to these for any given value of ϵ, but because the objective function is separable and symmetric, for any given value of ϵ we'll get the same solution for both, $\alpha_A=\alpha_B=\alpha$ for some α that depends on ϵ. This simplifies the problem even further:

$$\begin{aligned}\max_{\alpha,\epsilon}\ & \left[2\epsilon+\epsilon^2+\epsilon^2\rho\right]-2\left(\left[\alpha^2-\epsilon\alpha\right]+\left[\alpha\epsilon-\alpha^2\right]\rho^2\right)\\ \text{(s.t.)}\ & 0\leq\alpha\leq\epsilon.\end{aligned}\tag{7}$$

The (unconstrained) maximum of the above occurs when $\alpha=\epsilon/2$ (why?) so since this is feasible for the constraints, this is the solution α for any given value of ϵ. Since $\alpha_A=\alpha_B=\alpha=\epsilon/2$, and $\alpha_C=\epsilon-\alpha_A,\alpha_D=\epsilon-\alpha_B$, we get that $\alpha_A=\alpha_B=\alpha_C=\alpha_D=\epsilon/2$. But what is the optimal value of ϵ? Going back to the original problem, for effect, we have

$$\begin{aligned}\max_{\epsilon}\ & [4(\epsilon/2)]-\frac{1}{2}\left[4(\epsilon/2)^2\right]-(\epsilon/2)^2\left[-\rho+\rho^2-\rho\right]-(\epsilon/2)^2\left[-\rho+\rho^2-\rho\right]\\ \text{(s.t.)}\ & \epsilon\geq 0.\end{aligned}\tag{8}$$

or

$$\begin{aligned}\max_{\epsilon}\ & [2\epsilon]-\frac{1}{2}\left[\epsilon^2\right]-2(\epsilon/2)^2\left[\rho^2-2\rho\right]\\ \text{(s.t.)}\ & \epsilon\geq 0.\end{aligned}\tag{9}$$

Maximizing this in an unconstrained way, the maximum occurs when $\epsilon=2/(1-\rho)^2$, which satisfies $\epsilon>0$, so we get a final solution of

$$\alpha_A=\alpha_B=\alpha_C=\alpha_D=\frac{1}{(1-e^{-4/\sigma^2})^2}.\tag{10}$$

To reconstruct the primal, note that each vector is a support vector, taking A for instance we want (looking at the kernelized version of $y^i(\underline{w}.\underline{x}^i+b)=1$,

$$y^A(\alpha_A y^A K(A,A)+\alpha_B y^B K(B,A)+\alpha_C y^C K(C,A)+\alpha_D y^D K(D,A)+b)=1,\tag{11}$$

or

$$\left(\frac{1}{(1-e^{-4/\sigma^2})^2}-2\frac{1}{(1-e^{-4/\sigma^2})^2}\exp(-4/\sigma^2)+\frac{1}{(1-e^{-4/\sigma^2})^2}\exp(-8/\sigma^2)+b\right)=1,\tag{12}$$

or

$$\frac{1}{(1-e^{-4/\sigma^2})^2}\left(1-2\exp(-4/\sigma^2)+\exp(-8/\sigma^2)\right)+b=1,\tag{13}$$

and it's relatively straightforward to show that the above solves for $b=0$.

Having solved for the bias, we are now in a position to build the SVM classifier:

$$\text{classify}(\underline{x})=\text{sign}\left(\frac{K(A,\underline{x})-K(B,\underline{x})+K(C,\underline{x})-K(D,\underline{x})}{(1-e^{-4/\sigma^2})^2}\right).\tag{14}$$

However, scaling by a positive constant doesn't actually influence the result of the classifier, so we get a final classifier of

$$\text{classify}(\underline{x})=\text{sign}\left(e^{-\frac{1}{\sigma^2}||A-\underline{x}||^2}-e^{-\frac{1}{\sigma^2}||B-\underline{x}||^2}+e^{-\frac{1}{\sigma^2}||C-\underline{x}||^2}-e^{-\frac{1}{\sigma^2}||D-\underline{x}||^2}\right).\tag{15}$$

At this point we are in a position to ask - what does the value σ^2 do to the classifier? Notice what happens as $\sigma^2 \to 0$. As we take this limit, we are dividing by a smaller and smaller value, which is effectively 'blowing up' the exponents. The exponential term with the smallest exponent will be *the point with the minimal distance to* $\underline{x}$! This term is going to 'dominate' the overall sum in the limit as $\sigma^2 \to 0$, which means that the sign of the sum is going to be determined by the point $\underline{x}$ is closest to!

We see then that what the Gaussian classifier is doing in the limit is performing a version of 'k-Nearest Neighbors' classification. k-NN Classification classifies a new point by looking at its nearest neighbors among the training data, and classifying the new point based on the majority classes of its neighbors. The Gaussian Kernel Classifier is effectively approximating this, where σ^2 controls how far away you are willing to look for neighbors (note that as $\sigma^2 \to \infty$, all exponents are roughly the same (near 0) so all points are considered equally regardless of distance/neighborliness), and the sign of the sum is determined by the 'weighted majority class' of these nearest neighbors.

The Gaussian Kernel SVM has two advantages potentially over 'raw' k-Nearest Neighbors classification. The first is that the model is 'smooth' - k-NN has very discontinuous properties based on how many neighbors you choose to classify with; the Gaussian Kernel SVM scales point importance with distance. The second is that while k-Nearest Neighbors requires you to remember *every* training point, the Gaussian Kernel SVM identifies only the potential neighbors that really matter to classification (the support vectors). As $\sigma^2 \to \infty$, fewer neighbors will matter for classification and it is easy to imagine the number of support vectors should go way down; as $\sigma^2 \to 0$, you're essentially looking at 1-NN classification, and it is reasonable that *all* the vectors will be support vectors on some (high dimensional) boundary. But the Gaussian Kernel SVM provides a very convenient framework for both solving a given model and exploring what kinds of models (based on σ^2) you might need for your data.