Assignment 5

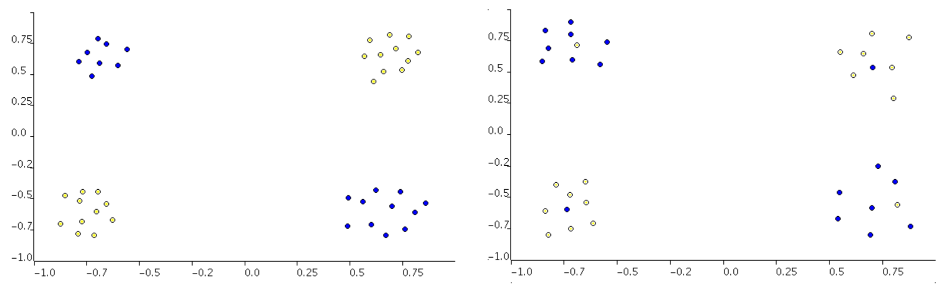
Machine Learning, Summer term 2015

Tobias Lang and Ulrike von Luxburg

To be discussed in exercise groups on May 18/20

Total number of points: 20

Exercise 1 (Play with SVM, 3 points)
In this exercise, you play with applets to get a better understanding of SVMs. You can find two SVM applets here, but you may also use alternatives:
`www.ml.inf.ethz.ch/education/lectures_and_seminars/annex_estat/Classifier/JSupportVectorApplet.html`
`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
Try to set the training points as depicted in Figure 1-a.



(a) (b)

Figure 1: (a) Train data. (b) Train data with outlier.

Now train the SVM with the following settings. Capture the output screen for your report.

1. Linear kernel (simple dot product) with $C = 100$.
2. Polynomial kernel of degree 2, and degree 8. Choose a proper C.
3. Gaussian kernel (radial basis function): Try different values for σ. Note that often a different parameterization is used (for example, in the applets mentioned above): $\beta = 1/(2\sigma^2)$. Try different values for β in this case. Choose a proper C.

Now add noise to your training data as depicted in Figure 1-b. Try the Gaussian kernel with $\beta = 10$ and $C = 0, 10, 1000$. Based on your observation, describe the effect of the parameter C.

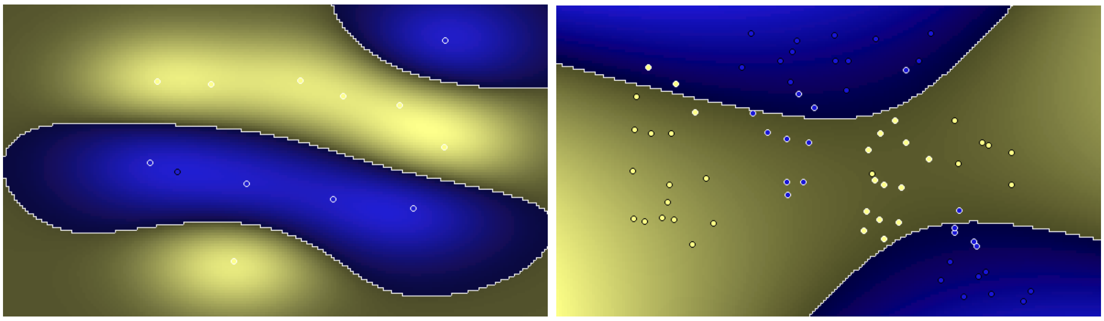
Exercise 2 (Play with SVM II, 3 points)
Use again the applets for SVMs as in the previous exercise.

1. Create a data-set which for fixed $C = 0.0001$ is perfectly separable with a polynomial kernel of degree 3, but not with a polynomial kernel of degree 2.
2. Create a data-set which for fixed $C = 0.0001$ is perfectly separable with a radial basis function kernel with small σ (e.g., $\sigma = 0.1$ / $\beta = 50$), but not with large σ (e.g., $\sigma = 10$ / $\beta = 1/200$).
3. Create a data-set which for fixed $C = 0.0001$ is perfectly separable with a radial basis function kernel, but not with a polynomial kernel of degree 3.

Exercise 3 (Understanding kernel SVM, 2 points) The output of kernel SVM in two problems with different parameters and kernels are depicted in Figure 2-a and 2-b. For each figure, answer the following questions:

- Which type of kernel is used: linear, polynomial or rbf?
- Argue if this is a good classifier? How we should change the parameters of the classifier to avoid this problem?

Can you guess the support vectors in Figure 2-a?



(a) (b)

Exercise 4 (Building new kernels, 3 points) Assume that $K_1, K_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are kernel functions. Which of the following functions are also a valid kernel? Prove or bring a counterexample.

- $K = \alpha K_1$ for $\alpha > 0$
- $K = K_1 + K_2$
- $K = K_1 - K_2$

Exercise 5 (Polynomial kernel, 3 points) Consider the second degree polynomial kernel function $K(x, y) = (x^T y + 1)^2$ with inputs $x, y \in \mathbb{R}^2$.

- Show that the corresponding feature map function is $\Phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^T$ where $x = (x_1, x_2)^T \in \mathbb{R}^2$.
- If we use the second degree polynomial kernel for inputs from $\mathbb{R}^d$, what would be the dimensionality of the corresponding feature space?

Exercise 6 (SVM cancer detection, 6 points) In this exercise you should learn a (soft margin) SVM that classifies cancers as either benign (-1) or malignant (+1) depending on the characteristics of sample biopsies. Load the patients data from `cancer_data2014.mat` (available on the course webpage). For every patient, 9 attributes are measured:

1- Clump thickness 2- Uniformity of cell size 3- Uniformity of cell shape 4- Marginal Adhesion 5- Single epithelial cell size 6- Bare nuclei 7- Bland chomatin 8- Normal nucleoli 9- Mitoses.

For $C \in \{0.01, 0.1, 0.5, 1, 5, 10, 50\}$, plot the train and the test error (with respect to the 0-1-loss) as a function of C for a linear SVM. What is the effect of choosing a large C on the training error? Does this effect coincide with what you are expecting?
Now, try out different kernel functions. Find optimal kernel parameters and C by cross-validation. Which SVM kernel performs best on the test data?
As an SVM implementation, you can use the corresponding Matlab libraries or any free implementations you can find, for example libSVM (which you need to compile first):
`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`