Özge Akin
Machine Learning for Applications in Computer Vision
Report

SVM

1. What's the objective?
   Predicting test features using SVM
2. What's the dataset?
   MNIST dataset which consist of 60000 train and 10000 to test sets
3. How did you modify it?
   The 28x28 images are flattened and shaped into vectors (1x784)
4. What method did you use and why?
   Classifiers with different parameters (kernel, gamma, degree) have been tried in order to pick a classifier which requires less computation and higher accuracy.
5. What experiments did you run?

SVM Classifier	kernel	gamma	degree	precision	recall	f1-score	Training sample
Default SVC	rbf	0.0	3	0.01	0.10	0.02	1000
Default SVC	rbf	0.0	3	0.01	0.11	0.02	10000
SVC	linear	0.0	3	0.91	0.91	0.91	"
rbf_svc1	rbf	0.05	3	0.01	0.11	0.02	"
rbf_svc2	rbf	0.7	3	0.01	0.11	0.02	"
poly_svc1	poly	0.0	3	0.96	0.96	0.96	"
poly_svc2	poly	0.0	5	0.93	0.93	0.93	"
LinearSVC	-	-	-	0.86	0.85	0.85	"

6. What are the results?

Complexity is high especially with large number of training samples such as 60000 it gives slow performance. On the other hand, it gives good results with poly kernels and bad results with rbf kernels for this dataset.

7. Comparison with the state of the art results

SVM brings huge complexity at runtime. With the state-of-the-art technique of SVM that is called additive-kernel SVM, mentioned at S. Maji and J. Malik's Handwritten Digit Classification paper http://people.cs.umass.edu/~smaji/projects/digits/[1] , it is possible to have low complexity and high accuracy. This approach has 0.79% error rate.

8. Give reasons why your method performed well / poorly

According to scikit documentation if features are very sparse, which is the case with MNIST dataset, it should be replaced by the average number of non-zero features.

Decision Tree and Random Forest

1. What's the objective?
   Predicting test features using decision trees and random forests
2. What's the dataset?
   MNIST dataset which consist of 60000 train and 10000 test sets
3. How did you modify it?
   The 28x28 images are flattened and shaped into vectors (1x784)
4. What method did you use and why?
   Classifiers with different parameters (max_depth, max_features) have been tried in order to

pick a classifier which requires less computation and higher accuracy.

5. What experiments did you run?

Decision Trees C.	criterion	max_depth	max_features	Score
Default Tree	gini	None	None (#features)	0.8764
Tree 1	entropy	15	None	0.8872
Tree 2	gini	15	None	0.8802
Tree 3	entropy	15	(28*28/2)	0.8802
Tree 4	entropy	25	(28*28/2)	0.8826

Random Forest C.	criterion	max_depth	max_features	n_estimators	Score
Default Forest	gini	None	auto (sqrt(#features))	10	0.9479
Forest 1	entropy	15	auto	10	0.9509
Forest 2	entropy	25	auto	10	0.9467
Forest 3	entropy	15	auto	20	0.9579
Forest 4	gini	15	None	20	0.9518
Forest 5	entropy	15	None	20	0.9533

Cross Validation Score with best hyperparameters:

Classifier	Mean Score
Tree 1	0.869867022192
Forest 3	0.95366664568

6. What are the results?

Random forest tree classifiers gives very high scores. For random forests, there are not much difference when changing parameters in a certain range. Therefore, it is computationally faster also relatively accurate if we select the alternative with the one that has less max_depth, max_features and n_estimators.

Decision trees performs good but not better than random forests according to output scores.

7. Comparison with the state of the art results

Thresholding a Random Forest Classifier method proposed by paper, https://www.tnt.uni-hannover.de/papers/data/1055/isvc2014_baumann.pdf , gives improvement to original random forests. Several datasets including MNIST is used and proposed method has 2.69% error rate with the same dataset.

8. Give reasons why your method performed well / poorly

It produces generic cost function that accurately predicts test sets.

Citations

1. http://people.cs.umass.edu/~smaji/projects/digits/
2. https://www.tnt.uni-hannover.de/papers/data/1055/isvc2014_baumann.pdf
3. http://scikit-learn.org/dev/documentation.html
4. http://yann.lecun.com/exdb/mnist/