OMP-BASED DATA TRAINING FOR OCR

Steven Feng

@Rose-Hulman Institute of Technology

ABSTRACT

The paper mainly discusses how a dictionary-based approach in conjunction with sparsity-based linear algebra methods can be used in classification problems in machine learning. It introduces the rationale behind the algorithm and methods to improve the dictionary with examples, including real image data of hand-written digits collected from MNIST.

INTRODUCTION

In USPS, it has thousands of mails with handwritings on it to sort every day there that people cannot easily sort by hand. There is a solution that by using sorting machines with OCR technique, it saves either labor or time cost – estimated to be 600 million dollars per year for them. However, the algorithms they used need harsh hardware support. The question is: can we come up with a method that apply the same technique on a small machine (maybe a "tiny computer" in some vending machines) that does not rely on a cloud server, at the same time, keep a high accuracy when doing work of number recognition?

OBJECTIVE & INTENTION

My intention with this thesis was to write an explication on an intriguing subject. I chose machine learning as my subject due to its benefits to contemporary society and my passion in the field as I am doubling with a computer science major. My ultimate goal with this thesis was to possibly provide someone with a strong mathematical foundation and some expertise with MATLAB with the tools necessary to rapidly begin working on classification challenges in real life.

PROBLEM FORMULATION

We chose to use the MNIST data as the database, since it provides large database for training.

We use symbols to present the problem and help establish the model. An image will be represented as a vector $\vec{d}$. The dictionary we will use is a matrix D, which contains trained samples for every classes of the target. And we want to know which class will vector $\vec{d}$ belongs to. In the USPS example or even the vending machine, we care about which digit it is when we have an image for a single number, which serves as the specific case for the classification problem. Besides the result, how to have a good dictionary D concern.

Mathematically, we set up the equation D$x = \vec{d}$ to solve. To solve all the image vectors in a matrix X collectively, we will set up DA $= X$, where A has multiple sparse solution vectors like x, and X has multiple data vectors like $\vec{d}$, correspondingly.

A TINY EXAMPLE FOR ILLUSTRATION

Suppose we have a very simple dictionary D with two classes A, B with one instance for each only, where $A = \begin{bmatrix}1\\0\end{bmatrix}$ and $B = \begin{bmatrix}0\\1\end{bmatrix}$. Then $D = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$. Assume we get the vector data $\vec{d} = \begin{bmatrix}2.20\\0.07\end{bmatrix}$ from scanning an instance. We can setup the equation: D$x = \vec{d} \Rightarrow \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} x = \begin{bmatrix}2.20\\0.07\end{bmatrix}$. The exact solution is $x = \begin{bmatrix}2.20\\0.07\end{bmatrix}$, but we want it to be sparse, then $x = \begin{bmatrix}2.20\\0\end{bmatrix}$. Since x has the greatest value in the row index for the class A, we determine that the instance belongs to class A.

METHODOLOGY WITH STEPS

The outline below illustrates the general method we apply to the problem and train a successful dictionary for use:

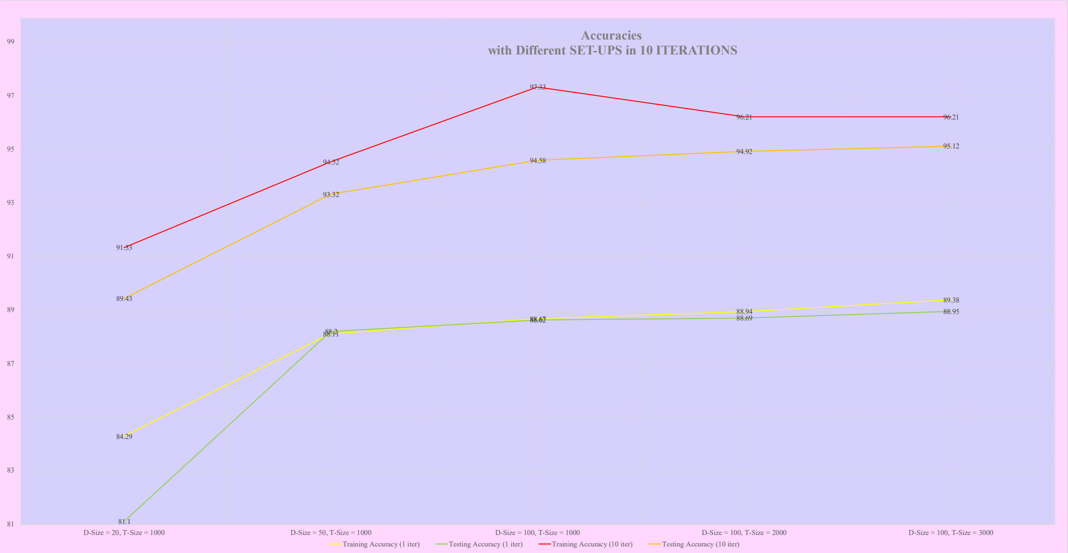
1. Use a basic dictionary D generated from anywhere and a training data matrix d to get started. Both should have attached class labels.
2. Find the sparse solution for each di to Dx_i by OMP algorithm, and construct A as the matrix with $\delta_{ci}(x_i)$ for each i^{th} column. (Note: $\delta_{ci}(x_i)$ is an operator that maps x to another vector by zeroing out all components of x except for those are in class $ci.$)
3. Minimize $\|\mathrm{D}x \; - d\|_{fro}^2$ with respect to D and get the dictionary improved once.
4. Repeat step 2-3 until any iteration limit or tolerance is reached.
5. Finally, verify the dictionary with testing data and check the validation error / accuracy.

In the experiment, we have 5 different setups to examine our method in 10 iterations.

With the increment in either dictionary size or training data size, the total elapsed time goes from 60.79s to 138.64s.

The vertical values below are represented in percentage (%).

TEST RESULTS WITH MNIST DATA

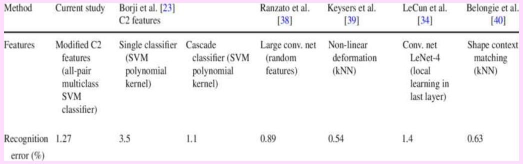


OBSERVATIONS

1. Greater dictionary size produces higher accuracies.
2. Greater training data size produces higher accuracies.
3. Dictionary size has larger impact on the accuracies compared to training data size.

Generally, the method draws up the accuracies of a default dictionary from ~10% to ~85% first by applying OMP to get sparse solutions stored, and then get it optimized by both classification approach and gradient approach we explored. It will finally get improved by ~10% than using OMP only.

COMPARATION & CONCLUSION



Method	Current study	Borji et al. [23] C2 features		Ranzato et al. [38]	Keysers et al. [39]	LeCun et al. [34]	Belongie et al. [40]
Features	Modified C2 features (all-pair multiclass SVM classifier)	Single classifier (SVM polynomial kernel)	Cascade classifier (SVM polynomial kernel)	Large conv. net (random features)	Non-linear deformation (kNN)	Conv. net LeNet-4 (local learning in last layer)	Shape context matching (kNN)
Recognition error (%)	1.27	3.5	1.1	0.89	0.54	1.4	0.63

As the figure above indicated, this OMP method alone does not show high competency in efficiency or recognition rate. However, it does have hardware barrier or device limitation for model training such as KNN and neural networking, and will provide convenient and responsive service to users in real life, apart from getting connected to the server (such as a vending machine). And another reason could be that I tested with 10 iterations beginning with sample size of 100 only since I am using a computer, which significantly reduce the success rate of our method also.

Overall, the result is satisfactory with my small machine that runs the improved and deliberated OMP algorithm.

ACKNOWLEDGEMENT & CONTACT

For details or questions, please contact at the email: fengr@rose-hulman.edu. Or visit our work at the GitHub: https://github.com/rhit-fengr/OMP-Based-Data-Training

REFERENCES

[1] Kurt Bryan, verbal and email communications.

[2] J. Brownlee, 4 Types of Classification Tasks in Machine Learning, Retrieved from: https://machinelearningmastery.com/types-of-classification-in-machinelearning.

[3] Kassidy Kelley, Top 9 types of machine learning algorithms, with cheat sheet, Retrieved from: https://www.techtarget.com/searchenterpriseai/feature/5-types-of-machine-learning-algorithms-you-should-know.

[4] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Introduction to Data Mining, Pearson Education India, 2016.

[5] Hamidi, Mandana & Borji, Ali. Invariance analysis of modified C2 features: Case study-handwritten digit recognition. Mach. Vis. Appl.. 21. 969-979. 10.1007/s00138-009-0216-9.

[6] Wright, John, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. IEEE transactions on pattern analysis and machine intelligence 31, no. 2 (2008): 210-227.

[7] Alcin, Omer F., Abdulkadir Sengur, Jiang Qian, and Melih C. Ince. OMP-ELM: orthogonal matching pursuit-based extreme learning machine for regression. Journal of Intelligent Systems 24, no. 1 (2015): 135-143.

[8] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. THE MNIST DATABASE of handwritten digits. Retrieved from: http://yann.lecun.com/exdb/mnist/.