

Exploring the SVM-KNN with Vision Problems

Joon Hyuck Choi (jchoi100, jchoi100@jhu.edu)
Joo Chang Lee (jlee381, jlee381@jhu.edu)

November 20, 2016

1 Abstract

The k-Nearest Neighbor (KNN) algorithm is a non-parametric method used for classification and regression. In a previous homework assignment, we implemented the standard KNN classifier and one of its variants, the distance weighted KNN. We felt the need to take another step from this assignment and explore another type of the KNN algorithm, which is a bit more involved.

The KNN variant that we explore in this work is the SVM-KNN, a KNN algorithm that makes use of a kernel multiclass SVM as a subprocedure. The nearest neighbor approach in visual recognition problems has proven to work well in the past [1]. However, despite its benefits, because the NN approach may suffer from high variation due to finite sampling. The incorporation of an SVM can remedy this situation.

Note that the two extreme cases of the SVM-KNN are the standard KNN for small K values and the regular SVM for $K = n$. The algorithm first attempts multiclass classification using the standard KNN with a unanimous voting scheme. When at least one vote is different, it turns to the kernel multiclass SVM for prediction.

Using a SVM can be effective in the neighborhood of a small number of examples and a small number of classes [1]. We aim towards testing this algorithm on two datasets: MNIST [3] and USPS [4]. Extended goals include testing the algorithm on the Caltech 101 [5] dataset, which consists of more sophisticated real world images with more labels.

2 Methods

First run our own standard KNN with unanimous voting. In case of disagreement, use the kernel SVM from scikitlearn with our own distance metric as input. Implement on our own the kernel trick provided in [1] as necessary:

$$K(x, y) = \langle x, y \rangle = \frac{1}{2}(\langle x, x \rangle + \langle y, y \rangle - \langle x - y, x - y \rangle) = \frac{1}{2}(d(x, 0) + d(y, 0) - d(x, y))$$

Basic Algorithm [1]

For input instance,

- (1) Compute distances of the instance to all training instances and pick K nearest neighbors.
- (2) If the K neighbors have the same labels, the instance is labeled and terminate.
Else, compute the pairwise distances between the K neighbors.
- (3) Convert the distance matrix to a kernel matrix and apply multiclass SVM.
- (4) Use the resulting classifier to label the input instance.

The KNN algorithm often suffers from the problem of high variance in the case of limited sampling. Therefore, SVMs might help alleviate the high variance problem although it might consume more computational time. However, given a small neighborhood of instances and a small number of classes, SVMs may give better performance than other types of classifiers.

3 Resources

- (1) **External Libraries:** SVM algorithm (e.g. scikitlearn [2])
- (2) **Dataset:** MNIST, USPS, Caltech 101
- (3) **Programming Language:** Python 2.7

4 Milestones

4.1 Must achieve

- (1) Create SVM-KNN by using our own KNN (modify from previous homework assignment to fit our datasets) and incorporating scikitlearn's SVM.
- (2) Write kernel trick function mentioned in the abstract above.
- (3) Parse MNIST and USPS datasets to fit as input to our SVM-KNN implementation.
- (4) Run SVM-KNN, KNN, and SVM on MNIST and USPS using L_2 distance function. (Try different K values for SVM-KNN and KNN)
- (5) Compare and analyze results from (4).

4.2 Expected to achieve

- (1) Use shape context instead of L_2 to compute distance matrix to feed to Kernel SVM when running SVM-KNN on MNIST.
- (2) Use tangent distance instead of L_2 to compute distance matrix to feed to Kernel SVM when running SVM-KNN on USPS.

4.3 Would like to achieve

- (1) Experiment SVM-KNN on Caltech 101 dataset.
- (2) Reasons Caltech 101 dataset is more involved:
 - a) Caltech 101 consists of real world images instead of digits (MNIST, USPS). Color (not gray-scale as in MNIST and USPS), pose, and lighting makes things more difficult.
 - b) More sophisticated distance operations per [1].
 - c) Tests on Caltech 101 requires tests on two different distance functions [1].
 - d) More labels (101 of them) in Caltech 101.

5 Final Writeup

- (1) Motivation
- (2) Datasets
- (3) Algorithm
- (4) Experiment
- (5) Results
- (6) Conclusion

6 Bibliography

- (1) H. Zhang, A. Berg, M. Maire, and J. Malik. "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition", CVPR, 2006.
- (2) <http://scikit-learn.org/stable/modules/svm.html>
- (3) <http://yann.lecun.com/exdb/mnist/>
- (4) <https://www.otexts.org/1577>
- (5) https://www.vision.caltech.edu/Image_Datasets/Caltech101/