CS229 Project: Particle classification

Kahye Song
kahye@stanford.edu

December 7, 2010

1 Introduction

In electron tomography, averaging multiple particles of the macromolecule of interest is an essential tool to achieve high resolution reconstruction. A particle means one measurement of the object of interest in structural biology. Averaging requires aligning and classifying particles to obtain the highest resolution [1]. These tasks are very challenging because of low SNR(below 1) and missing frequency information(30 % along Z-axis) due to the image acquisition restrictions. In Fig. 1, we can clearly see the differences between two different layering in the averaged reconstructions(left column). However, in a single particle before averaging(right column), we can hardly see any structure at all due to low SNR. This makes the classification task challenging for humans do with naked eyes. It took 1688 and 1104 particles to create averaged single and double S-layer pore respectively.

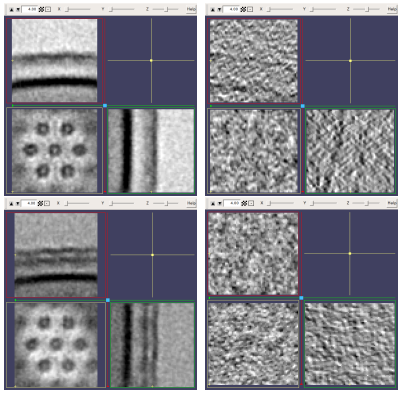

Figure 1: XY,YZ and ZX cross-sections of averaged(upper left) and single particle(before averaged, upper right) single S-layer structure. Cross-sections of averaged(bottom left) and single particle(before averaged, bottom right) double S-layer structure

Also, the size of volumetric feature, N^3, is prohibiting when processing 1000 or more particles. In this project, we are focused on 1. evaluating a supervised classifier, SVM and a unsupervised one, k-means and 2. finding low dimensional features of S-layer of bacterial cells which have two categories, single layer and double layer. The classifiers and features are going to be evaluated jointly according to the classification accuracy and training and prediction time.

2 Classification methods

2.1 Support vector machine(SVM)

Given a set of instance-label pairs (x_i, y_i), $i = 1, \cdots, m$, $x_i \in R^n$, $y_i \in [1, +1]$, solving the following optimization problem in Eq. 1 provides the maximum margin classifier

$$\min_w \frac{1}{2} w^T w + C \sum_{i=1}^{m} max(1 - y_i w^T x_i, 0)^2 \tag{1}$$

where $C > 0$ is a penalty parameter. This is the exact formulation implemented in LIBLINEAR SVM library [4]. In our particular problem, the double layer particles are labeled 1 and the single layers are labeled -1.

2.2 K-means

Given a training set $X = x_1, \cdots, x_m,\ x_i \in R^n$ but not the associated labels y_i, k-means clustering algorithm aims to partition these observations into k sets ($k \leq m$),with associated centroids $\mu = \{\mu_1, \mu_2, \cdots, \mu_k\}$ so as to minimize the sum of within-cluster distances in Eq. 2 [3].

$$J(X, \mu) = \sum_{i=1}^{m} d(x_i, \mu_{c_i}) \tag{2}$$

Here $d(\cdot)$ is a distance metric function in R^n and c_i is the index of a partition which an observation x_i belongs to. The algorithm starts with a random set of centroids and divide the observations into partitions which minimize $J(X, \mu)$ in Eq. 2. Then it updates the centroids as the mean of the partitioned observations. The algorithm typically uses Euclidean distance as a distance metric. However, it can be generalized to use different distance metrics with proper centroid update equations. In this project, *normalized correlation* is used as a distance metric. It is defined as $d_{nc}(x, y) = 1 - \frac{x^T y}{||x||||y||}$, where $||\cdot||$ is L2 norm. This is equivalent to Euclidean distance if all data points are normalized. The normalized correlation is scale invariant and it is more robust to distinguish data points which are only differ in a subset of dimensions, especially in low SNR. This is well known phenomenon in high dimensional clustering problem [2]. The double Slayer and the single Slayer do resemble each other a lot and it is mostly distinguishable along z-axis. Therefore the normalized correlation metric did better job in clustering particles than Euclidean distance.

3 Feature selection

Four features listed below are tested on both SVM and K-means classifiers.

1. Volume - This is a raw observation of a hexagonal Slayer structure. It is in $R^{65\times65\times65}$.

2. 3D FFT of volume - This is a set of 3D FFT coefficients of a raw volume. It is in $C^{65\times65\times65}$.

3. Collapsed volume - This is a sum of a raw volume over x and y axis where the double layer is not visible. It is in R^{65}.

4. FFT of Collapsed volume - This is a set of 1D FFT coefficients of a collapsed volume. It is in C^{65}.

I tried to use PCA to reduce the original feature space which is the original volume space. Due to the large dimension($R^{65\times65\times65}$) of the data points, it cannot run in Matlab. I did not tried to use PCA on the collapsed volume since it is already in a manageable low dimension such that it does not really justify running PCA to reduce the feature space further. Also, a study suggests that classifying principal components does not necessarily improve the classification accuracy [5].

4 Data and Software

422 particles of single S-layer and 276 particles of double S-layer are randomly mixed and split into 10 groups of approximately equal number(70) of particles. The ratio between the single and double layer particles in each group is kept equal. To evaluate how the number of training data impacts the classification accuracy, a sequence of k-fold cross validation has been performed. The number of training data starts from using only 1 group out of 10 and increases upto 9 groups. The rest of the data is used for testing. For a given number of training data, 10 experiments are carried out by selecting different training data groups. The size of each particle is $65 \times 65 \times 65$ voxels.

I have used LIBLINEAR SVM library(`http://www.csie.ntu.edu.tw/~cjlin/liblinear/`) used in problem set 2 for SVM and KMEANS provided in Matlab. The distance metric used for K-means is 'correlation'.

5 Performance summary and discussion

5.1 Prediction accuracy

There are many metrics to evaluate a classifier. One of the widely used metrics is a precision-recall curve. *Precision* is defined as a ratio between the number of true positives and both true and false positives. *Recall* is defined as a ratio between the number of true positives and the number of both true positives and false negatives. To evaluate the prediction performance of a classifier, 10-fold cross validation is performed on 698 data points. All eight combinations between four features and two classifiers are tested on the same training and testing data sets. Precision and recall in Fig 2. are the average of the values calculated on the testing data set labels given by the classifier trained on the training data set of all 10 trials.

Most of the classifier-feature combinations performed well achieving more than 90 % on both precision and recall. K-means in general tends to label fewer positives achieving better precision than recall. SVM tends to be more aggressive on labeling positives achieving better recall than precision. When we look at the overall *accuracy*, which is defined as the ratio between the number of both true positives and true negatives and the total data size, SVM outperforms K-means on classifying any features tested. See Fig. 2. This is an expected result since supervised learning can utilize correct labels of training data points. The performance of K-means is especially worse than that of SVM when there are not many training data available. Classifying the FFT coefficients of the raw volume or the collapsed volume with K-means did not provide satisfying result. The FFT coefficients of collapsed volume were not very easy to distinguish between the single and double layers and K-means failed even to start. Therefore the data point is missing for the 1D FFT of collapsed volume and K-means combination.

5.2 Training efficiency

Training efficiency is a metric to evaluate how many training data points are needed to attain certain level of prediction accuracy. The prediction accuracy is the ratio between the number of correct labels given by a classifier trained on a training data set and the size of the testing data set. Here the plotted accuracy value is an average of 10 trials on different training data set with the same size. The testing data set of each trial is the rest of the data points which are not used for training.

As expected, SVM utilized the training data more efficiently achieving better accuracy than K-means on classifying each feature. See Fig. 2. It is noticeable that classifying collapsed volume with both SVM and K-means have high training data efficiency. The accuracy of classifying any size of training data remains consistently high around 80 % or above. However classifying whole volume with very small training data set suffers the accuracy. This can be explained by the increased SNR by integrating over the less relevant dimension of the volume. Therefore the classifiers are not as confused as the raw volume when there are not many training data points available.

5.3 Computational efficiency

It is also crucial to have reasonably short training time when there can be 1000 or more particles to classify. As predicted by the size of each feature, classifying the collapsed volume or its FFT coefficients using both SVM and K-means took a lot shorter training and testing time. The increase is linear on the dimension of the features and square root of the training data size. See Fig. 2. Note that this observation is only partial to these particular implementations.

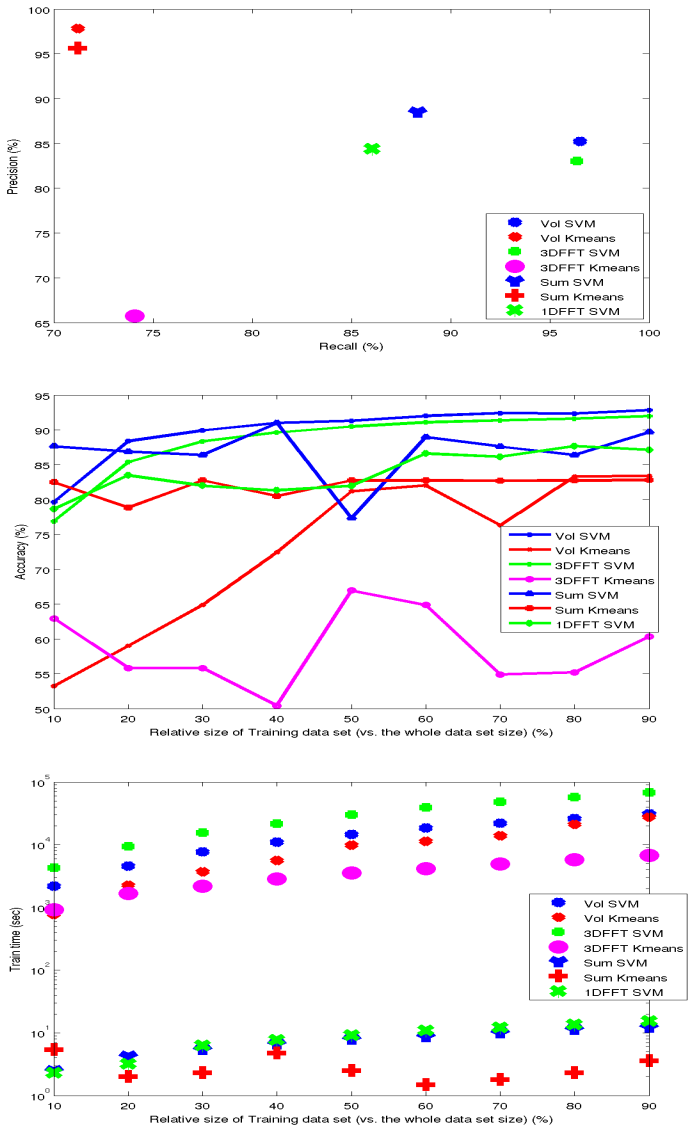

Figure 2: From top to bottom, 1. Precision - Recall 2. Training efficiency(training data set size VS. Accuracy) 3. Computational efficiency(training data set size VS. train time)

Features	Dimension (# of real numbers)	Precision (%)	Recall (%)	Accuracy (%)	Train Time (Sec)
Volume	274625	96.5270	85.2116	97.8682	309.9760
FFT3	549250	96.3847	83.0159	97.8627	676.8210
Collapsed volume	65	88.3478	88.5185	90.5205	0.1250
FFT	130	86.0456	84.4180	88.8649	0.1540

Table 1: Summary of the classification performance of all features using SVM. Train data size = 90 % of the total.

Features	Dimension (# of real numbers)	Precision (%)	Recall (%)	Accuracy (%)	Train Time (Sec)
Volume	274625	71.2247	97.8042	73.9258	275.1140
FFT3	549250	74.0656	65.7275	56.5947	67.2720
Collapsed volume	65	71.2244	95.6481	74.4020	0.036
FFT	Failed to	distinguish.	Failed	at	the first run.

Table 2: Summary of the classification performance of all features using K-means. Train data size = 90 % of the total.

6 Conclusions

Classifying noisy volume data is a challenging problem for humans because of very low SNR and large data dimension. However it turned out that by exploiting some prior information, we can achieve high classification accuracy with very short training time compared to classifying the volume itself. Here, it is known that one distinct difference between single and double Slayer is that these layers are along z-axis which is the direction normal to the surface plane. The hexagonal shapes on the surface plane are not exactly the same between these two classes of layers but this is more subtle to detect than the number of layers. Therefore summing over the XY plane which does not provide much information can speed up the procedure without sacrificing too much accuracy.

Both classification methods can classify the particles reasonably well. However, if we can have some learned data points, SVM can classify about 10-20 % more accurately than K-means. However, in reality it is almost impossible to tag single particle by naked eyes. Therefore using K-means to classify the particles can be a good start. Once some data points can be tagged, we can switch to SVM to refine the classification. The number of training data does not seem to impact the accuracy too much for both SVM and K-means one it is more than 20 % of the total data set or more.

References

[1] F. Moussavi-J. Smit K. H. Downing M. Horowitz F. Amat, L. R. Comolli. Subtomogram alignment by adaptive fourier coefficient thresholding. *Journal of Structural Biology*, 171:1047–8477, 2010.

[2] Peer; Zimek Arthur Kriegel, Hans-Peter; Krger. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3:1–158, 2009.

[3] Andrew Ng. Cs229 lecture notes on the k-means clustering algorithm. 2010.

[4] C.-J. Hsieh X.-R. Wang R.-E. Fan, K.-W. Chang and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[5] K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17:763–774, 2001.