CISC 371 Class 34

SVM – Kernel Classification

Texts: [1] pp. 432–434

Main Concepts:

- *Kernel function in the decision inequality*
- *Support vectors used to compute the inequality*
- *Kernels produce nonlinear separation*
- *Gaussian kernel can be sensitive to data vectors*

Sample Problem, Machine Inference: How do we evaluate the performance of Gaussian kernels in a SVM classifier?

In the previous class, we explored how a kernel function can be used in training a SVM on n-dimensional data vectors that were the columns of a design matrix X. The key idea was to replace the symmetric positive definite matrix XX^T with the Gram matrix of the kernel function. This had the effect of computing a separating hyperplane that was embedded in a higher-dimensional space.

After the separating hyperplane is computed, we must next decide on how to use it to score new data and then to classify the new data. We can examine the embedding by expanding the score of some data vector $\vec{x}_j$. The score of the data vector, from the primal Lagrange equation for the SVM, is

$$\begin{aligned} z_j &= \vec{x}_j^T \vec{w} + b \\ &= \vec{x}_j \cdot \vec{w} + b \end{aligned} \tag{34.1}$$

The classification of vector $\vec{x}_j$, using the score of Equation 34.1, is

$$q_j \stackrel{\text{def}}{=} \text{sign}(z_j) \tag{34.2}$$

When we use some function $\phi(\cdot)$ to embed $\vec{x}_j$ in a higher-dimensional space as $\hat{x}_j = \phi(\vec{x}_j)$, and embed the weight vector $\vec{w}$ as $\hat{w} = \phi(\vec{w})$, it is unclear whether the constraint would be correctly managed in Equation 34.1. This does not seems to be a useful avenue to follow.

Instead, let us recall how the weight vector $\vec{w}$ was computed in our earlier derivations. We found, when working with the primal Lagrange equation that involved $\partial/\partial\vec{w}$, that we had the equality

$$\vec{w} = X^T Y \vec{\alpha} \tag{34.3}$$

If we expand the term for $\vec{w}$ from Equation 34.3 into Equation 34.1, and convert the vector products to summation, then we have

$$\begin{aligned} z(\vec{x}_j) &= \vec{w}^T \vec{x}_j + b \\ &= [X^T Y \vec{\alpha}]^T \vec{x}_j + b \\ &= \vec{\alpha}^T Y X \vec{x}_j + b \\ &= \left(\sum_{i=1}^{m} \alpha_i y_i (\vec{x}_i \cdot \vec{x}_j) \right) + b \end{aligned} \tag{34.4}$$

Equation 34.4 is the score for a linear SVM that is used in `MATLAB`.

Next: consider replacing the dot product of Equation 34.4 with the kernel function $\kappa(\vec{x}_j, \vec{x}_i)$. This substitution gives us, for any kernel function, a score that is commonly written as

$$z(\vec{x}_j) = \left(\sum_{i=1}^{m} \alpha_i y_i \kappa(\vec{x}_i, \vec{x}_j) \right) + b \tag{34.5}$$

Consider the j^{th} column of the Gram matrix K, which is $\vec{k}_j$. The i^{th} entry of $\vec{k}_j$ is $K_{ij} = \kappa(\vec{x}_i, \vec{x}_j)$ so we can also write Equation 34.5 as

$$z(\vec{x}_j) = \vec{\alpha}^T Y \vec{k}_j + b \tag{34.6}$$

The non-support vectors have a Lagrange multiplier of zero, so Equation 34.5 and Equation 34.6 are inefficient. We can improve the computation if we let $\mathbb{N}_S$ be the set of support vectors in the design matrix X. We can compute a score for a vector $\vec{x}_j$ of a kernel SVM as

$$z(\vec{x}_j) = \left(\sum_{i \in \mathbb{N}_S} \alpha_i y_i \kappa(\vec{x}_i, \vec{x}_j) \right) + b \tag{34.7}$$

Any vector $\vec{x} \in \mathbb{R}^n$, whether it is a given data vector $\vec{x}_j$ or a new vector, can be classified. The vector $\vec{x}$ is in Class +1 if its score is non-negative, and is in Class -1 otherwise. Thus, we can write the decision of how to classify $\vec{x}$ using the function

$$q(\vec{x}) = \text{sign}(z(\vec{x})) \tag{34.8}$$

The score in Equation 34.7, and the corresponding decision in Equation 34.8, are those presented in the current MATLAB documentation for the SVM.

How do various kernels work in practice? One that is common, and easily understood in 2D, is the Gaussian kernel. We can use the data provided in the extra notes for this class to train a SVM. Basic data, with 6 vectors, are shown in Figure 34.1.

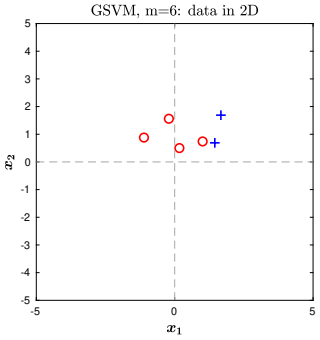


Figure 34.1: Sample 2D data. The vectors with label +1 are plotted as a plus sign; vectors with label -1 are plotted as open circles.

An SVM was trained using these data and a basic Gaussian kernel, with no scaling to account for variance. The level curve of the score function is the *decision curve* for the SVM, which is shown in Figure 34.2.

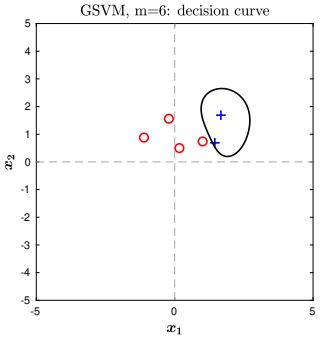
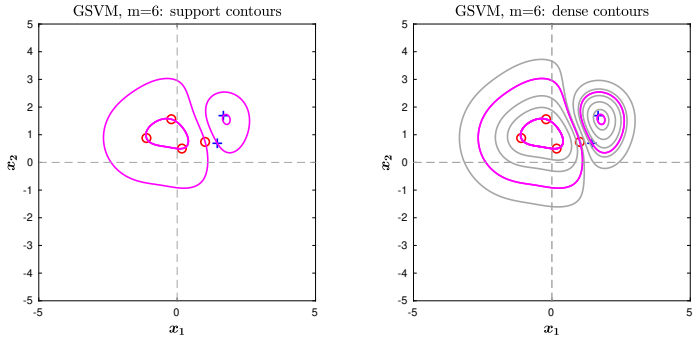


Figure 34.2: Sample 2D data. The vectors with label +1 are plotted as a plus sign; vectors with label -1 are plotted as open circles. The black line separates the classes that were computed using a Gaussian kernel.

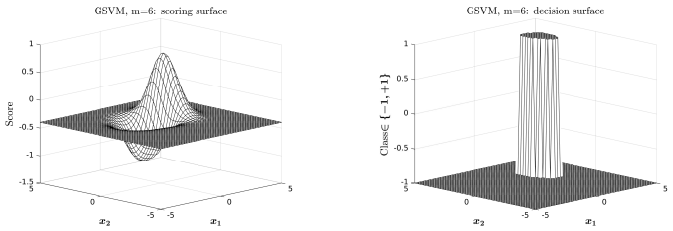
The Lagrange multipliers α_i of the support vectors provide values for the level curves of the decision surface. The multiplier level curves are shown in Figure 34.3(A), with additional level curves shown in Figure 34.3(B). These level curves suggest that there is a maximum near the cluster of Class +1 data and a minimum near the cluster of Class -1 data.



(A) **(B)**

Figure 34.3: Sample 2D data. The vectors with label +1 are plotted as a plus sign; vectors with label -1 are plotted as open circles. (A) The cyan contours are the level curves of the scores for the support vectors. (B) More contours indicate the presence of a local maximum and a local minimum.

The score of $\vec{x}$ from Equation 34.7 can be used as the "height" of the score surface at the point $\vec{x}$. The classification of Equation 34.8 can also be used as the "height" of the decision surface at the point $\vec{x}$. The score surface for the 6 data vectors is shown in Figure 34.4(A); the decision surface is shown in Figure 34.4(B).



(A) **(B)**

Figure 34.4: Score surface and decision surface for 6 data vectors. (A) The scores plotted as a surface mesh; the local maximum and minimum are easily distinguished. (B) The decision surface isolates the local maximum; most data vectors would be classified as -1 by this surface.

Adding a single data vector, so that there are 7 columns in the design matrix X, can substantially change the SVM. The data and a level curve are shown in Figure 34.5(A). The score surface

for the 7 data vectors is shown in Figure 34.5(B); the decision surface is shown in Figure 34.5(C).

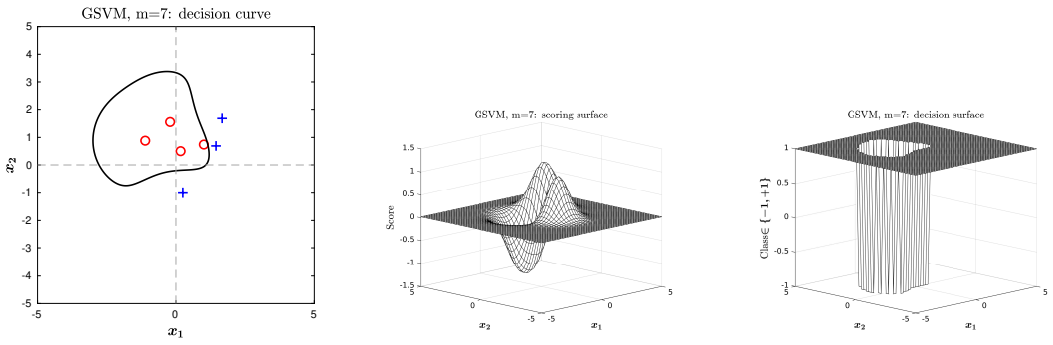


Figure 34.5: Score surface and decision surface for 7 data vectors. (A) The vectors with label +1 are plotted as a plus sign; vectors with label -1 are plotted as open circles. The decision curve, in black, is a single contour. (B) The scores plotted as a surface mesh; the local maxima and minimum are visible. (C) The decision surface isolates the local minimum; most data vectors would be classified as +1 by this surface.

The substantial change in the SVM, caused by the addition of a single vector to the data set, is readily apparent. The score surface was changed from having a single global maximum to having a secondary local maximum. The decision surface was changed from classifying most of the 2D plane as Class -1 to classifying most vectors as Class +1.

Observation: RBF terminology.

The Gaussian kernel operates by using each support vector as a central point for a function that has a symmetric Gaussian distribution. The score function of Equation 34.7 computes a linear sum of the Gaussian distributions. A common term for a linear sum of a fixed set of functions is that the functions are *basis functions*. The Gaussian distributions are symmetric, so together they are *Gaussian radial basis functions*; each function is abbreviated as a GRBF. In the SVM literature, the Gaussian adjective is often omitted so each is referred to as a RBF.

Extra Notes

Data for creating the plots

The data used to create the plots for a 6-vector design matrix and label vector were:

Type	**Data**					
$\boldsymbol{X}$	0.17	-1.11	1.45	1.67	-0.21	1.01
	0.50	0.88	0.69	1.69	1.56	0.74
$\boldsymbol{y_j}$	-1	-1	1	1	-1	-1

The data used to create the plots for a 7-vector design matrix and label vector were:

Type	**Data**						
$\boldsymbol{X}$	0.17	-1.11	1.45	1.67	-0.21	1.01	1.00
	0.50	0.88	0.69	1.69	1.56	0.74	-1.00
$\boldsymbol{y_j}$	-1	-1	1	1	-1	-1	1

End of Extra Notes

References

[1] Hastie T, Tibshirani R, Friedman J: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, 2009