

### Paper and pen problems

1. Analytically solve the following problem: Minimize  $\min f(x, y) = x^2 + y^2 + 3$ .
2. Find  $\mathbf{w}$  in the optimal margin (linear) SVM (in pp. 11) with inputs  $\mathbf{x}_1 = [1 \ 1]^T \in C_+$ ,  $\mathbf{x}_2 = [0 \ 0]^T \in C_-$ ,  $\mathbf{x}_3 = [2 \ 2]^T \in C_+$ ,  $\mathbf{x}_4 = [-3 \ 2]^T \in C_-$ . Hint: Locate support vectors first, and then the decision boundary is obvious without computation.
3. In the AdaBoost example given in the lecture (pp. 2-5), we are able to correctly classify linearly non-separable data. It implies that we use a nonlinear function during the computational steps. In which step did we introduce the nonlinear function and what was the function?
4. We use a special method to classify the iris dataset (see also Fig. below). The used discriminative classifiers are binary classifiers (meaning that it can only indicate whether the input sample is from class + or class -). To extend the model to three classes, we use two identical models trained with different training sets. Then, during the test (prediction) phase, the final decision is made with the following rule:

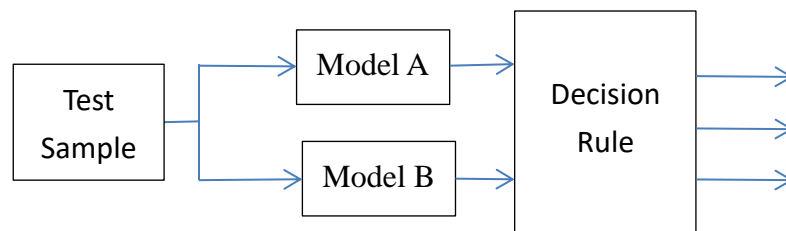
**If** (model A predicts +) **then** class is Versicolor

**else if** (model B predicts +) **then** class is Setosa

**else** class is Virginica

Which classes of iris flowers should be included in the class “-” of model A?

How about class “+” of model B?



5. We have a dataset  $S = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$ . Follow the k-means algorithm to complete the assignment step and the update step for one run (iteration). Use  $k = 2$ , initially  $\boldsymbol{\mu}_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$ , and  $\boldsymbol{\mu}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$  in the computation.

### Computer-based problems

6. Write a program to find the optimal solution of  $\min f(x, y) = x^2 + y^2 + 3$  by

using the gradient descent method. Let the initial value be  $x = 1, y = 1.5$ . Figure out the maximum acceptable value of  $\eta$  where the program still converges.

7. Use the SVM classifier with RBF kernel for the classification task of cancer data. Do a 70/30 split for training and test set. Repeat the trials 50 times and compute the average accuracy. Observe and plot the variation of training and testing accuracy when  $C$  is from 0.1 to 3.0 in an increment of 0.1. The parameter  $C$  is an L2 regularization parameter, particularly useful if the classes are imbalanced.
8. Repeat problem 7, but using the Adaboost approach. Use the CART tree with depth of 2 as the weak classifier. How many weak classifiers are needed to have no training error? Plot the accuracy curve for number of trees from 5 to 95 in an increment of 10. Do we observe any accuracy improvement along with the number of weak classifiers? If we use a tree depth of one, can we get satisfactory results using less than 100 classifiers?
9. Use the cancer dataset to plot the ROC. In this case, we just build one model (for benign). If the predicted likelihood of the input sample is greater than a threshold, the sample belongs to benign, otherwise malignant. Compare the ROC curves of 5-NN and GMM (5 components). Can you point out which classifier is better based on ROC? Note: Usually we don't use GMM in a way we use it now.
10. Apply the k-means algorithm to the Wine recognition dataset in sklearn. Use `n_clusters=3` in this example. Once the algorithm converges, you can also check whether the clusters roughly match the corresponding classes. Specifically, print out the number of samples from each class for each cluster. For example, cluster 0 (also 1, and 2) has xx samples from class 0, xx samples from class 1, and xx samples from class 2. Do you get the same results every time executing the program? If not, where is the randomness from? Recall that we use all samples in the dataset for clustering, the randomness is not from `train_test_split()` in sklearn.