Pattern Recognition And Machine Learning: Assignment #2

Vaisakh Shaj

Problem 1

Consider $w\epsilon\Re^n$ .Let f(X) = $\langle w, x\rangle$, $\forall x\epsilon\Re^n$ .Find $\|f\|$.

Solution

Since $f(x) =< w, x >, \forall x \in \Re^n$ and $w \in \Re^n$
$\Rightarrow$ f is a bounded linear functional, since its given its possible to represent f in the dot product form in $\Re^n$,which is an RKHS. (By Riesz representation lemma)c

Since f is bounded,

$\|f(x)\| \leq c\|x\|$
$\Rightarrow \|\langle w, x\rangle\| \leq c\|x\|$
Using Cauchy-Schwarz inequality,

$\|f(x)\| = \|\langle w, x\rangle\| \leq \|w\|\|x\|$

$\Rightarrow \frac{\|f(x)\|}{\|x\|} \leq \|w\|$

But $\sup\frac{\|f(x)\|}{\|x\|} = \|f\|$
Therefore, $\|f\| \leq \|w\|$
In general, $\|f\| = \|w\|$

Problem 2

Let the data $(x_i, y_i), i = 1, 2, ...N$, $x_i\epsilon\Re^n$, $y_i\epsilon R$ be generated by ahyperplane. By kernel theory, the function that generates the data can be written as a linear combination of N training points and a bias term. However if the data is n dimensional the equation of the hyper plane consists of n+1 terms. Are these the same? Justify your answer.

Solution

Yes, they are the same.

PROOF

f(x) = $\alpha_1 * K(x_1, x) + \alpha_2 * K(x_2, x) + ... + \alpha_N * K(x_N, x) + b$, that is in the RKHS the function has N+1 terms including the bias term.

Considering linear Kernel,

Problem 2 continued on next page. . .

f(x) = $\alpha_1 * \langle x_1, x \rangle + \alpha_2 * \langle x_2, x \rangle + ... + \alpha_N * \langle x_N, x \rangle + b$ —— (1)

x = $(x_1, x_2, ...x_n)$, has n terms —— (2)

substituting (2) in (1),

f(x)= $\alpha_{\mathbf{1}} * [x_{11} * x^{(1)} + x_{12} * x^{(2)} + ...x_{1n} * x^{(n)}] + \alpha_{\mathbf{2}} * [x_{21} * x^{(1)} + x_{22} * x^{(2)} + ...x_{2n} * x^{(n)}] + ... +$
$\alpha_{\mathbf{N}} * [x_{N1} * x^{(1)} + x_{N2} * x^{(2)} + ...x_{Nn} * x^{(n)}] + b$

taking $x^(i)$ terms together, we get

f(x) = $[\alpha_1 * x_{11} + \alpha_2 * x_{21} + ... + \alpha_N * x_{N1}] * \mathbf{x^{(1)}} + [\alpha_1 * x_{12} + \alpha_2 * x_{22} + ... + \alpha_N * x_{N2}] * \mathbf{x^{(2)}} + ... +$
$[\alpha_1 * x_{1N} + \alpha_2 * x_{2N} + ... + \alpha_N * x_{NN}] * \mathbf{x^{(N)}} + b$ —-(3)

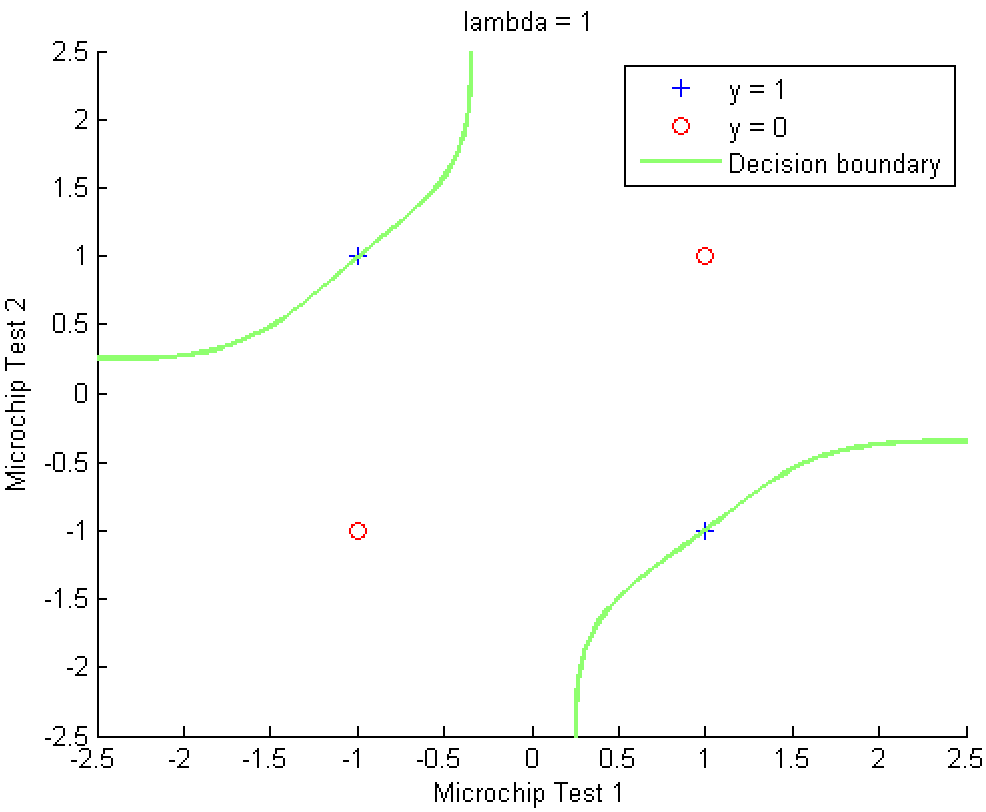
(3) is of the form f(x) = $w_0 * x_0 + w_1 * x_1 + w_2 * x_2 + ... + w_n * x_n$,which has n+1 terms including the bias term.
Hence the function that generates the data written as a linear combination of N training points and a bias term using linear kernel, is essentially the equation of hyperplane for n dimensional data with n+1 terms.

Problem 3

3. Consider the following data $((1,1),-1),(1,-1),1),(-1,-1),-1),(-1,1),1)$
(a) **Plot the nonlinear boundary in the input space.**
(b) **Plot the linear boundary in RKHS space generated by the kernel k(x,y) =** $\langle K_x, K_y \rangle$

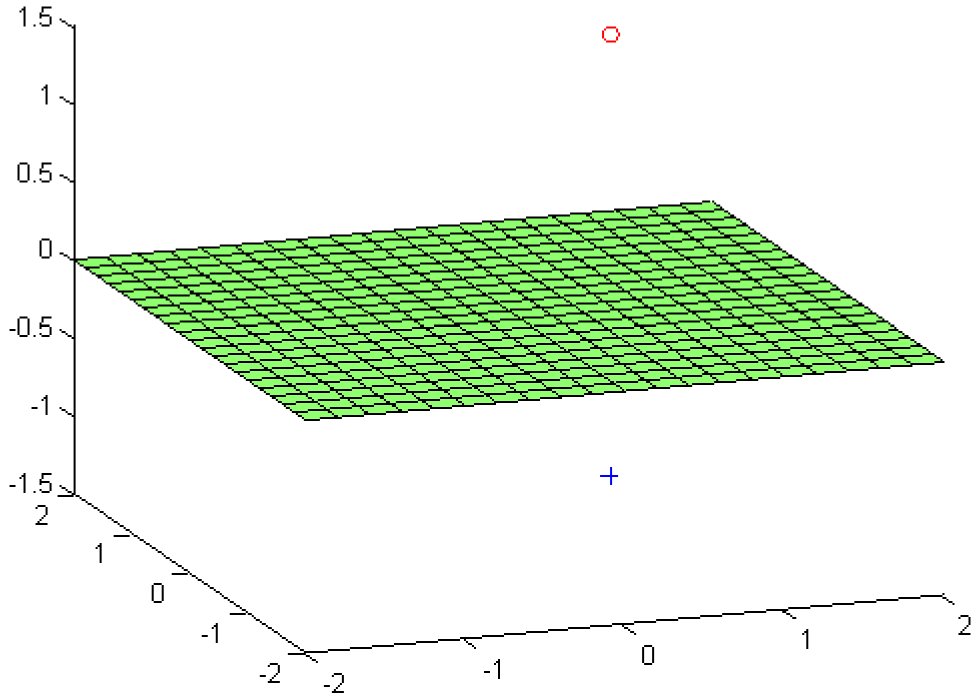
Solution
(a). Non-Linear Decision Boundary in the input space plotted using logistic regression with higher degree polynomial.

Problem 3 continued on next page. . .



(b). Linear Decision Boundary In The RKHS Space Created By Polynomial Kernel Of Order 2

Problem 3 continued on next page. . .


1.5
1
0.5
0
-0.5
-1
-1.5
2
1
0
-1
-2
-2
-1
0
1
2

Problem 4

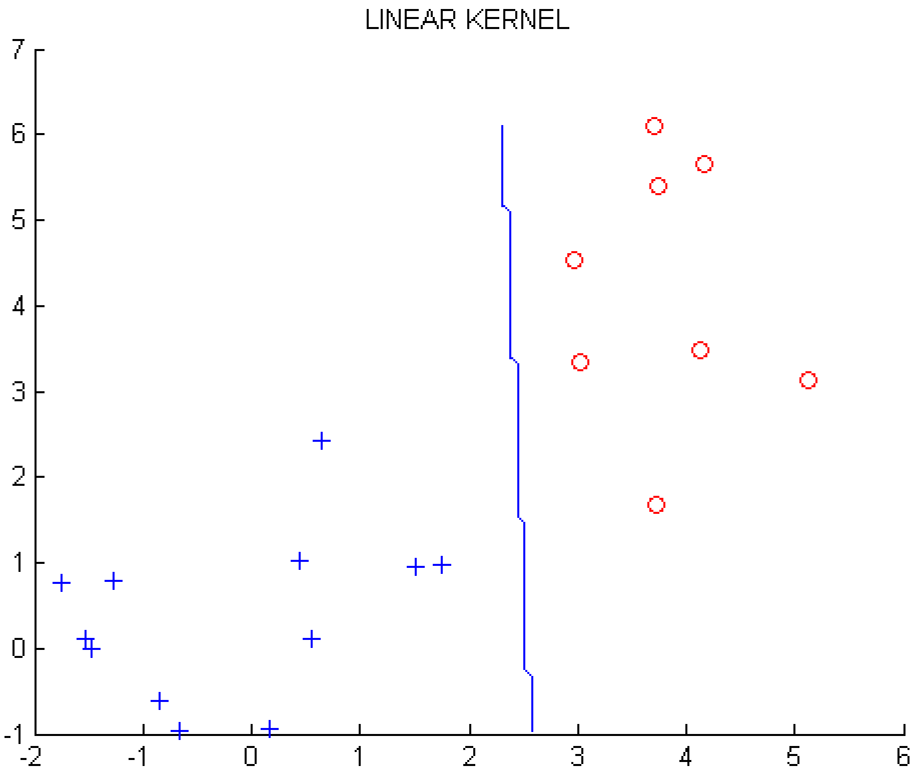
Apply SVM classification on Data 1 and regression on Data 2 (find the attached documents).
(a) Apply direct method and an iterative technique to solve the problem.
(b) Find suitable kernel using cross validation techniques.
(c) Plot the decision boundary for classification and the SVM points.
(d) Plot the function that generates the data for the regression.
(e) Plot the value of primal and dual objective function against iteration.
(f) Assess the performance of the model.

Solution

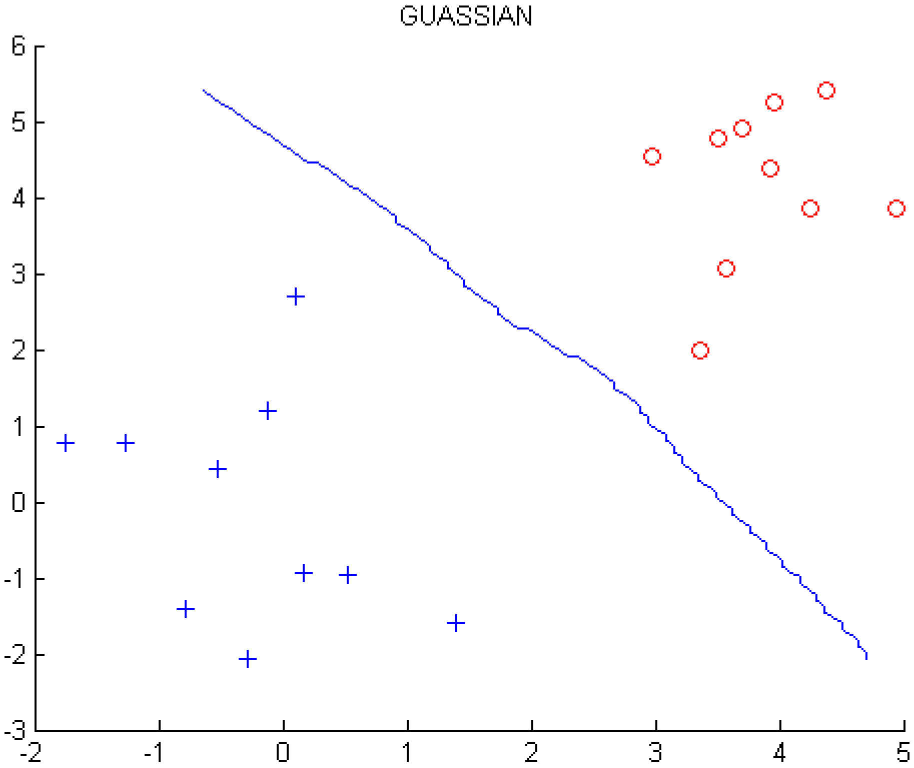
(a) **Classification**
Code for implementing SVM using Direct Method And Iterative Method of SMO was written on MATLAB and was applied to $Data_1.csv$
The data is 2 attribute 2 class classification data, with the classes being 1 and -1.
The classification was done using SVM(with direct and gradient descent algorithm used for optimization) with different kernels and the accuracy and F measure were compared..
(b)
Cross-validation was used to find the best kernel and Linear and Guassian Kernel gave the equally good result, better than polynomial kernel of degree 5.

(c)

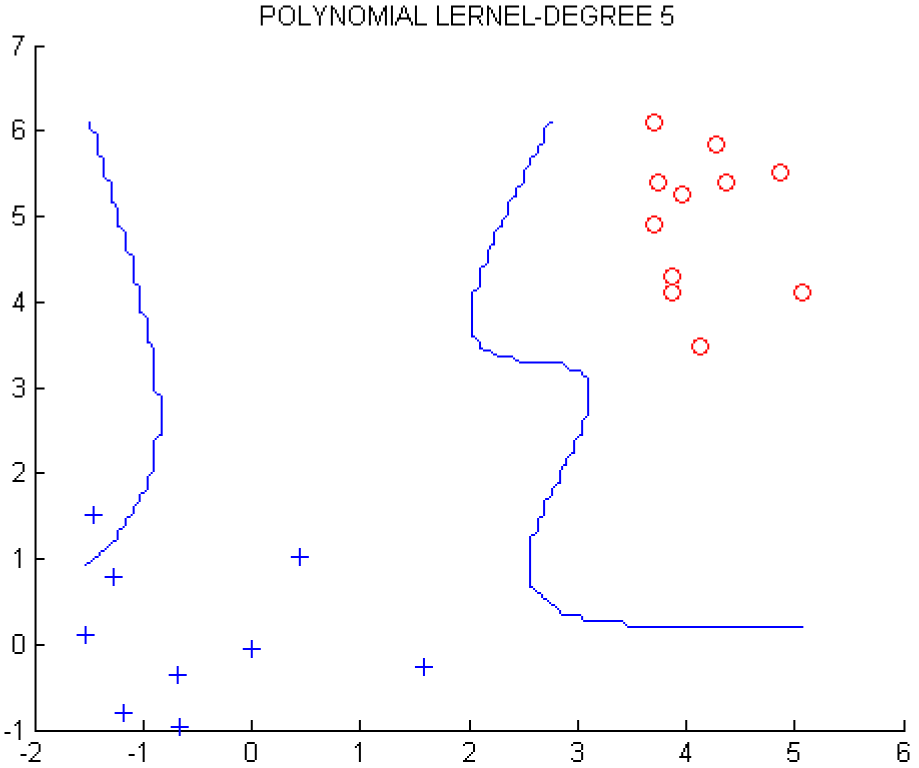
Decision Boundary-Direct Method



Decision Boundary-Direct Method

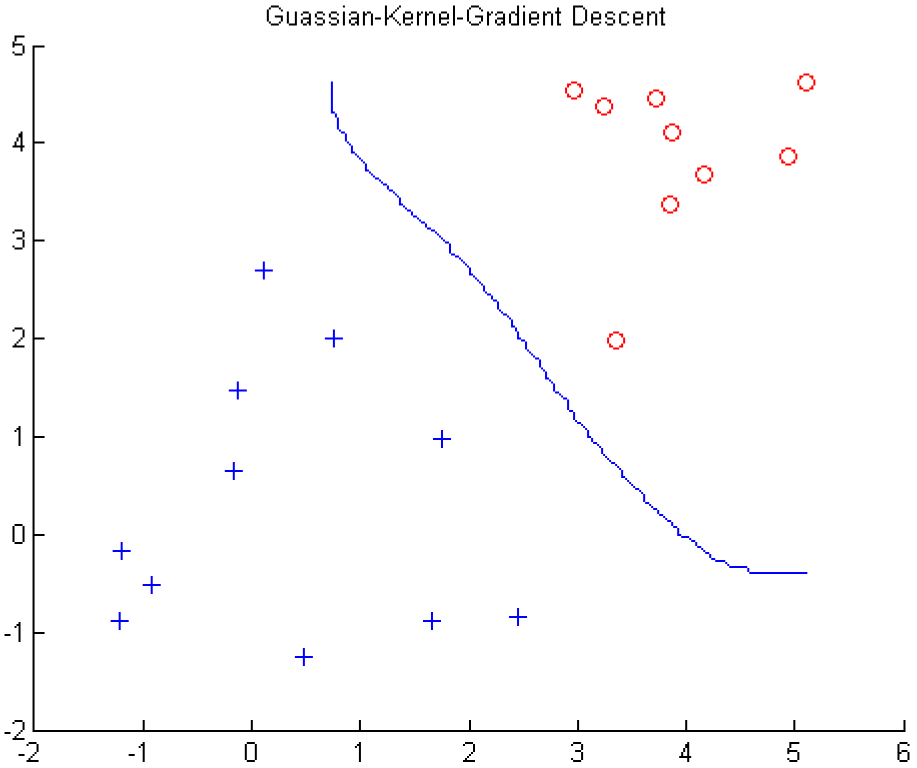


Decision Boundary-Direct Method

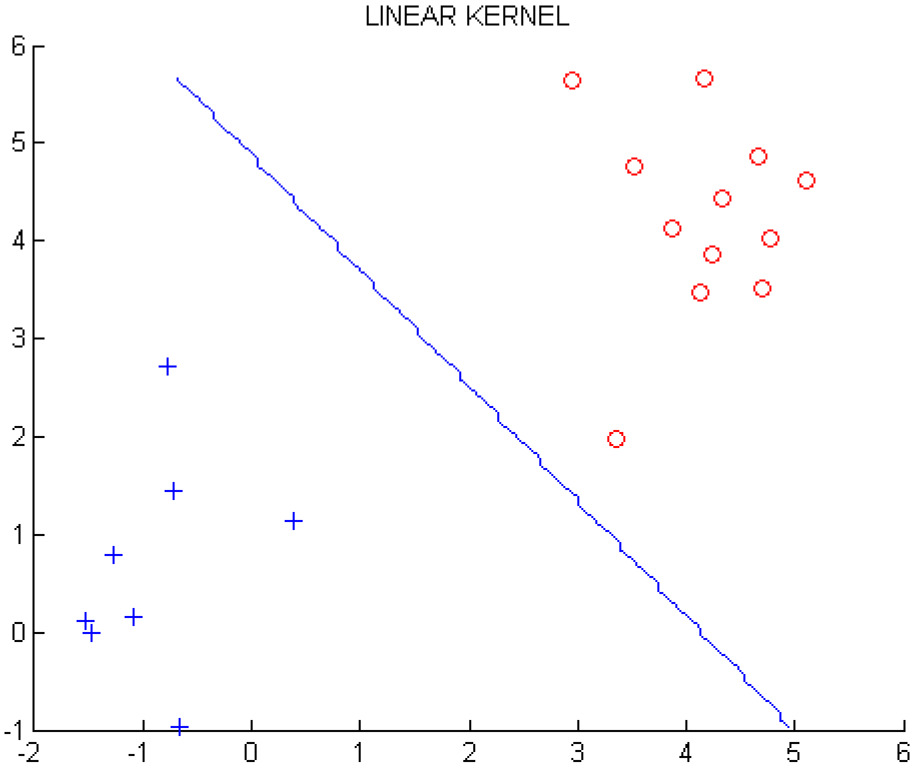


Problem 4 continued on next page. . .

Decision Boundary-Gradient Ascent Method



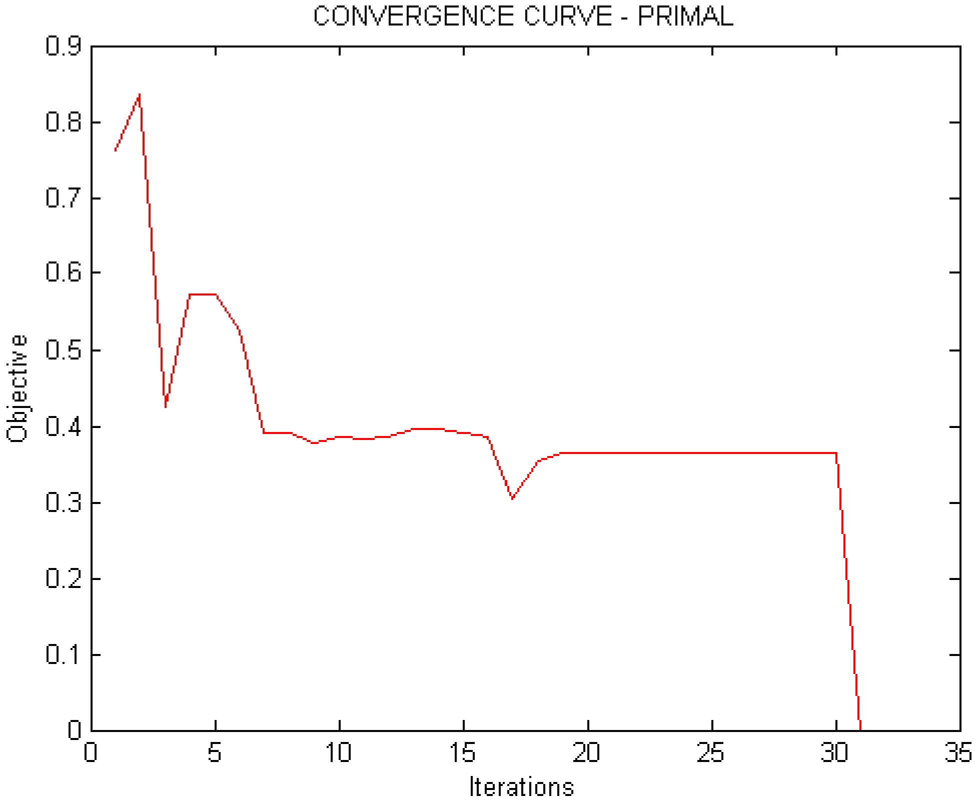
Decision Boundary-Gradient Ascent Method



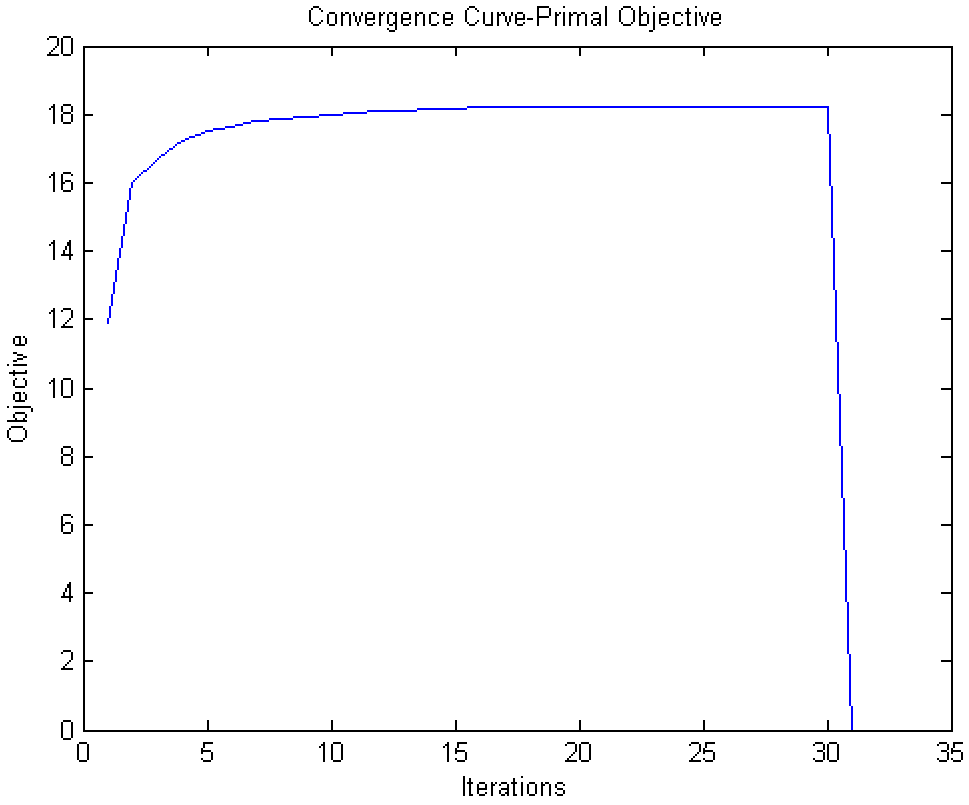
(e) Convergence Curve For Both Primal And Dual Objective

Problem 4 continued on next page. . .

Convergence-Gradient Ascent Method



Convergence-Gradient Ascent Method



Problem 4 continued on next page. . .

(f)

Analysis Of The Classification Data With Direct Method

Kernel	C value	Accuracy	F Score
Linear	.1	100	1
Guassian(Sigma=3)	.1	100	1
Polynomial(d=5)	.01	95	1

Analysis Of The Classification Data With Iterative Method

Kernel	C value	Accuracy	F Score
Linear	.1	100	1
Guassian(Sigma=3)	.1	100	1
Polynomial(d=5)	.01	95	1

Problem 5

Discuss the scalability of kernel methods.

Solution
For many applications, Support Vector Machines (and other Kernel based algorithms) alone or in combination with other methods yield superior performance to other machine learning options. In general SVMs, work very well in practice, are modular, have a small number of tunable parameters (in comparison with neural networks) and tend toward global solutions.

However, a significant disadvantage of kernel methods is the problem of scalability to a large number of data points. The problems associated with large data sets using SVMs include a drastic increase in training time, increased memory requirements and a prediction time that is proportional to the number of kernels (support vectors) which also increases with the number of data points.

In 1997 Burges and Scholkopf suggested exploiting knowledge of a problems domain to achieve better accuracy for large scale support vector machines. Their contributions of the virtual support vector method and the reduced set method boasted a SVM that was 22 times faster and yielded a better generalization performance

Despite the minimal results, the methods are based on important principles: improve accuracy by incorporating knowledge about the invariances of the problem and increase classification speed by reducing the complexity of the representation of the decision function. Problem domain knowledge can be utilized either by being used directly in the coding of an algorithm or it can be used to generate artificial (virtual) training examples. Due to high correlations between the artificial data and the larger training set size, generating virtual examples can significantly increase training time, but has the advantage of being easily used for any learning machine. Support Vector Machines can combine both approaches. Support Vector Machines have the following characteristics: if all other training data was removed and the system was retrained, the solution is unchanged, the support vectors, are close to the decision boundary, and different SVMs tend to produce the same set of support vectors. Burges and Scholkopf propose training an SVM to generate a set of support vectors, generating artificial examples by applying desired invariance transformations to the support vectors and then training another SVM on the new set.

In 2001, Tresp and Schwaighofer compared three methods designed to improve the scalability of kernel based systems. The methods examined include Subset of Representers Methods (SRM), Reduced Rank Approximation (RRA), and an improved BCM Approximation. SRM is based on a factorization of the kernel function. In the SRM method, a set of base kernels are selected (either subset of training data or of the test data) and the covariance of two points is approximated by using the covariance matrix of the base Kernel points. The Gram matrix is then approximated using these covariances. The key to this method is the reduction in rank of the Gram matrix (which will be of rank equal or less than the number of the base kernels). The Reduced Rank Approximation, RRA, uses the SRM decomposition of the Gram matrix to calculate the kernel weights. The difference between RRA and SRM is that the SRM decomposition changes the covariance structures of the kernels, while RRA leaves the covariance structures unchanged. In addition, with RRA the number of kernels with nonzero weights is identical to the number of training points, while in SRM the number is identical to the number of base points. Finally, Tresp offers his own method, the Bayesian Committee Machine (BCM) approximation, which uses an optimal projection of the data on a set of base kernels. Using assumptions about conditional independencies, Tresp calculates the optimal prediction at the base kernel points. However, this calculation requires computing the inverse of the covariance of label y given a function f of the base kernel. In order to avoid the inverse calculation of an NxN matrix, the BCM

Problem 5 continued on next page. . .

approximation uses a block diagonal approximation of the covariance matrix and the computation of the associated weight vector requires only the inversion of a block of size B. The approximation improves when few blocks are used. In fact the BCM approximation becomes SRM when the covariance of y given f is set to alpha squared times the identity.