6.867 Machine learning

Mid-term exam

October 15, 2008

(2 points) Your name and MIT ID:

Problem 1

Assume that we have a training set consisting of examples $(\underline{x}_i, y_i)$ for $i = 1 \ldots n$. The task is a binary classification problem so each label y_i is either -1 or $+1$.

In training a **hard margin SVM with bias**, the final classifer is $\hat{y} = \text{sign}(\hat{\underline{\theta}} \cdot \underline{x} + \hat{\theta}_0)$ where the parameters $\hat{\underline{\theta}}$ and $\hat{\theta}_0$ solve the following primal optimization problem:

Primal: find $\underline{\theta}, \theta_0$ that
minimize $\frac{1}{2}||\underline{\theta}||^2$
subject to $y_i \left(\underline{\theta} \cdot \underline{x}_i + \theta_0\right) \geq 1,\ i = 1, \ldots, n$

Dual: find α_i that
maximize $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \underline{x}_i \cdot \underline{x}_j$
subject to $\alpha_i \geq 0,\ \sum_{i=1}^{n} \alpha_i y_i = 0$

We have also included the dual for your reference.

1.1 (5 points) Assume that $n = 4$, and that $\underline{x}_1 = (1, 1)^T$, $\underline{x}_2 = (2, 2)^T$, $\underline{x}_3 = (-1.5, -1.5)^T$, and $\underline{x}_4 = (4, 4)^T$. We now train an SVM with bias, and in addition **with slack variables**. Show that for any labelling of the four training examples, the optimal parameter vector $\hat{\underline{\theta}} = (\hat{\theta}_1, \hat{\theta}_2)^T$ has the property that $\hat{\theta}_1 = \hat{\theta}_2$.

1.2 (5 points) Consider the SMO algorithm applied to training a hard-margin SVM with slack variables and a bias variable. Initially all dual variables α_i for $i = 1 \ldots n$ are set to 0. At each step in the SMO algorithm, two variables α_i and α_j are chosen such that $y_i = y_j$; these two variables are optimized in the usual way for SMO. What solution will you find with this constrained SMO procedure? Give a justification for your answer.

1.3 (5 points) Consider training an SVM with slack variables, but with no bias variable. The kernel used is $K(\underline{x}, \underline{z})$; it has the property that for any two points $\underline{x}_i$ and $\underline{x}_j$ in the training set, $-1 < K(\underline{x}_i, \underline{x}_j) < 1$. $K(\underline{x}_i, \underline{x}_i) < 1$ as well. There are n points in the training set. Show that if the slack-variable constant C is chosen such that $C < \frac{1}{n-1}$, then all dual variables α_i are non-zero (i.e., all points in the training set become support vectors).

1.4 (5 points) Consider the kernel

$$K(\underline{x}, \underline{z}) = \underline{x} \cdot \underline{z} + 4\,(\underline{x} \cdot \underline{z})^2$$

where the vectors $\underline{x}$ and $\underline{z}$ are 2-dimensional. This kernel is equal to an inner product $\phi(\underline{x}) \cdot \phi(\underline{z})$ for some definition of ϕ. What is the function ϕ?

Problem 2

As you may have suspected, the course staff enjoys writing endless varieties of SVM-like training methods. It is time to sort them out a bit. Figure 1 shows both decision boundaries and support vectors (circled) from different SVM-like training methods. In all cases, the boundaries correspond to $\hat{\underline{\theta}} \cdot \underline{x} + \hat{\theta}_0 = 0$, where $\hat{\theta}_0 = 0$ unless θ_0 is included in the training method. J_+ and J_- index positive ('x') and negative ('o') training examples, respectively. There are five methods and four figures. Please assign *each method to all the figures that they could potentially produce* (there may be multiple choices).

2.1 **(2 points)**

$$\min \ \frac{1}{2}\|\underline{\theta}\|^2 + C\sum_{t=1}^{n}\xi_t \ \text{ s.t.}$$
$$\xi_t \geq 0, \ \ y_t(\underline{\theta}^T\underline{x}_t + \theta_0) \geq 1 - \xi_t \ \ t = 1, \ldots, n$$

where $C = \infty$.

2.2 **(2 points)**

$$\min \ \frac{1}{2}\|\underline{\theta}\|^2 + C\sum_{t=1}^{n}\xi_t \ \text{ s.t.}$$
$$\xi_t \geq 0, \ \ y_t(\underline{\theta}^T\underline{x}_t) \geq 1 - \xi_t, \ \ t = 1, \ldots, n$$

where $C = \infty$.

2.3 **(2 points)**

$$\min \ \frac{1}{2}\|\underline{\theta}\|^2 + C\sum_{t=1}^{n}\xi_t \ \text{ s.t.}$$
$$\xi_t \geq 0, \ \ y_t(\underline{\theta}^T\underline{x}_t) \geq 1 - \xi_t, \ \ t = 1, \ldots, n$$

where $C = 1$.

2.4 **(2 points)**

$$\min \ \frac{1}{2}\|\underline{\theta}\|^2 + C_+\sum_{t\in J_+}\xi_t + C_-\sum_{t\in J_-}\xi_t \ \text{ s.t.}$$
$$\xi_t \geq 0, \ \ y_t(\underline{\theta}^T\underline{x}_t) \geq 1 - \xi_t, \ \ t = 1, \ldots, n$$

where $C_+ = 1$ and $C_- = 0$.

2.5 **(2 points)**

$$\min \ \frac{1}{2}\|\underline{\theta}\|^2 + C_+\sum_{t\in J_+}\xi_t + C_-\sum_{t\in J_-}\xi_t \ \text{ s.t.}$$
$$\xi_t \geq 0, \ \ y_t(\underline{\theta}^T\underline{x}_t) \geq 1 - \xi_t, \ \ t = 1, \ldots, n$$

where $C_+ = \infty$ and $C_- = 0$.

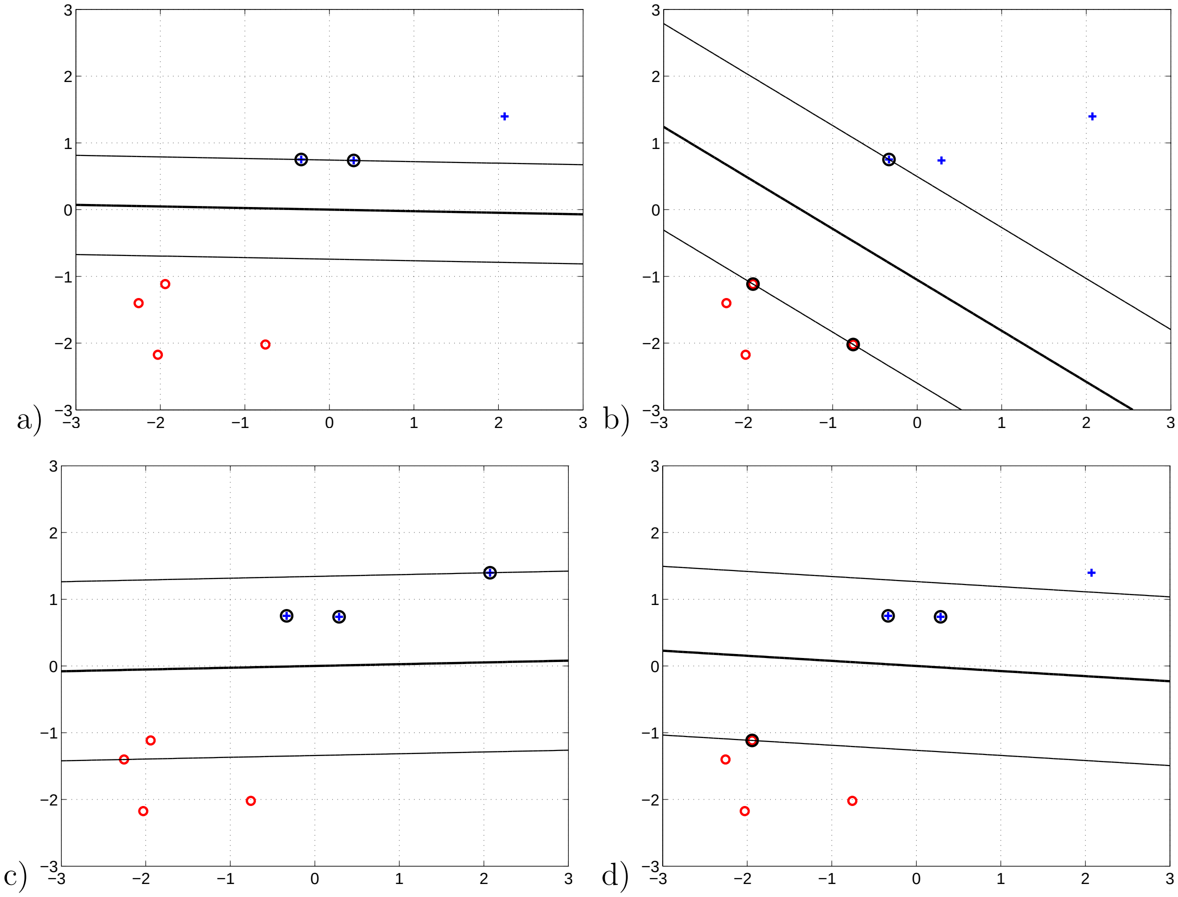


Figure 1, problem 2: plots of $\underline{\hat{\theta}} \cdot \underline{x} + \hat{\theta}_0 = 0$ for different training methods along with the support vectors. Points labeled $+1$ are in blue, points labeled -1 are in red. The line $\underline{\hat{\theta}} \cdot \underline{x} + \hat{\theta}_0 = 0$ is shown in bold; in addition we show the lines $\underline{\hat{\theta}} \cdot \underline{x} + \hat{\theta}_0 = -1$ and $\underline{\hat{\theta}} \cdot \underline{x} + \hat{\theta}_0 = 1$ in non-bold. Support vectors have bold circles surrounding them.

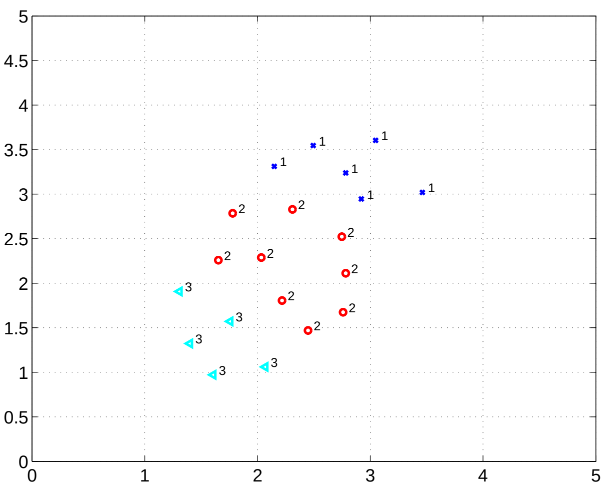
Problem 3

3.1 (6 points) Consider a three-class classification problem shown in the figure below (left figure). Design an output code matrix R for linear classifiers such that a) each binary subtask is linearly separable as far as the training set is concerned, and b) the multi-class classifier has zero training error in the sense that the predictions

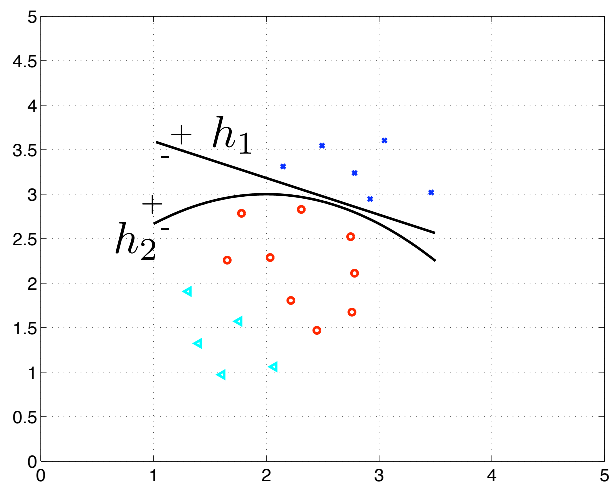
$$\hat{y}_i = \text{argmax}_{y=1,2,3} \left\{ \sum_{j=1}^{k} R(y,j) h(\underline{x}_i; \theta_j) \right\}$$

are correct for all the training points in the figure. k is your choice of the number of columns in the matrix. Here $h(\underline{x}_i; \theta_j) = \text{sign}(\underline{\theta}_j \cdot \underline{x}_i + \theta_{j0})$ denotes the binary output from a linear classifier corresponding to task j in the output code. Some of the columns of R below may be left empty as they may not be needed.

$$\begin{matrix} y=1 \\ y=2 \\ y=3 \end{matrix} \begin{bmatrix} (\quad) & (\quad) & (\quad) & (\quad) & (\quad) & (\quad) \\ (\quad) & (\quad) & (\quad) & (\quad) & (\quad) & (\quad) \\ (\quad) & (\quad) & (\quad) & (\quad) & (\quad) & (\quad) \end{bmatrix} = R$$



Multi-class problem for 3.1



1 versus {2 and 3} solutions for 3.2 and 3.3.

3.2 (points 4) In order for the output code to work well for test examples, we would like the corresponding binary classifiers to generalize well. Consider 1 versus {2 and 3} classification task in the figure (right side). We will consider two sets of classifiers to solve this task:

$\mathcal{H}_1$: $h_1(\underline{x};\theta) = \text{sign}(\underline{\theta} \cdot \underline{x} + \theta_0)$ with adjustable parameters $\underline{\theta}$ and θ_0.

$\mathcal{H}_2$: $h_2(\underline{x};\theta) = \text{sign}(\|\underline{x} - \underline{\mu}\| - 2)$ with adjustable parameters $\underline{\mu}$.

What are the VC-dimensions of these two sets of classifiers $\mathcal{H}_1$ and $\mathcal{H}_2$ in *two dimensions*?

3.3 (points 3) We trained both of above classifiers based on the data in the figure. The resulting decision boundaries are also shown in the figure. Based on the VC dimension of the two classifiers, which classifier would you expect to generalize better? Briefly justify your answer.

3.4 (points 8) Next, we'll consider a variant of the perceptron algorithm, for a 3-class problem (each label y takes a value of $1, 2$, or 3). The training set consists of n examples $(\underline{x}_i, y_i)$ for $i = 1 \ldots n$, where $\underline{x}_i \in \mathcal{R}^d$, and $y_i \in \{1, 2, 3\}$. The following figure shows the algorithm:

Initialization: for $y \in \{1, 2, 3\}$, set $\underline{\theta}_y = \underline{0}$.

Algorithm:
Repeat until convergence:

- For $i = 1 \ldots n$:
  - Set $z = \arg\max_{y \in \{1,2,3\}} \; \underline{\theta}_y \cdot \underline{x}_i$
  - If $z \neq y_i$:
    - $\underline{\theta}_{y_i} = \underline{\theta}_{y_i} + \underline{x}_i$
    - $\underline{\theta}_z = \underline{\theta}_z - \underline{x}_i$

Classification function on a test point $\underline{x}$:

$$f(\underline{x}) = \arg \max_{y \in \{1,2,3\}} \; \underline{\theta}_y \cdot \underline{x}$$

Question: We'd like to derive a kernel version of this perceptron algorithm. Assume the kernel we will use is $K(\underline{x}, \underline{z})$. Complete the algorithm below to give a kernelized form of the perceptron algorithm shown above.

Initialization: for $y \in \{1, 2, 3\}$, for $i = 1 \ldots n$, set $\alpha_{i,y} = 0$

Algorithm:

Repeat until convergence:

Classification function on a test point $\underline{x}$:

$$f(\underline{x}) = \arg \max_{y \in \{1,2,3\}} \sum_{i=1}^{n} \alpha_{i,y} K(\underline{x}, \underline{x}_i)$$

Problem 4

One evening we thought we had come up with a great machine learning approach to predicting movie ratings. The idea was to base the predictions solely on positive training examples, movies we already know we like ($y = +1$), and simply ignore (as far as the training is concerned) all the negative examples ($y = -1$). Assume movies are represented by vectors $\underline{x}_1, \ldots, \underline{x}_m$, where $\underline{x}_j \in \mathcal{R}^d$. We created these vectors from movie descriptions (automatically, of course).

Our primal SVM optimization problem, written only for positive examples without offset, is given by

$$\min \ \frac{1}{2}\|\underline{\theta}\|^2 \quad \text{subject to } \underline{\theta} \cdot \underline{x}_j \geq 1, \quad j \in J_+ \tag{1}$$

where $J_+ \subset \{1, \ldots, m\}$ indexes our positive training examples (movies we already know we like).

4.1 (3 points) What would the solution $\hat{\theta}$ be if we included an offset parameter θ_0, i.e., changed the constraints to be $\underline{\theta} \cdot \underline{x}_j + \theta_0 \geq 1$?

4.2 (2 points) Assume we can find the solution $\hat{\underline{\theta}}$ to the problem described in Eq.(1). What is the value of $\min_{j \in J_+}(\hat{\underline{\theta}} \cdot \underline{x}_j)$?

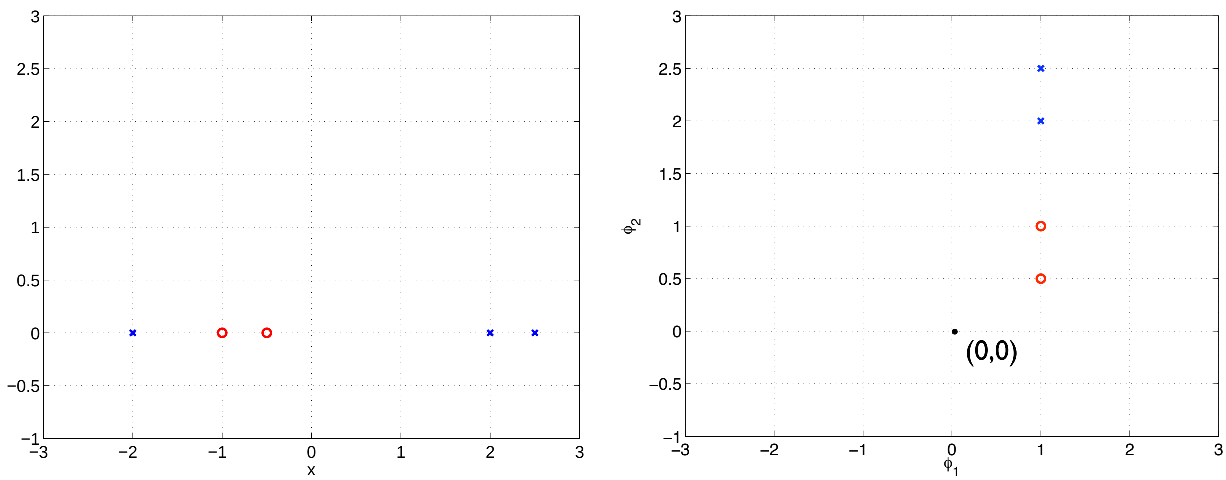
4.3 (3 points) Suppose again that the solution $\hat{\underline{\theta}}$ to Eq.(1) exists. Based on this $\hat{\underline{\theta}}$, we predict labels for movies $\underline{x}$ (new and training examples) according to

$$\hat{y} = \begin{cases} 1, & \text{if } (\hat{\underline{\theta}} \cdot \underline{x}) \geq \min_{j \in J_+}(\hat{\underline{\theta}} \cdot \underline{x}_j) - \epsilon \\ -1, & \text{otherwise} \end{cases}$$

for some small $\epsilon > 0$. Would this decision rule ensure that all the training movies, positive and negative, are classified correctly? Briefly justify your answer.

4.4 (2 points) The problem might sometimes get a little challenging. The figure below (see left, below question 4.5) shows the movie data, positive ('x') and negative ('o') examples, when movies are represented by real numbers x_j. Briefly describe why we cannot solve Eq.(1) in this case.

4.5 (6 points) We will apply the algorithm described in Eq.(1) with a feature mapping, i.e., we replace one dimensional x with $\underline{\phi}(x) = [\,1,\, |x|\,]^T$. In the figure below (right), we have plotted the movie data, positive ('x') and negative ('o'), in the feature coordinates ϕ_1 and ϕ_2. Sketch the solution $\hat{\underline{\theta}}$ in the feature space by drawing $\hat{\underline{\theta}} \cdot \underline{\phi} = 0$ and $\hat{\underline{\theta}} \cdot \underline{\phi} - 1 = 0$ in the figure on the right.



Movie data for problem 4.4 and 4.5. Original space (left). Feature space (right).

4.6 (2 points) Is $\hat{\underline{\theta}} \cdot \underline{\phi}(x) > 0$ at $x = -1$ (Y/N)?

Additional set of figures

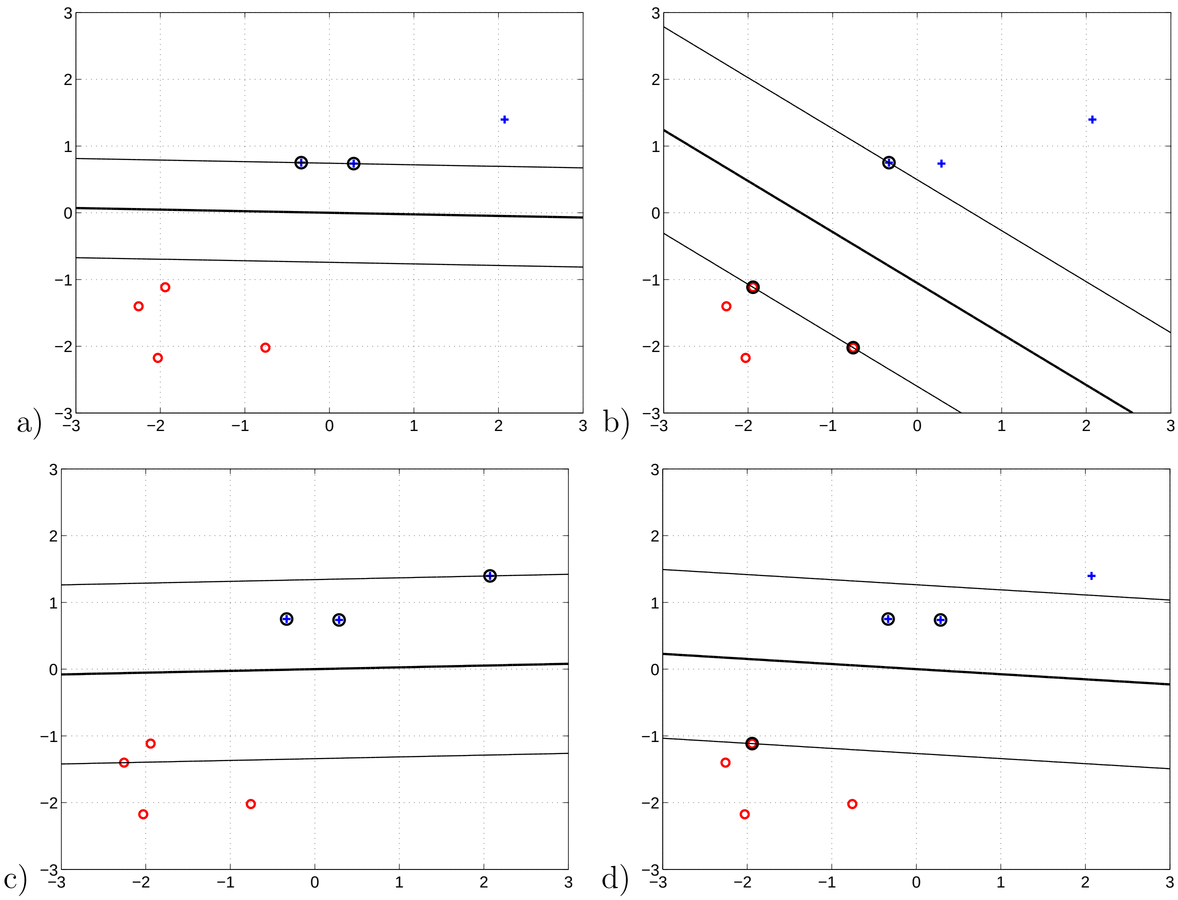
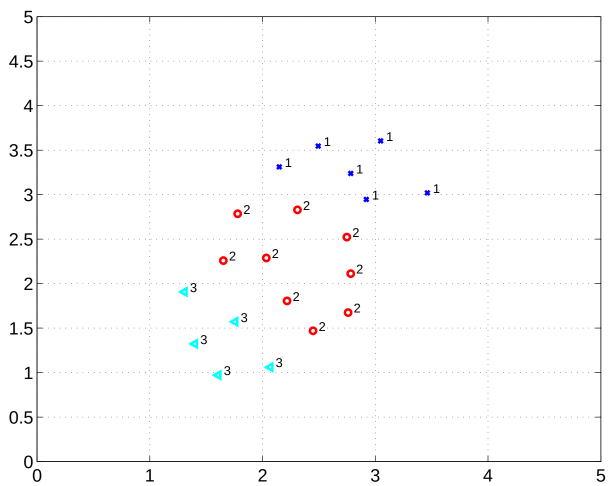
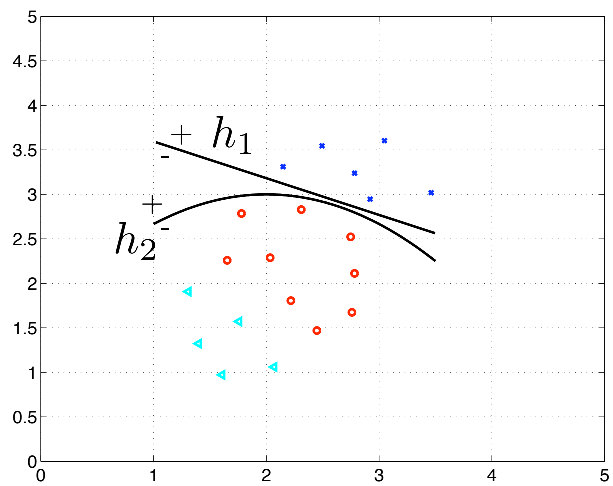


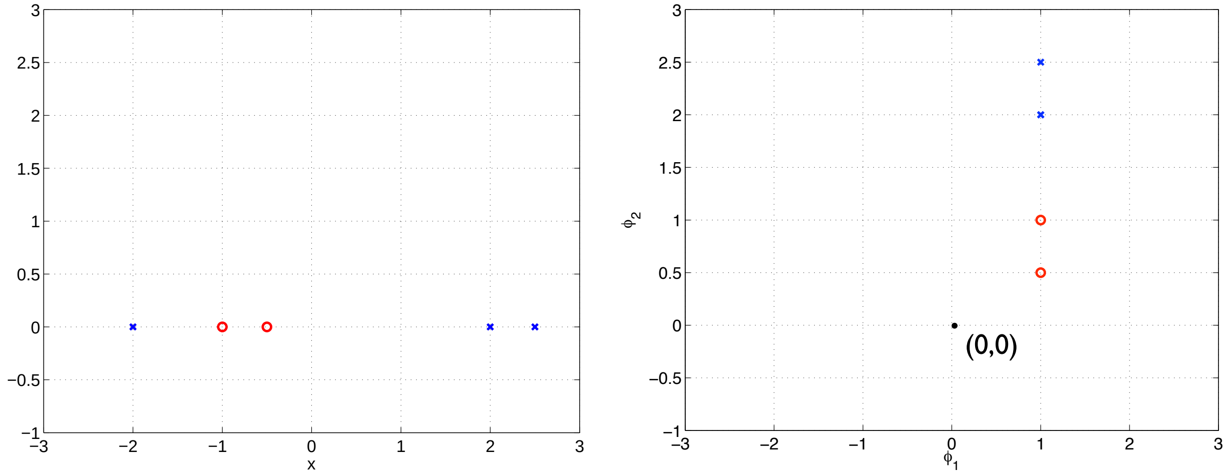
Figure for problem 2: plots of $\hat{\underline{\theta}} \cdot \underline{x} + \hat{\theta}_0 = 0$ for different training methods



Multi-class problem for 3.1



1 versus {2 and 3} solutions for 3.2



New movie data for problem 4.5. Original space (left). Feature space (right).