Recognition of Different Datasets Using PCA, LDA, and Various Classifiers

Nazila Panahi
Department of Electrical Engineering
Urmia University
Urmia, Iran
st_n.panahi@urmia.ac.ir

Mahrokh G. Shayesteh
Department of Electrical Engineering
Urmia University
Urmia, Iran
m.shayesteh@urmia.ac.ir

Sara Mihandoost
Department of Electrical Engineering
Urmia University
Urmia, Iran
st_s.mihandoost@urmia.ac.ir

Behrooz Zali Varghahan
Department of Electrical Engineering
Urmia University
Urmia, Iran
st_b.zali@urmia.ac.ir

Abstract—**Bayesian, *k*-nearest neighbor, and Parzen window classifiers along with PCA and LDA methods, are effective tools in machine learning. In this work, a hybrid method is formed by the above mentioned methods. The aim is to achieve a successful, fast, and low computational classification. Performance of the new method is evaluated on five various kinds of datasets, from UCI machine learning datasets, including Breast Cancer, Iris, Glass, Yeast, and Wine. The experimental results indicate the superior performance of the proposed method in comparison with the previous works.**

Keywords- classification, PCA, LDA, Bayesian, *k*-nearest neighbor, Parzen window

I. INTRODUCTION

In machine learning, the aim is to realize the human learning job by computers. Various methods and algorithms form the base of machine learning. *k*-nearest neighbor, Bayesian, and Parzen window methods are commonly used classification algorithms. The main goal is to find out the class of new data when the information about the classes of past data is given. This process is named as classifying [1].

In the *k*-nearest neighbor method, a constant *k* value is selected. Then, the unknown sample is assigned to the most similar class from the *k*-nearest neighbors. Bayesian method [2] is a simple and high computational efficiency algorithm, because the attributes are assumed conditionally mutually independent given the class label. Despite this unrealistic assumption, Bayesian has shown surprisingly good performance in lots of domains [3]. In Parzen window method, a constant hypercube is selected. Then, the class of given data is determined using a number of learned data into the hypercube.

In this study, we propose a non-complex and fast method to improve the classifier performance. Our method has three main steps. In the first step, the dataset is preprocessed by principle component analysis (PCA). Then, data is processed with linear discriminant analysis (LDA), to eliminate the redundancy information, increase between class distances, and decrease within class distances. In the last step, the achieved dataset is classified by one of the Bayesian, *k*-nearest neighbor, or Parzen window classifiers. The experimental results indicate the superior performance of the proposed method in comparison with the traditional classifiers.

The rest of the paper is organized as follows. In section 2, normalization, PCA, LDA, and three different classifiers are briefly explained. In section 3, we describe the proposed method. In section 4, the experimental results are presented. Finally, section 5 concludes the paper.

II. PRELIMINARIES

A. Normalization

Some features in dataset, are large numerically but essentially not important in the classification result, and some others are important in classification result but small numerically. Large value features affect the correct classification rate (CCR) more than the important ones but small numerically. To have a fair comparison within all features, normalization procedure maps all features into a specific range [-1, 1], to have zero mean and unit variance. In this way, they will have the same effect on CCR. The normalization is performed as

$$\tilde{X}_{mn} = 2[(X_{mn} - X_{n,\min}) / (X_{n,\max} - X_{n,\min})] - 1 \qquad (1)$$

where $\tilde{X}_{mn}$ is the normalized element, X_{mn} is the *m*th element of the *n*th feature vector of dataset for all classes, $X_{n,\max}$ and $X_{n,\min}$ denote the maximum and minimum values of the feature vector, respectively.

B. Principal Component Analysis (PCA)

PCA [9] is a widely used technique for dimensionality reduction, feature extraction, and data visualization. PCA can be defined as the orthogonal projection of the data into a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized. Equivalently, it can be defined as the linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections.

978-1-61284-832-7/11/$26.00 ©2011 IEEE

The eigenvectors compose the transformation matrix. The eigenvectors corresponding to the directions of principal components of the original data and their statistical significance are given by their corresponding eigenvalues. To have dimensionality reduction, only a small part of the eigenvectors is kept.

Assuming N numbers of n-dimensional vectors, X_i, i= 1, 2, …, N. Firstly, we compute the mean as

$$\mu_X = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{2}$$

and the covariance matrix as

$$\sum X = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mu_X)(X_i - \mu_X)^T \tag{3}$$

Then, we compute the eigenvectors $\underline{\Phi_i}$ and eigenvalues λ_i of the covariance matrix according to the equation

$$\sum \underline{X}.\underline{\Phi}_i = \lambda_i.\underline{\Phi}_i \tag{4}$$

By ordering the eigenvalues, we remain the most significant m eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq \ldots \geq \lambda_N \tag{5}$$

The transform matrix related to the selected m eigenvectors is obtained as

$$KLT = (\Phi_1\Phi_2 \ldots \Phi_m)^T \tag{6}$$

C. Linear Discriminant Analysis (LDA)

The main idea of LDA [10] is: it searches for the project axes on which the data points of different classes are far from each other and the data points of the same class to be close to each other. For a set of N labeled samples $x_1, x_2, \ldots, x_N$, in R^n, which belongs to c different classes, The objective function of LDA (w^*) is obtained as follows

$$w^* = \arg\max_{w} \frac{w^T S_b w}{w^T S_w w} \tag{7}$$

where

$$S_b = \sum_{k=1}^{C} N_k (\mu_k - \mu)(\mu_k - \mu)^T \tag{8}$$

$$S_w = \sum_{k=1}^{C}\Big(\sum_{i=1}^{N_k}(x_{i,k} - \mu_k)(x_{i,k} - \mu_k)^T\Big) \tag{9}$$

where N_k is the number of samples in the k-th class, μ is the total mean or center vector, μ_k is the average vector of the k-th class samples, and $x_{i,k}$ is the i-th sample of the k-th class. S_w and S_b are the within-class scatter matrix and the between-class scatter matrix, respectively.

Also, if we define [11] the total scatter matrix as

$$S_t = \sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^T \tag{10}$$

then $S_t = S_b + S_w$ holds. Thus, the objective function of LDA in (7) will be equivalent to

$$w^* = \arg\max \frac{w^T S_b w}{w^T S_t w} \tag{11}$$

The optimal $\boldsymbol{w}$'s are the eigenvectors corresponding to the non-zero eigenvalue:

$$S_b w = \lambda S_t w \tag{12}$$

Furthermore, without loss of generality, we can assume $\mu = \mathbf{0}$ ($\mathbf{0}$ is zero vector). Then

$$\begin{aligned} S_b &= \sum_{k=1}^{C} N_k \mu_k \mu_k^T \\ &= \sum_{k=1}^{C} X_k P_k X_k^T = XPX^T \end{aligned} \tag{13}$$

where P_k is a $N_k \times N_k$ matrix with all the elements equal to $1/N_k$, $X_k = [\, x_{1,k}, \ldots, x_{Nk,k} \,]$ is the data matrix of the k-th class, $X = [\, X_1, \ldots, X_C \,]$, and P is a $N \times N$ matrix as follows

$$P_{N\times N} = \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_c \end{bmatrix}$$

Here, the objective function in (11) can be represented as

$$w^* = \arg\max \frac{w^T XPX^T w}{w^T XX^T w} \tag{14}$$

D. Bayesian Classifier

Bayes theorem [12] is an effective and simple method which is used frequently in classification problems. In machine learning, determining the best hypothesis from some space H, given the observed training data D is of interest. Bayes theorem [13] provides a way to calculate the posterior probability $P(h|D)$, from the prior probability $P(h)$, together with $P(D)$ and $P(D|h)$ as:

$$P(h\,|\,D) = \frac{P(D\,|\,h)P(h)}{P(D)} \tag{15}$$

where $P(D)$ and $P(D|h)$ denote the prior and the posterior probability of the observed training data D, respectively. Bayesian methods are based on probability calculations. Expectation maximization algorithm is one of them. It is utilized for unknown values with known probability distribution. Radial basis function is a popular function for explaining probability distributions which is given by

$$Y = e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{16}$$

where x is the training data, μ and σ are the mean and variance, respectively. One way to have an expectation maximization algorithm is to guess the mean of Gaussian functions [12]. Let's have a dataset with *c* different classes. Therefore, the dataset is formed from a probability distribution which consists of *c* different normal distributions. Each sample is processed in two steps. At the first step, a random normal distribution is chosen from the *c* normal distributions. At the second step, a sample data is formed according to the selected distribution. At last, the largest probability distribution is selected for each data. These steps repeat for all of points in the dataset (Fig. 1).

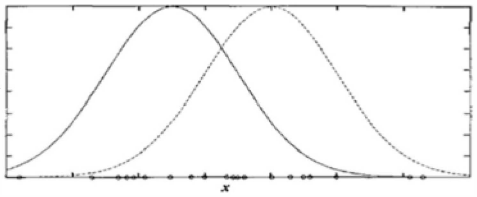

Fig. 1. Normal distribution for k=2, sample data are on the x-axis [13].

E. k-Nearest Neighbor Classifier (KNN)

k-nearest neighbor [12] classification is based on the supervised learning. The aim is to find the nearest *k* samples from the existing training data when a new sample appears and classify the appeared sample according to most similar class [13]. Generally, similarity is measured with Euclidean distance. An arbitrary sample x^k is described by the feature vector as

$$< x_1^k, x_2^k, \ldots, x_N^k >$$

where x_n^k is the value of *n*-th feature of sample x^k. Then, the distance between two samples x^i and x^j is defined as

$$d(x^i, x^j) = \sqrt{\sum_{k=1}^{n} x_k^i - x_k^j} \qquad (17)$$

The unknown sample is assigned to the most similar class from the *k* nearest neighbors [14]. In general, the following steps are done in this algorithm [14]:

- Selection for *k*: If is completely up to the user. Generally, after some trials a *k* value which gives the best result is chosen.
- Distance calculation: any distance measurement can be used for this step. Most known distance measurements like Euclidean distance are used.
- Distance sort in ascending order: computed distances are sorted in ascending order and minimum *k* distances are selected. Finding *k* class values: the corresponding classes of the *k*-nearest data are identified.
- Finding dominant class: In the last step, the identified the *k* classes are formed a ratio and the class which has maximum ratio is selected as the class of the test data.

F. Parzen Window Classifier

Parzen window algorithm [15] is a kind of nonparametric estimation algorithms. Non parametric estimation is an algorithm used for the population distribution estimation by the sample directly, without knowing the pattern of the population distribution or the pattern is not one of the typical patterns.

Assuming the area R_n is a *d*-dimensional hypercube and h_n is the length of one side of the hypercube, so the volume of the hypercube is: $V_n = h_n^d$. We define the window function:

$$\varphi(u) = \begin{cases} 1 & |u_j| \le 0.5 \quad , j = 1, 2, \ldots, d \\ 0 & else \end{cases} \qquad (18)$$

so *φ(u)* is a hypercube whose central point is the origin. When the sample x_i is inside the hypercube whose central point is x_i and volume is V_n , the value of *φ(u)* is 1 else is 0.
The count of the samples in the hypercube is obtained as:

$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{x^k - x^j}{h_n}\right) \qquad (19)$$

and the *n*-times estimation of probability *p*(*x*) is defined as:

$$\tilde{P}_n = \frac{k_n}{nV_n} \qquad (20)$$

The window described above is a square window; other types can also be selected, such as: normal window function, exponential window function, and so on.

III. Proposed Method

Flow chart of the proposed method is shown in Fig. 2. The steps of the new method can be summarized as below:

- Firstly, PCA is applied to dataset. PCA optimizes dataset when the number of features is too much or they have different scales.
- In the second step, LDA is performed on the obtained dataset to eliminate the redundancy information, increase between class distances, and decrease within class distances.
- Then, data is normalized, and the training and test datasets are separated; here the percentage of training and test dataset is 35% and 65%, respectively.
- At the last step, a classifier is applied to achieved dataset and the correct classification rate (CCR) is calculated.

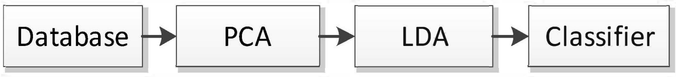

Fig. 2. Flow chart of the proposed method.

III. Exprimental Results

The experiments are applied on the five of well-known UCI machine learning data sets including Iris, Breast Cancer, Glass, Yeast, and Wine data sets [16], the main properties of the datasets are given in Table 1.

The proposed method is evaluated on the below datasets and encouraging improvements are achieved. The results are compared with the case that features are directly applied to the different classifiers, shown in Figs. 3, 4, and 5.

TABLE 1
FIVE DATASETS USED IN THIS STUDY

Properties	Iris	Breast cancer	Glass	Yeast	Wine
# of classes	3	2	6	10	3
# of total samples	150	569	214	1484	178
# of samples of each class	50 50 50	212 357	70 17 76 13 9- 29	463- 429 244- 163 51- 44 37- 30 20- 5	59 71 48

It is observed that proposed method significantly improves the classification performance. For instance, in the case of Iris dataset, there are 150 data in three different classes. By directly applying the features to the KNN classifier, 100 data are wrong classified which means 66.67% miss-classification. However, we have examined the proposed method 20 times. In the 3th, 4th, 5th, 6th and 13th trials, only three data are wrong classified, which show 2.22% miss-classification. Fig. 6 demonstrates the number of wrong classified data in two cases.

All results belonging to the three Bayesian, KNN, and Parzen window classifiers over 5 datasets are shown in Tables 2 and 3.

As mentioned, Figs. 3, 4, and 5 present the percentage values of the improvement. The best result is achieved on Yeast dataset, 81.24%, (using Bayesian classifier), in the 12th trial. Also Iris, Wine, and Breast Cancer datasets generally give good results. The best improvements are achieved in the 1st, 3th, and 13th trial as 67.34% (using KNN classifier), 57.3% (using KNN classifier), and 56.5% (using KNN classifier), respectively. For Glass dataset the best improvement is achieved at 20th trial as 16.41% (using Parzen window classifier).

Further, our study showed that using LDA and PCA alone, has less performance in comparison with the case of using both of them.

We have also compared the correct classification rate (CCR) for the proposed method using Bayesian classifier with the recently introduced methods, where the results are depicted in Table 4. The bold numbers present the best accuracy values for each dataset. It is observed that the new method achieves higher CCR than the other methods for 3 datasets and for the Iris dataset its performance is slightly less than the ALH [4], RBF network using PSO [6], and RBF network using GA [7]. However, note that the methods of [4], [6], and [7] are time consuming and need high computational complexity, while our method is fast and has less computational complexity.

IV. CONCLUSION

In this paper a fast and low computational complexity classification method were proposed. Which used PCA and LDA for processing dataset and the performance of different classifiers was evaluated including Bayesian, Parzen window, and *k*-nearest neighbor. The proposed method is tested on the five different datasets including Iris, Wine, Breast cancer, Yeast, and Glass. The improved classification performance is reported in comparison with the case that features are directly applied to the classifiers, e.g. On the Yeast dataset 81.24% improvement is achieved.

The method is proposed for noisy data set classifying applications and low cost hardware based solutions. The method is useful on the data sets that have few numbers of features. It optimizes the data which increases correct classification rate.

REFRENCES

[1] M. F. Amasyali, *Introduction to machine learning*, <http://www.ce.yildiz.edu.tr/mygetfile.php?id=686>, 2006.

[2] R. O. Duda and P. E. Hart,,*Pattern classification and scene Analysis*, John Wiley, 1973.

[3] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss", *Machine Learning*, vol. 29, pp. 103-130, 1997.

[4] T. Yang, V. Kecman, L. Cao, and C. Zhang, "Testing adaptive local hyperplane for multi-class classification by double cross-validation", *the International Joint Conference on Neural Network*, pp. 1-5, 2010

[5] Y. Zhang, and H. Liu, "Classification systems based on fuzzy cognitive maps", *Fourth International Conference on Genetic and Evolutionary Computing*, pp. 538-541, 2010.

[6] A. Esmaeili and N. Mozayani, "Adjusting the parameters of radial basis function networks using particle swarm optimization", *International Conference on Computational Intelligence for Measurement Systems and Applications*, pp.179-181, 2010.

[7] N. Naveen, V. Ravi, and C. Raghavendra Rao, "Rule extraction from different evalution trained radial basis function network using genetic algorithms", *IEEE International Conference on Automation Science and Engineering*, pp.152-157, 2009.

[8] N. S. Chaudhari, A. Tiwari, and J. Thomas, "Performance evaluation of SVM based semi-supervised classification algorithm", *10th International Conference on Control, Automation, Robotics, and Vision*, pp. 1942-1947, 2008.

[9] V. E. Neagoe, A. C. Mugioiu, and I. A. Stanculescu, "Face recognition using PCA using ICA versus LDA cascaded with the neural classifier of concurrent self-organizing maps", *International Conference on Communications (COMM)*, pp. 225-228, 2010.

[10] M. Yang, J. WU Wan, and G. L. JI."Random sampling LDA incorporating feature selection for face recognition", *Wavelet Analysis and Pattern Recognition (ICWAPR), International Conference* , pp.180-185 , 2010.

[11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Second edition, 1991.

[12] M. Aci, C. Inan, and M. Avcir, " A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm ". *Expert Systems with Applications*, Vol. 37, pp. 5061-5067, 2010

[13] T. M. Mitchell, *Machine learning*, McGraw-Hill, 1997.

[14] T. Yildiz, S. Yildirim, and D. T. Altilar, "Spam filtering with parallelized KNN algorithm". Akademik Bilisim, 2008.

[15] Y. Zhang, Z. Zheng, "Amplitude Distribution Estimation of White Noise Based on EMD and Parzen window", *International Conference on Electronic Measurement & Instruments*, pp. 2-292-2-296, 2009.

[16] http://archive.ics.uci.edu/ml/

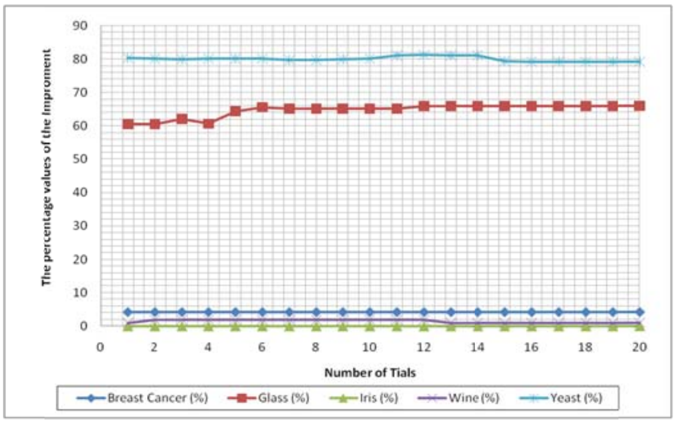

Fig. 3. The percentage values of the improvement on the number of wrong classified data on different datasets after applying the proposed method on Bayesian classifier.

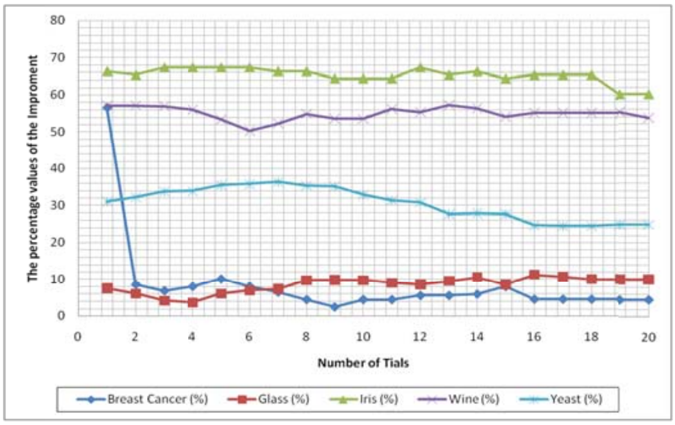

Fig. 4. The percentage values of the improvement on number of wrong classified data on different datasets after applying the proposed method on KNN classifier.

TABLE II.
Mean Of Achived Number (And Percentage Values %) Of Wrong Classified Data On Five Datasets With Directly Implementing The Features To Classifiers Over 20 Triels

Dataset/ classifier	Iris	Breast Cancer	Glass	Yeast	Wine
Bayesian	4 (2.5%)	23 (4.13%)	138 (64.77%)	1410 (94.85%)	3 (1.44%)
KNN	100 (66.67%)	222 (38.94%)	153 (71.31%)	1447 (97.51%)	110 (62.04%)
Parzen window	26 (19.24%)	286 (50.18%)	161 (75.25%)	1422 (95.84%)	81 (45.74%)

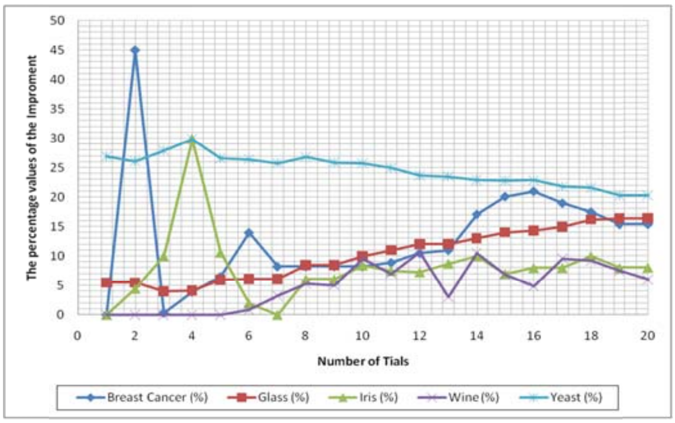

Fig. 5. The percentage values of the improvement on number of wrong classified data on different datasets after applying the proposed method on Parzen window classifier**.**

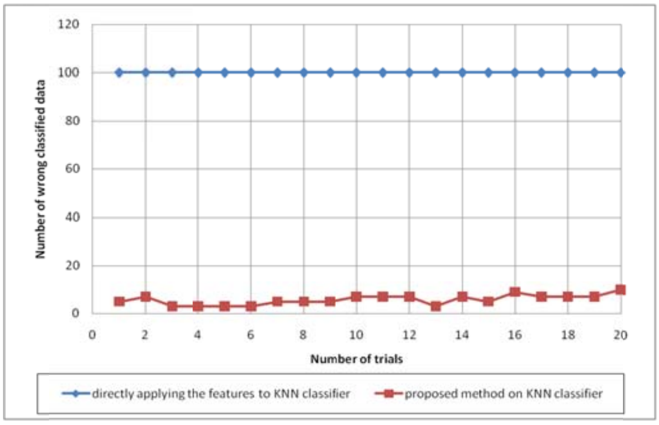

Fig. 6. Achieved number of wrong classified data after the trials on Iris dataset with directly applying the features to KNN classifier and the proposed method**.**

TABLE III.
Mean Of Achived Number (And Percentage Values %) Of Wrong Classified Data On Five Datasets With Proposed Method Over Classifiers Over 20 Triels

Dataset/ classifier	Iris	Breast Cancer	Glass	Yeast	Wine
Bayesian	3 (2.22%)	**0** **(0%)**	**0** **(0%)**	225 (15.16%)	**0** **(0%)**
KNN	6 (4.44%)	189 (33.23%)	134 (62.50%)	986 (66.45%)	13 (7.57%)
Parzen window	17 (12.58%)	225 (39.52%)	139 (65.20%)	1058 (71.29%)	73 (40.83%)

TABLE IV.
Comparision Of The Experimental Results With Different Methodes (CCR%)

Method/ Dataset	proposed method	ALH [4]	Fuzzy Cognitive Maps [5]	RBF Network using PSO [6]	RBF Network using GA[7]	SVM based Classification [8]
Iris	97.78%	97.90%	97.33%	**98.21%**	98.66%	95%
Wine	**100%**	98.50%	97.75%	97.74%	98.23%	_
Breast Cancer	**100%**	77.80%	94.58%	_	97.35%	71%
Glass	**100%**	70.10%	_	77.48%	_	72%
Yeast	84.84%	_	_	_	_	_