T-61.6040 Assignment-09/2011

Santosh Tirunagari (245577)
santosh.tirunagari@aalto.fi

December 18, 2011

AIM

Implemention of soft margin SVM algorithm discussed in the lecture using Histogram intesection kernel and χ^2 kernel with γ value as 2 for classification of male and female faces from olivetti dataset for $C = \{0.01, 0.1, 1, 10, 100\}$.

Datasets

The datasets for face used in the assignment are training set and test set with 200 samples and 4096 features (64 x 64 pixel image). Class labels for training and test set are denoted as $+1$ for *male* and -1 for *female*.

Implementation

No need for zscore normalization. Just calculate histogram intersection kernel function.

```
his   = (histc(X', 0:1:255) / length(X))';
his_t   = (histc(X_t', 0:1:255) / length(X_t))';
%%
for it  = 1:1:200
    for jit  =  1:1:200

    K(it,jit) = sum(min([his(it,:);his(jit,:)]));
    end
end
```

Solve SVM dual optimization problem in standard QP form using quadprog function in matlab and find alpha values.

```
H = (Y*Y').*K;              A = [];
Aeq = Y';                   l = zeros(200,1);
c = -1*ones(200,1);         b = [];
beq = 0;                    u = C(k)*ones(200, 1);
options = optimset('Algorithm','interior-point-convex');
alpha = quadprog(H, c, A, b, Aeq, beq, l, u,[],options);
```

Compute the near boundary coefficients and move near boundary alpha values to boundaries

```
alpha(alpha < C(k) * 0.001) = 0;
alpha(alpha > C(k)*0.999) = C(k);
```

Compute value for bias parameter b

```
sv = find(alpha >0 & alpha<C(k));
sv_one = zeros(200,1);
sv_one(sv,1) = 1;
b = sv_one'*(Y-((alpha.*Y)'*K')')/sum(sv_one);
```

Compute the decision function on training set

```
temp = bsxfun(@plus,K(sv,:)'*(alpha(sv,:).*Y(sv,:)),b);
```

Compute the decision function on test set

```
his_sv = his(sv,:);
for it  = 1:1:length(sv)
for jit  =  1:1:200

    Ki(it,jit) = sum(min([his_sv(it,:);his_t(jit,:)]));
end
end
temp = bsxfun(@plus,Ki'*(alpha(sv,:).*Y(sv,:)),b);
res = temp;
```

Threshold the decisions to get proper +1 and −1

```
res(res>=0) = 1;
res(res<0) = -1;
```

Calculate the misclassification errors

```
r = sum(res~=Y_t);
r=(r/200)*100;
```

Calculate the % of support vectors utilised.

```
s = length(sv);
s=(s/200)*100;
```

Plot the graphs showing the misclassification errors.

```
hut = log10(C);
plot(hut,result_train,'k-*','LineWidth',5,'MarkerSize',10);
hold on
plot(hut,result_test,'r-o','LineWidth',5,'MarkerSize',10);
```

Plot the graph showing the % of support vectors utilised.

```
hut = log10(C);
plot(hut,support,'k-*','LineWidth',5,'MarkerSize',10);
```

The same implementation is carried out for all the values of $C = \{0.0001, 0.001, 0.01, 0.1, 1\}$ and mis-classification error and support vector count percentages are computed.
These values are given in table 1.

The implentation of chisquare kernel is the same as the implementation of histogram intersection kernel except the kernel computation. chisquare kernel is computed as

```
his   = (histc(X', 0:1:255) / length(X))';
his_t  = (histc(X_t', 0:1:255) / length(X_t))';
%%
D = zeros(size(his,1),size(his,1));
    for i=1:size(his,1)
        d = bsxfun(@minus, his, his(i,:));
        s = bsxfun(@plus, his, his(i,:));
        D(:,i) = sum(d.^2 ./ (s/2.5+eps), 2);
    end
    K = -2*D;
```

The decision function for chisquare kernel is computed as follows:

```
temp = (S(k).^2);
temp1=-2.*X(sv,:)*X_t';
temp3=sum(X_t.^2,2)';
Ki = exp(-bsxfun(@plus,sum(X(sv,:).^2,2),bsxfun(@plus,temp3,temp1))./temp);
res = bsxfun(@plus,Ki'*(alpha(sv,:).*Y(sv,:)),b);
```

The chisquare kernel implementation is carried out for all the values of $C = \{0.0001, 0.001, 0.01, 0.1, 1\}$ and mis-classification error and support vector count percentages are computed. These values are given in table 2.

Running of code for histogram intersection kernel in matlab: svm_kernel
Running of code for chisquare kernel in matlab: svm_kernel_chi

Results

Table 1 and figure 1 shows that the support vectors are decreasing as the value of C increases. The mis-classification error for the test data is minimum at $C = 10$.

Table 1: Mis-classification errors and % of support vectors utilized in implementing histogram intersection kernel

C	Mis-Classification Error on Training set	Mis-Classification Error on Test set	% of Support vectors
0.01	10.000	10.000	100.000
0.1	10.000	10.000	47.000
1	10.000	10.000	42.500
10	0.000	5.000	48.800
100	0.000	5.500	47.500

Figure 1: Plot of mis-classification errors and % of support vectors utilized in implementing histogram intersection kernel.

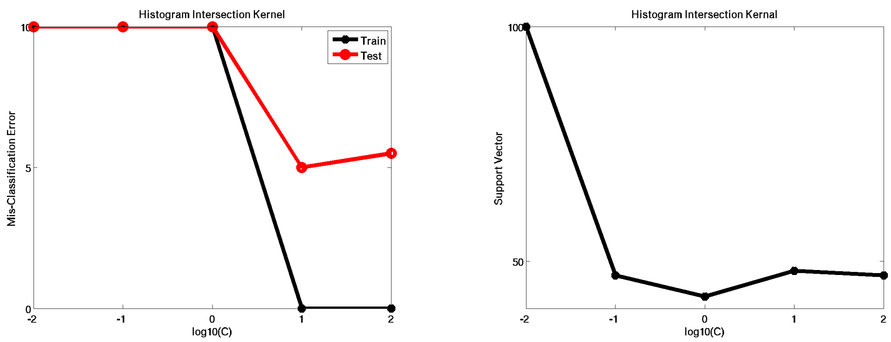
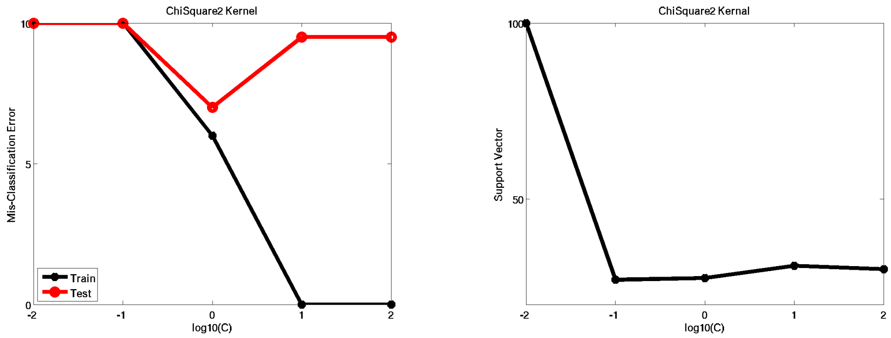


Table 2 and figure 2 shows the mis-classification error for the test data is minimum at $C = 1$.

Table 2: Mis-classification errors and % of support vectors utilized in implementing chisquare kernel

C	Mis-Classification Error on Training set	Mis-Classification Error on Test set	% of Support vectors
0.01	10.000	10.000	98.000
0.1	10.000	10.000	27.000
1	6.000	7.000	27.500
10	6.000	7.500	31.000
100	0.000	7.500	29.500

Figure 2: Plot of mis-classification errors and % of support vectors utilized in implementing chisquare kernel.



References

[2] http://en.wikipedia.org/wiki/Support_vector_machine
[3] http://www.mathworks.se/help/toolbox/optim/ug/quadprog.html

Attachments

Matlab code svm_kernal.m and svm_kernel_chi.m and datasets have been archived along with this report.