SVM & K-Means

Assignment Report

Machine Learning

To the

The University Of Texas at Dallas

Submitted by:

Prashant Prakash

(Net id: pxp141730)

The University of Texas at Dallas

Dallas – 75252

April, 2015

- **Support Vector Machine (SVM)**

Steps:

1. Download libsvm
2. We can directly use libsvm on training data to build model and predict on testdata.
3. java -cp libsvm.jar svm_train
   E:\MLAssignments\promoters_data.tar\promoters_data\training.new
   E:\MLAssignments\promoters_data.tar\promoters_data\svm.model
   The model is created as svm.model
4. java -cp libsvm.jar svm_predict
   E:\MLAssignments\promoters_data.tar\promoters_data\validation.new
   E:\MLAssignments\promoters_data.tar\promoters_data\svm.model predict.out
   This command gives the accuracy.
5. Similarly we can use other parameters to set Degree and Kernel like:
   -t kernel_type : set type of kernel function (default 2)

0 -- linear: u'*v
1 -- polynomial: (gamma*u'*v + coef0)^degree
2 -- radial basis function: exp(-gamma*|u-v|^2)
3 -- sigmoid: tanh(gamma*u'*v + coef0)

-d degree : set degree in kernel function (default 3)

Kernel	Accuracy
Linear	85.71%
Poly Degree 2	80%
Poly Degree 3	74.28%
Poly Degree 4	77.14%
Poly Degree 5	80%
Radial	77.14%
Sigmoid	45.71%

Using Perceptron

Perceptron	85.71%
Multilayer Perceptron with one hidden Unit	77.14%

Conclusion: SVM with Linear Kernel and Perceptron giving similar accuracies. It can only happen when the data set is linear. So this concludes that promoter database is linear in nature.

- ## Boosting/Bagging

  -Error Measured Used: The choice of measuring error rate can be either from accuracy but I found on web that **Relative Absolute Error (RAE)** is a good measure for error rates.

Data Set 1 : Diabetes							
# of Iterations	**30**			**100**		**150**	
Base Learner	**Vanilla**	**Bagging**	**Boosting**	**Bagging**	**Boosting**	**Bagging**	**Boosting**
J48	69.20%	68.21%	60.10%	68.19%	55.82%	68.05%	57.11%
Logistic	68.08%	68.14%	73.14%	67.99%	73.14%	68.00%	73.14%
Decision Stump	83.64%	82.84%	65.53%	83.03%	64.14%	83.10%	63.63%

Data Set 2: Inospheric							
# of Iterations	**30**			**100**		**150**	
Base Learner	**Vanilla**	**Bagging**	**Boosting**	**Bagging**	**Boosting**	**Bagging**	**Boosting**
J48	20.29%	25.24%	14.61%	24.91%	13.53%	25.15%	12.98%
Logistic	27.85%	33.24%	26.68%	32.91%	26.68%	32.81%	26.68%
Decision Stump	58.86%	60.18%	25.51%	60.2%	17.81%	60.18%	14.71%

Data Set 3: Vote							
# of Iterations	**30**			**100**		**150**	
Base Learner	**Vanilla**	**Bagging**	**Boosting**	**Bagging**	**Boosting**	**Bagging**	**Boosting**
J48	11.63%	13.89%	11.05%	13.89%	11.05%	14.01%	11.05%
Logistic	10.45%	13.17%	8.65%	12.91%	8.65%	12.92%	8.65%
Decision Stump	16.63%	16.68%	8.42%	16.61%	7.95%	16.65%	8.13%

1. Which algorithms+data set combination is improved by Bagging?
   Ans: Generally for all the data sets Bagging has increased the error rates. For Diabetes data there is a decreases but it not that significant. Diabetes with J48 and Decision stump.

2. Which algorithms+data set combination is improved by Boosting?
   Ans: Vote with J48, Logistic , Decision Stump and Inospheric with J48 and Decision Stump.

3. Can you explain these results in terms of the bias and variance of the learning algorithms applied to these domains? Are some of the learning algorithms unbiased for some of the domains? Which ones?
   Ans: Bagging reduces variance by averaging and Boosting averages and reduces bias. Decision Tree is unstable algorithm Bagging which is re- sampling increases its performance by very less margin for Diabetes dbs. Decision Stump is poor learner so with Boosting increases its performance considerably for Ionosheric and Vote dbs. But Logistic regression is stable there is no significant change with Bagging and boosting for ionospheric db but for vote db with Boosting performance is increased with some. Diabetes db seems noisy because by Boosting (which is re-weighting) performance is reduced. Decision Stump with AdaBoost seems to give best results for less noisy dbs.

- **K-Means Clustering**

For Image Koala (Original File Size: 763kb)

	Compression Ratio											
K-value	1st run	2nd run	3rd run	4th run	5th run	6th run	7th run	8th run	9th run	10th run	Average	Variance
2	5.96	6.93	7.26	5.96	5.96	6.35	5.96	5.96	6.93	5.96	6.32	0.24
5	5.01	4.92	4.43	4.95	5.01	4.95	5.01	4.76	5.01	4.95	4.9	.029
10	4.68	4.68	4.59	4.68	4.65	4.68	4.68	4.68	4.59	4.68	4.65	.0012
15	4.54	4.82	4.54	4.73	4.54	4.73	4.68	4.73	4.54	4.82	4.66	.012
20	4.48	4.54	4.85	4.85	4.48	4.52	4.48	4.54	4.48	4.52	4.57	.019

For Image Penguins (Original File Size: 760kb)

	Compression Ratio											
K-value	1st run	2nd run	3rd run	4th run	5th run	6th run	7th run	8th run	9th run	10th run	Average	Variance
2	9.04	9.12	9.04	9.04	9.14	9.19	9.04	9.04	9.04	9.12	9.08	0.0028
5	7.45	7.37	7.37	7.45	7.75	7.45	7.42	7.37	7.45	7.45	7.45	.011
10	6.60	6.72	6.72	6.60	6.66	6.55	6.60	6.60	6.60	6.55	6.62	.0034
15	6.55	6.6	6.72	6.6	6.97	6.55	6.6	6.55	6.72	6.6	6.64	.015
20	6.44	6.9	6.6	6.9	6.44	6.55	6.55	6.84	6.84	6.78	6.68	.031

All Calculations are done with keeping units in bytes. Compression ratio is defined as

inputFilesize/outputfilesize

Is there a tradeoff between image quality and degree of compression. What would be a good value of K for each of the two images?

Ans: As K increases image quality increases for both the images and image compression decreases.at k=20 image quality is good for koala but for penguins it's not that good. Out of all k values I think k=2 will be good for both the images as image quality is much better than previous values.