Supplementary material: Prediction of protein-protein interactions using one-class classification methods and integrating diverse biological data

José A. Reyes[1,2] and David Gilbert[1]

[1]Bioinformatics Research Centre, Department of Computing Science, University of Glasgow, G12 8QQ, UK.

[2]Facultad de Ingeniería, Universidad de Talca, Chile

S1 - Description of one-class classification (OCC) methods

OCC methods can be classified according to the way in which they analyze, describe and generate a model for the separation of targets and outlier examples [S1]. Two types of OCC were employed in this reseach: *Density estimation* methods based in the estimation of the probability density distribution of the training data using some probabilistic model (i.e. Gaussian distribution), then a threshold is selected and then used to compare with the density of new objects in order to classify them; And *Boundary* methods, based in the generation of a frontier or boundary around the target objects, which is optimized to accept most of the target examples and at the same time reject most of the outliers. Boundary approaches are mainly focused on those examples objects which are located near the boundary. Here we present a detailed description of the four OCC approaches evaluated in this research:

Gaussian density estimation:

This is the simplest of the OCC density approaches. The examples of the target class used for training are modeled as a Gaussian distribution. In the dd_tools implementation the complete density estimation is not obtained and just the Mahalanobis distance is employed and calculated for each example X as:

$$f(X) = (X - \mu)^T \sum^{-1} (X - \mu) \tag{1}$$

where the mean μ and the covariance matrix $\sum$ are estimated from the whole sample of objects used. The $f(X)$ value for new objects is then compared against a threshold θ and classified as a target if $f(X) \leq \theta$ or else as an outlier.

Mixture of Gaussian density estimation:

In this case a linear combination of several (i.e. N) different Gaussian distributions is employed to model the target class examples used for training, obtaining a more flexible model compared with the single Gaussian distribution approach. The training data is divided into N different clusters, each of which is modeled by a single Gaussian distribution. The distance function $f(X)$ changes in this case to the form:

$$f(X) = \sum_{i=1}^{N} \alpha_i \exp(-(X - \mu_i)^T \sum_{i}^{-1} (X - \mu_i)) \tag{2}$$

where α_i are the mixing coefficients. The parameters of each cluster μ_i, $\sum_i$, and α_i are optimized using the EM algorithm. A threshold θ is fixed again and used to classify new objects as in the previous case. For this approach it is possible to include outlier objects in the training phase, setting independent mixture of Gaussian distributions for both target and outlier examples, considering N_{target} and $N_{outlier}$ different clusters. The number of clusters considered for target and outlier data should be fixed and can be varied in order to obtain the optimal performance of the generated model.

Parzen density estimation:

In Parzen density estimation an independent Gaussian distribution is considered for each one of the T target objects used for training a model for this class. Consequently in this case the distances to all training objects have to be considered. In the dd_tools implementation of this approach the function $f(X)$ is as follows:

$$f(X) = \sum_{i=1}^{T} \exp(-(X - X_i)^T h^{-2} (X - X_i)) \tag{3}$$

The smoothing parameter h, commonly called the *Parzen width*, is introduced here and is related to the width of a region R (in a Gaussian space) generated around each object in order to separate the target from outlier zones. The rest of the classification process follows in a similar manner to those in the previous approaches. The value of h can be varied in order to find an optimal performance related to the specific task conditions.

Support vector data description(SVDD):

This technique is a boundary approach based on the support vector machines (SVM) theory, which aim to create a closed hyper-spherically shaped boundary around the examples used to train the model of the target class. Following the description in [S1, S2] the hyper-sphere is characterized by the centre a and radius R and is supported for several objects as in the case of SVMs. The objective then is to minimize the volume of the sphere which is possible by minimizing the value of R^2. This minimization problem is similar to that in the SVM approach and consequently it is possible to generate the same kind of approximation solution, with the advantage of employing a more flexible feature representation using kernel functions (i.e. linear, polynomial and Gaussian kernels). This approach permits the use of outlier examples in the training stage in order to generate a more tight description of the hyper-spherical boundary. The kernel type and its respective parameters can be varied in this implementation in order to obtain the optimal performance conditions.

S2 - Difference between the AUC and AUC-50 analysis

In this research we observed that while the AUC-50 analysis presents substantial differences between the diverse learning methods evaluated, the comparison of AUC scores shows that there is no appreciable significant difference between them on these specific conditions. The difference between the AUC and AUC-50 analysis can be clearly appreciated from the ROC curves of the different learning methods evaluated. Figure 1(a) shows an example of the ROC

curves for the different learning techniques used for evaluation of one validation subset. There are no apparent important differences between these curves, although it is possible to observe that in this case the DT curve performs slightly better than the other methods. However when zooming in on these ROC curves and considering only the portion of them related with the AUC-50 region, presented in Figure 1(b), it is possible to appreciate that there is a noticeable difference in the performance of the diverse methods and also that parzen OCC method clearly outperform the rest of the conventional learning approaches evaluated.

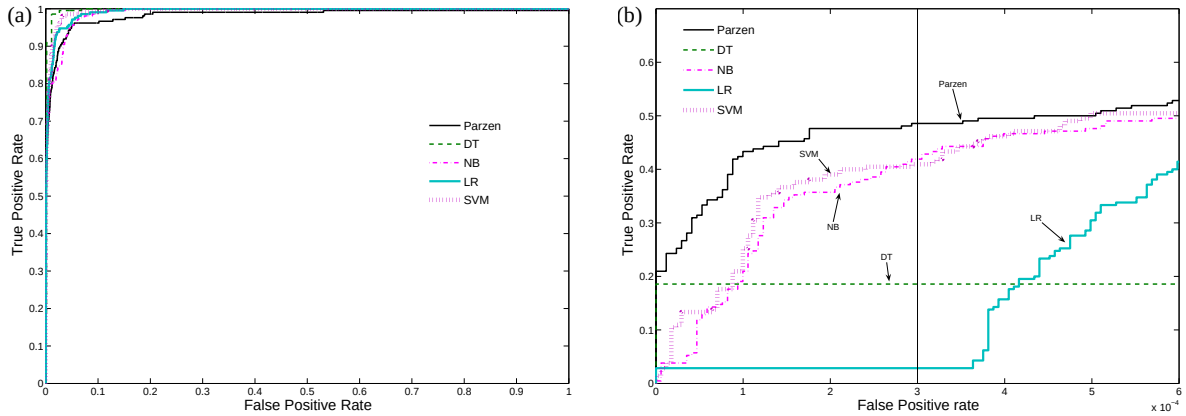

Figure 1: Example of ROC curve analysis: (a) Whole ROC curves for the different learning methods evaluated. (b) Partial ROC curves for the different learning methods evaluated, showing the region related to AUC-50. The vertical line on (b) indicates the point where approximately the first 50 false-positive examples are reached.

S3 - Evaluation of feature importance

Finally we evaluated the individual effect of the different biological features used in this research on the performance of the parzen OCC approach. For this we made the following procedure for each one of the features: We first removed one of the attributes from the original data set, then we train and test a parzen OCC model on this reduced scenario estimating AUC and AUC-50 scores, finally we compared these performance measures with the obtained when all available biological information is used.

Table 1 shows the results of the application of this procedure to the different features. We observed that the major effect on the performance of parzen OCC method is produced when functional similarity and m-RNA expression data is removed. This is consistent with previous reported studies which considered these type of biological information as the most important for this task [S3, S4, S5]. On these cases AUC-50 scores decreased drastically up to one half of the baseline value. In relation to overall AUC scores, a significantly decrement is only noticed when functional similarity data is removed.

The elimination of essentiality and high-throughput information seems to have a reduced impact on AUC-50 scores, but it is interesting to observe that the overall AUC performance increase slightly when high-throughput information is removed which can be explained due the high false-positive and false-negative rates attributed to this kind of features.

Table 1: Evaluation of the individual effect of the different biological attributes in the performance of the OCC parzen approach

Feature description	AUC	AUC-50
ALL features	0.9801 ± 0.0075	0.4010 ± 0.0282
GO removed	0.9186 ± 0.0121	0.2094 ± 0.0189
MIPS removed	0.9412 ± 0.0135	0.1983 ± 0.0225
Expression removed	0.9775 ± 0.0050	0.1883 ± 0.0238
Essentiality removed	0.9800 ± 0.0081	0.3380 ± 0.0273
High-throughput removed	0.9887 ± 0.0037	0.3463 ± 0.0261

References

[S1] D. M. J. Tax. One-class classification. PhD thesis, Delft University of Technology, 2001.

[S2] D. M. J. Tax and R. P. W. Duin. Support vector data description. Machine Learning, 54(1):45–66, 2004.

[S3] N. Lin, B. Wu, R. Jansen, M. Gerstein, and H. Zhao. Information assessment on predicting protein-protein interactions. BMC Bioinformatics, 5(1):154, 2004.

[S4] L. J. Lu, Y. Xia, A. Paccanaro, H. Yu, and M. Gerstein. Assessing the limits of genomic data integration for predicting protein networks. Genome Res., 15(7):945–953, 2005.

[S5] Y. Qi, Z. Bar-Joseph, and J. Klein-Seetharaman. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. Proteins: Structure, Function, and Bioinformatics, 63(3):490–500, 2006.