

Session 2

Recognizing Handwritten Digits

Computer Vision Group
IIT Madras

December 2, 2014

- 1 Feature Extraction
 - Observing MNIST dataset
 - Saving features to a file
- 2 Training a classifier
 - Support Vector Machines
 - Implementing an SVM
- 3 Testing the classifier
 - Using the classifier on new images
 - Evaluation Measure
- 4 Scope for improvement

Observing MNIST dataset

Feature Extraction

Training set

Unzip the .zip file which has been provided to you. We'll use contents of the extracted 'training' folder to train our classifier.

Test set

Unzip the .zip provided to you. We'll use the contents of the extracted 'test' folder to test the performance of our classifier.

Observe the folders labeled 0 and 1

Each folder has images of a particular handwritten digit.

Saving features to a file

Feature Extraction

- Each image is of dimension 28×28 . (Thats $28 \times 28 = 784$ pixel values)

Saving features to a file

Feature Extraction

- Each image is of dimension 28×28 . (That's $28 \times 28 = 784$ pixel values)
- Use pixel access to save each pixel intensity to a file in the csv format.

Saving features to a file

Feature Extraction

- Each image is of dimension 28×28 . (That's $28 \times 28 = 784$ pixel values)
- Use pixel access to save each pixel intensity to a file in the csv format.

CSV format

Traverse through each image in the row-major form and store the values separated by a comma. Each line in the file will correspond to features from a single image.

Example

A CSV form of five 28×28 images will have 5 lines, with each line containing $(28 \times 28 - 1)$ commas.

Saving features to a file

Feature Extraction

- Each image is of dimension 28×28 . (Thats $28 \times 28 = 784$ pixel values)
- Use pixel access to save each pixel intensity to a file in the csv format.

CSV format

Traverse through each image in the row-major form and store the values seperated by a comma. Each line in the file will correspond to features from a single image.

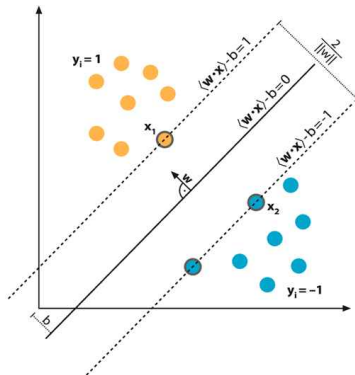
Example

A CSV form of five 28×28 images will have 5 lines, with each line containing $(28 \times 28 - 1)$ commas.

- You should now have two files, namely train.csv and test.csv which contain data corresponding to the training and test datasets.

Support Vector Machines

Training a Classifier



- They find an optimal separation between classes.
- OpenCV has inbuilt functions to implement this classifier.

Implementing an SVM

Training a classifier

Hands on

Using the classifier on new images

Testing the classifier

Hands on

Evaluation Measure - Misclassification rate

Testing the classifier

Since its a classification task (image is being classified as one of 10 digits), misclassification rate could be used as a criterion for evaluating the performance of the classifier.

Other performance measures

- Precision
- Recall

Scope for improvement

- Feature Extraction?

Scope for improvement

- Feature Extraction?
- Reducing the number of features?

Scope for improvement

- Feature Extraction?
- Reducing the number of features?
- Better classifier?

Summary

Today's session

We successfully built a handwritten gesture recognition. We also learnt about support vector machines.

Tomorrow's session

- **Segmenting out digits** using the classifier we built today.
- We used cropped images to train our classifier today. What if there are many digits in the image? How do we automatically find out the parts of the image which contain digits?