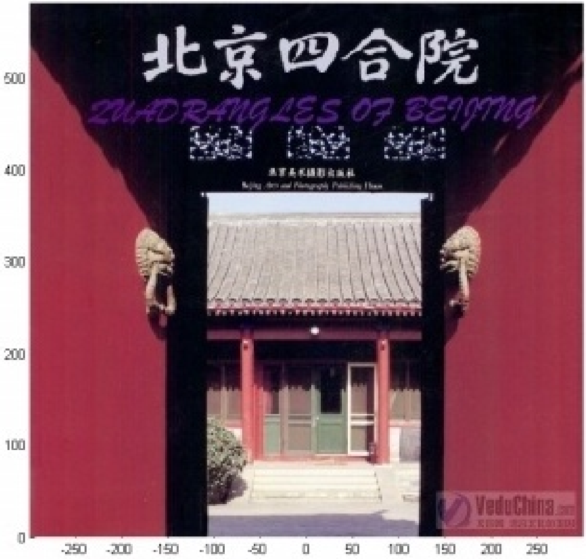
CPSC 8420 Advanced Machine Learning
Week 10: Kernel Support Vector Machines

Dr. Kai Liu

October 20, 2020

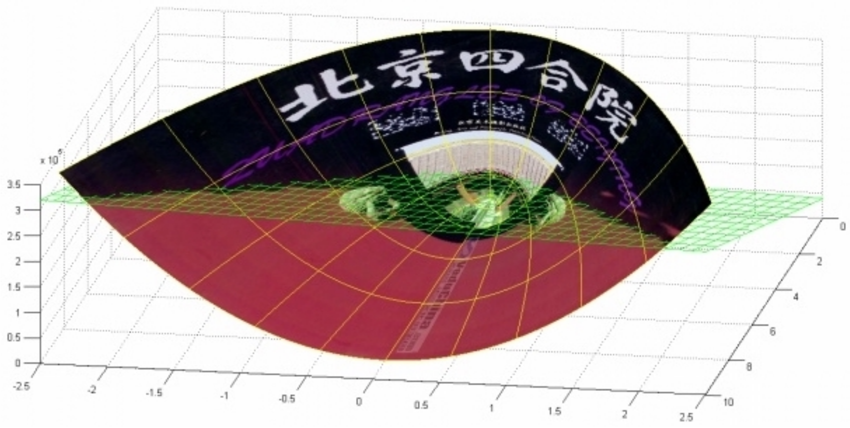
A Gentle Start

How can we seperate the purple characters from the red part?



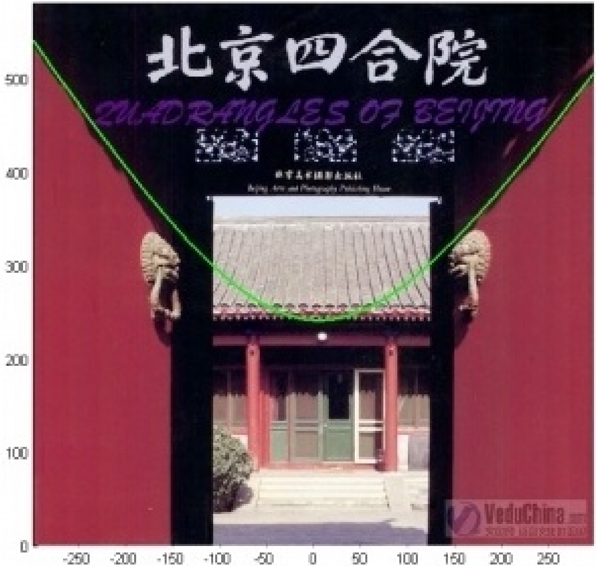
A Gentle Start

Consider the case $P(x, y) = (x^2, \sqrt{2}xy, y^2)$.



A Gentle Start

The green hyperplane corresponds to the green curve here:



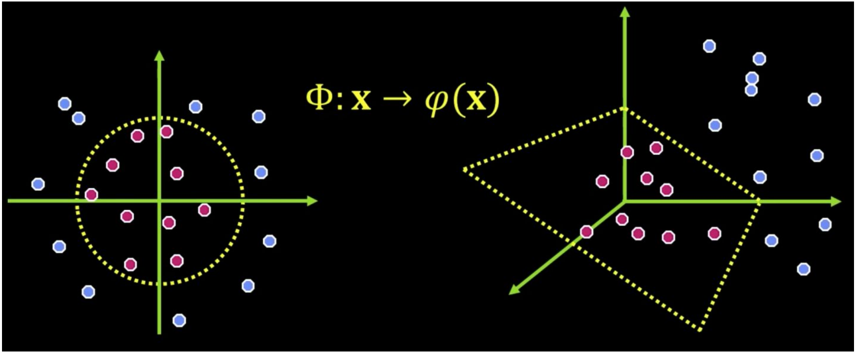
Kernel

In real-world, there are some datapoints they can't be seperated by a hyperplane, even we leverage soft-margin in SVM. But as a theorem: Almost all data points can be linearly seperated in a (sufficiently) high dimension space.

Now let's define function $\phi : \mathbb{R}^d \to \mathbb{R}^p$ and recall soft margin SVM objective:

$$
\begin{aligned}
min \;\; & \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m}\xi_i \\
s.t. \;\; & y_i(w^T x_i + b) \geq 1 - \xi_i \;\; (i = 1, 2, ...m) \\
& \xi_i \geq 0 \;\; (i = 1, 2, ...m)
\end{aligned} \tag{1}
$$

Kernel



Kernel

In the new dimension space, we formulate the objective as:

$$
\begin{gathered}
min \;\; \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m}\xi_i \\
s.t. \;\; y_i(w^T\phi(x_i)+b) \geq 1-\xi_i \;\; (i=1,2,...m) \\
\xi_i \geq 0 \;\; (i=1,2,...m)
\end{gathered}
\tag{2}
$$

the genelized Lagrangian is:

$$
L(w,b,\xi,\alpha,\mu) = \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(w^T\phi(x_i)+b)-1+\xi_i] - \sum_{i=1}^{m}\mu_i\xi_i \tag{3}
$$

Optimization

For $\underbrace{max}_{\alpha_i \geq 0, \mu_i \geq 0,} \underbrace{min}_{w,b,\xi} L(w, b, \alpha, \xi, \mu)$, now we can first optimize $\underbrace{min}_{w,b,\xi} L(w, b, \alpha, \xi, \mu)$ by taking the derivative and set it to be zero:

$$
\begin{aligned}
\frac{\partial L}{\partial w} &= 0 \ \Rightarrow w = \sum_{i=1}^{m} \alpha_i y_i \phi(x_i) \\
\frac{\partial L}{\partial b} &= 0 \ \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0 \\
\frac{\partial L}{\partial \xi} &= 0 \ \Rightarrow C - \alpha_i - \mu_i = 0
\end{aligned}
\tag{4}
$$

Optimization

$$\underbrace{max}_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$$

$$\begin{aligned} s.t. \ \sum_{i=1}^{m} \alpha_i y_i &= 0 \\ C - \alpha_i - \mu_i &= 0 \\ \alpha_i \geq 0, \mu_i &\geq 0 \ (i = 1, 2, ..., m) \end{aligned} \tag{5}$$

if we define $K(x, z) = \phi(x)^T \phi(z)$, then the above equation is equivalent to:

$$\underbrace{min}_{\alpha} \frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{m} \alpha_i$$

$$\begin{aligned} s.t. \ \sum_{i=1}^{m} \alpha_i y_i &= 0 \\ 0 \leq \alpha_i &\leq C \end{aligned} \tag{6}$$

Polynomial Kernel

Now since $\phi(x) \in \mathbb{R}^p$, if p is very large, then it is computationally demanding. Then Kernel Method is proposed, such that instead of operating on high dimension $\phi(x) \in \mathbb{R}^p$, we can alternatively compute on original space $\mathbb{R}^d$. Recall $P(x, y) = (x^2, \sqrt{2}xy, y^2)$, we can verify:

$$
\begin{aligned}
\langle P(v_1), P(v_2) \rangle &= \langle (x_1^2, \sqrt{2}x_1 y_1, y_1^2), (x_2^2, \sqrt{2}x_2 y_2, y_2^2) \rangle \\
&= x_1^2 x_2^2 + 2 x_1 x_2 y_1 y_2 + y_1^2 y_2^2 \\
&= (x_1 x_2 + y_1 y_2)^2 \\
&= \langle v_1, v_2 \rangle^2 \\
&= K(v_1, v_2)
\end{aligned}
\tag{7}
$$

Polynomial Kernel

$$\begin{aligned}
&\phi(\langle x_1, x_2, x_3\rangle) \\
&= \langle x_1^3, x_2^3, x_3^3, \sqrt{3}x_1^2x_2, \sqrt{3}x_1^2x_3, \sqrt{3}x_2^2x_1, \sqrt{3}x_2^2x_3, \sqrt{3}x_3^2x_1, \sqrt{3}x_3^2x_2, \sqrt{6}x_1x_2x_3\rangle \\
&\phi(\langle y_1, y_2, y_3\rangle) \\
&= \langle y_1^3, y_2^3, y_3^3, \sqrt{3}y_1^2y_2, \sqrt{3}y_1^2y_3, \sqrt{3}y_2^2y_1, \sqrt{3}y_2^2y_3, \sqrt{3}y_3^2y_1, \sqrt{3}y_3^2y_2, \sqrt{6}y_1y_2y_3\rangle
\end{aligned} \tag{8}$$

we can verify that $K(X, Y) = \phi(X)^T\phi(Y) = \langle X, Y\rangle^3$, which means we can reduce the compatation from p to d.

More broadly, the kernel $K(x, z) = (x^Tz + c)^k$ (Polynomial Kernel) corresponds to a feature mapping to an $C(k + d, k)$ feature space. However, despite working in this $\mathcal{O}(d^k)$-dimensional space, computing $K(x, z)$ still takes only $\mathcal{O}(d)$ time, and hence we never need to explicitly represent feature vectors in this very high dimensional feature space.

Gaussian Kernel

Radial Basis Function (RBF) is the Kernel used in libsvm by default:

$$
\begin{aligned}
K(\mathrm{x},\mathrm{y}) &= exp(-\frac{||\mathrm{x}-\mathrm{y}||^2}{2\sigma^2}) \\
&= exp(-\frac{1}{2\sigma^2}(\mathrm{x}-\mathrm{y})^T(\mathrm{x}-\mathrm{y})) \\
&= exp(-\frac{1}{2\sigma^2}(\mathrm{x}^T\mathrm{x}+\mathrm{y}^T\mathrm{y}-2\mathrm{y}^T\mathrm{x})) \\
&= exp(-\frac{1}{2\sigma^2}(||\mathrm{x}||^2+||\mathrm{y}||^2)) \cdot exp(\frac{1}{\sigma^2}\mathrm{y}^T\mathrm{x}) \\
&= C \cdot exp(\frac{1}{\sigma^2}\mathrm{y}^T\mathrm{x}) \\
&= C \cdot \sum_{i=0}^{\infty} \frac{(\frac{1}{\sigma^2}\mathrm{y}^T\mathrm{x})^i}{i!} \\
&= \sum_{i=0}^{\infty} \frac{C}{\sigma^{2i} i!}(\mathrm{y}^T\mathrm{x})^i
\end{aligned} \tag{9}
$$

Kernel SVM

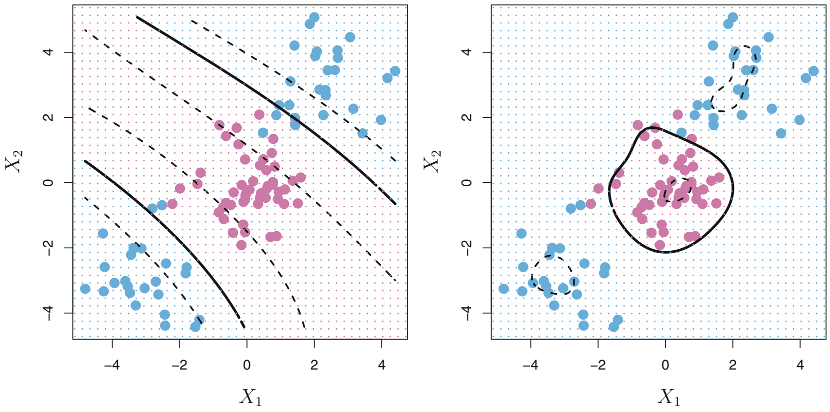
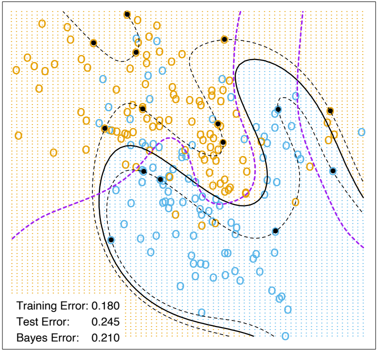


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

Kernel SVM

Two nonlinear SVMs for the mixture data. The upper plot uses a 4-th degree polynomial kernel, the lower a radial basis kernel. The radial basis kernel performs the best (close to Bayes optimal), as might be expected given the data arise from mixtures of Gaussians. The broken purple curve in the background is the Bayes decision boundary.

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space