

Multi-task Drug Bioactivity Classification with Graph Labeling Ensembles

Hongyu Su and Juho Rousu

Department of Computer Science,
P.O. Box 68, 00014 University of Helsinki, Finland
{hongyu.su, juho.rousu}@cs.helsinki.fi

Abstract. We present a new method for drug bioactivity classification based on learning an ensemble of multi-task classifiers. As the base classifiers of the ensemble we use Max-Margin Conditional Random Field (MMCRF) models, which have previously obtained the state-of-the-art accuracy in this problem. MMCRF relies on a graph structure coupling the set of tasks together, and thus turns the multi-task learning problem into a graph labeling problem. In our ensemble method the graphs of the base classifiers are random, constructed by random pairing or random spanning tree extraction over the set of tasks.

We compare the ensemble approaches on datasets containing the cancer inhibition potential of drug-like molecules against 60 cancer cell lines. In our experiments we find that ensembles based on random graphs surpass the accuracy of single SVM as well as a single MMCRF model relying on a graph built from auxiliary data.

Keywords: drug bioactivity prediction; multi-task learning; ensemble methods; kernel methods.

1 Introduction

Molecular classification, the task of predicting the presence or absence of the bioactivity of interest, has been benefited from variety of methods in statistics and machine learning [7]. In particular, kernel methods [9,16,2,7] have emerged as an effective way to handle the non-linear properties of chemicals. However, classification methods focusing on a single target variable are probably not optimally suited to drug screening applications where large number of target cell lines are to be handled.

In [15] a multi-task (or multilabel) learning approach was proposed to classify molecules according to their activity against a set of cancer cell lines. It was shown that the multilabel learning setup improves predictive performance over a set of support vector machine based single target classifiers. The multilabel classifier applied, Max-Margin Conditional Random Field (MMCRF) [11] relies on a graph structure coupling the outputs together. In [15] the graph was extracted

from auxiliary data, concerning sets of experiments conducted on the cancer cell lines, by simple techniques such as correlation thresholding and maximum weight spanning tree extraction.

In this paper, we develop ensemble learning methods for the multi-task learning setup. In our method, MMCRF models are used as the ensemble components. Unlike other ensemble learners for multi-task setups, our method does not require bootstrapping of the training data or changing instance weights to induce diversity among the ensemble components. In our case, the diversity is provided by the randomization of the output graphs, which combined with discriminative training of the base MMCRF classifiers, realizes the benefits typically seen from ensemble approaches. The random graph approach is compared against single classifiers and ensembles on graphs built from auxiliary data with different graph extraction methods, including inverse covariance learning [5] that is theoretically superior to correlation thresholding for extracting statistical dependencies.

Ensembles of multi-task or multilabel classifiers have been proposed in a few papers prior to ours, but with important differences both in the methods and the applications. In general, the previous approaches can be divided into two groups based on the source of diversity among the base classifiers of the ensemble. The first group of methods, boosting type, relies on changing the weights of the training instances so that difficult-to-classify instances gradually receive more and more weight. The Boostexter method [12] by the inventors of boosting has a multilabel learning variant. Later, Esuli et al. [4] developed a hierarchical multilabel variant of AdaBoost. Neither method explicitly considers label dependencies but the multilabel is considered essentially a flat vector. The second group of methods, bagging, is based on bootstrap sampling the training set several times and building the base classifiers from the bootstrap samples. Averaging over the ensemble gives the final predictions. Schietgat et al [13] concentrate a bagging in multilabel gene function prediction. They build ensembles of predictive clustering trees (PCT) by bagging, that is, bootstrap sampling of the data several times to arrive at a set of different models. In their approach, there is also no structure defined for the tasks, but the multilabel is essentially treated as a flat vector. Finally, Yan et al. [18] select different random subsets of input features and examples to induce the base classifiers.

The remainder of the article is structured as follows. In section 2 we present the base classifier MMCRF and the multi-task ensemble learning approach. In section 3 we validate the methods empirically, in particular we show that the ensemble approach exceeds the accuracy of MMCRF, which to our knowledge currently has the state-of-the-art predictive performance. In section 4 we aim to provide intuition of the hows and whys of the behaviour of the new method by relating the new ensemble approach to other multi-task and multilabel ensemble approaches. In section 5 finish with concluding remarks.

2 Ensemble Learning with Max-Margin Conditional Random Field Models on Random Graphs

Ensemble learners [3,8], such as boosting [12] and bagging [1] are based on the notion that a set of *weak learners*, those that have accuracy higher than coin tossing, may produce a strong learner with high accuracy when appropriately combined. It has been found that the diversity among the base models is the key property. The diversity may arise from re-weighting of examples, bootstrap resampling of examples, from the different inductive biases of the base learners, or in multiclass setup, or by generating a set of derived binary classification tasks (one-vs-the rest, one vs. one, and error-correcting output codes [3]).

In this section we present our ensemble learning approach where the diversity among the base learners comes from a different source, namely randomized graph structures that are used to couple the tasks. We use a majority voting approach over the predictions of the base classifiers, namely labelings of the randomized graphs. Two basic types of graphs are used, random spanning trees and random pairings of targets (Figure 1). As the base learner, we use the MMCRF algorithm [11].

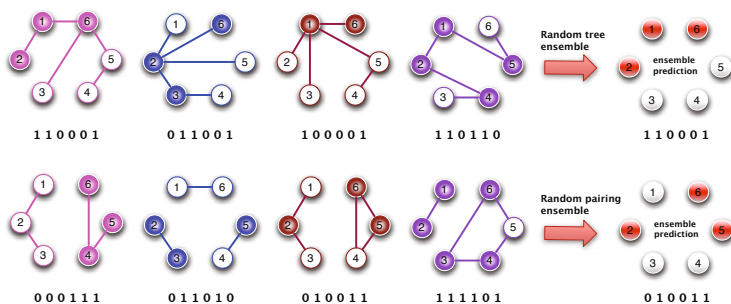


Fig. 1. Ensemble prediction from a set of random spanning trees (top) and a set of random pairings of tasks (bottom). The varying graph structures provide the required diversity among the ensemble components. Majority vote decides the final predicted label for each task.

The method for generating the ensemble is depicted in Algorithm 1. The algorithm receives a training sample of molecules x_i , computes the input kernel K and embarks on the ensemble learning phase. For each base model, a random graph G_t of the type specified by the user is drawn to couple the outputs \mathbf{y}_i which are the inhibition potentials of molecule x_i against 60 cancer cell lines. The input kernel, label data and the graph are given as input to the MMCRF (see Section 2.1) that learns the graph labeling. After the ensemble has been generated, the ensemble prediction is extracted in post-processing: we extract the majority vote over the graph labelings from the sign of the mean of the base classifier predictions.

Algorithm 1. Ensemble learning algorithm with random graph multi-task classifiers

Input: Training sample $S = \{(x_i, \mathbf{y}_i)\}_{i=1}^m$, ensemble size T , type of the graph generated $graphType$, n the number of nodes in the graph, type of input kernel applied $kernelType$

Output: Multi-task classification ensemble $(f^{(1)}, \dots, f^{(T)})$

```

1:  $K = buildKernel(\{x_i\}_{i=1}^m, kernelType)$ 
2:  $t = 0$ 
3: while  $t < T$  do
4:    $t = t + 1$ 
5:    $G_t = randomGraph(n, graphType)$ 
6:    $f^{(t)} = learnMMCRF(K, (\mathbf{y}_i)_{i=1}^m, G_t)$ 
7: end while
8: return  $f = (f^{(1)}, \dots, f^{(T)})$ 

```

2.1 Learning Graph Labeling with MMCRF

The MMCRF method used as the base learner in the multi-task ensembles is an instantiation of the structured output prediction framework MMCRF [11] for associative Markov networks and can also be seen as a sibling method to HM³[10], which is designed for hierarchies. We give a brief outline here, the interested reader may check the details from the above references.

The MMCRF learning algorithm takes as input a matrix $K = (k(x_i, x_j))_{i,j=1}^m$ of kernel values $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ between the training patterns, where $\phi(x)$ denotes a feature description of an input pattern (in our case a potential drug molecule), and a label matrix $Y = (\mathbf{y}_i)_{i=1}^m$ containing the multilabels $\mathbf{y}_i = (y_1, \dots, y_k)$ of the training patterns. The components $y_j \in \{-1, +1\}$ of the multilabel are called microlabels, which in multi-task learning setup, correspond to labels of different tasks. In addition, the algorithm assumes an associative network $G = (V, E)$ to be given, where node $j \in V$ corresponds to the j 'th component of the multilabel and the edges $e = (j, j') \in E$ correspond to a microlabel dependency structure.

The model learned by MMCRF takes the form of a conditional random field with exponential edge-potentials,

$$P(\mathbf{y}|x) \propto \prod_{e \in E} \exp(\mathbf{w}_e^T \varphi_e(x, \mathbf{y}_e)) = \exp(\mathbf{w}^T \varphi(x, \mathbf{y})),$$

where $\mathbf{y}_e = (y_j, y_{j'})$ denotes the pair of microlabels of the edge $e = (j, j')$. A joint feature map $\varphi(x, \mathbf{y}) = \phi(x) \otimes \psi(\mathbf{y})$ is composed via tensor product of input $\phi(x)$ and output feature map $\psi(\mathbf{y})$, thus including all pairs of input and output features. The output feature map is composed of indicator functions $\psi_e^u(\mathbf{y}) = \mathbb{I}_{\mathbf{y}_e = u}$ where u ranges over the four possible labelings of an edge given binary node labels. The corresponding weights are denoted by $\mathbf{w} = (\mathbf{w}_e)_e$. The benefit of the tensor product representation is that context (edge-labeling)

sensitive weights can be learned for input features and no prior alignment of input and output features needs to be assumed.

The parameters are learned by maximizing the minimum loss-scaled margin between the correct training examples (x_i, \mathbf{y}_i) and incorrect pseudo-examples $(x_i, \mathbf{y}), \mathbf{y} \neq \mathbf{y}_i$, while controlling the norm of the weight vector. The dual soft-margin optimization problem takes the form

$$\begin{aligned} \min_{\alpha \geq 0} \quad & \sum_{i, \mathbf{y}} \alpha(i, \mathbf{y}) \ell(\mathbf{y}_i, \mathbf{y}) - \frac{1}{2} \sum_{i, \mathbf{y}} \sum_{j, \mathbf{y}'} \alpha(i, \mathbf{y}) K(x_i, \mathbf{y}; x_j, \mathbf{y}') \alpha(i, \mathbf{y}') \\ \text{s.t.} \quad & \sum_{\mathbf{y}} \alpha(i, \mathbf{y}) \leq C, \forall i, \end{aligned} \quad (1)$$

where $K(x_i, \mathbf{y}; x_j, \mathbf{y}') = \Delta\varphi(i, \mathbf{y})^T \Delta\varphi(j, \mathbf{y}') = K_X(x_i, x_j) \odot K_{\Delta Y}(\mathbf{y}, \mathbf{y}')$ is the joint kernel composed of the input $K_X(x_i, x_j)$ and output $K_{\Delta Y}(\mathbf{y}_i, \mathbf{y}')$ kernels. The underlying joint feature map is expressed by

$$\Delta\varphi(i, \mathbf{y}) = (\varphi(x_i, \mathbf{y}_i) - \varphi(x, \mathbf{y})) = \phi(x_i) \otimes (\psi(\mathbf{y}_i) - \psi(\mathbf{y})),$$

that is, joint feature difference vectors between the true (\mathbf{y}_i) and a competing output (\mathbf{y}) .

As the input kernel we use the hash fingerprint Tanimoto kernel [9] that was previously shown [15] to be a well performing kernel in this task. Hash fingerprints enumerate all linear fragments of length n in a molecule. A hash function assigns each fragment a hash value that determines its position in descriptor space. Given two fingerprint vectors x and z , Tanimoto kernel is the way to measure their similarity defined as

$$K_X(x, z) = \frac{|I(x) \cap I(z)|}{|I(x) \cup I(z)|},$$

where $I(x)$ denotes the set of indices of 1-bits in x .

As the loss function we use *Hamming loss*

$$\ell_{\Delta}(\mathbf{y}, \mathbf{u}) = \sum_j \llbracket y_j \neq u_j \rrbracket$$

that is gradually increasing in the number of incorrect microlabels so that we can make a difference between 'nearly correct' and 'clearly incorrect' multilabel predictions.

2.2 Graph Generation for Cancer Cell Lines

In the anti-cancer bioactivity prediction problem, a single task entails classification of drug molecules according to whether they are active or inactive against one of the 60 cancer cell lines. The nodes of the graph G to be labeled thus correspond to cancer cell lines. The edges of the graph depict coupling of the tasks, denoting a potential statistical dependency that is to be utilized in predicting the graph labels (Figure 2).

To generate random graphs G_t we use two approaches.

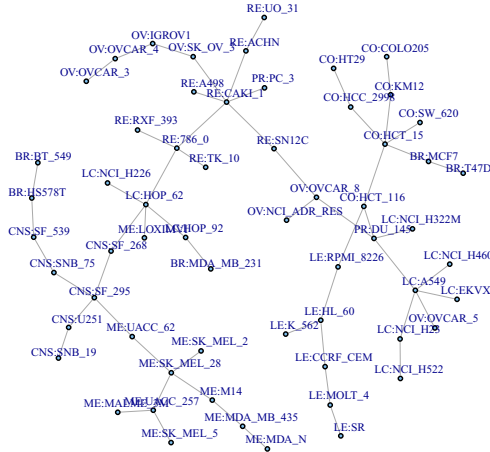


Fig. 2. Example of a cell line graph

- In the random spanning tree approach, we first generate a random correlation matrix and extract the spanning tree out of the matrix with the above described approach.
- In the random pairing approach, one takes each vertex in turn, randomly draws another vertex and couples the two with an edge.

We note that the random graph approach lets us build ensembles whose size is not limited in practice.

We compare the random graphs against the approach used by [15], namely, graphs built from Radiation RNA Array data, available for the cancer cell lines from NCI database¹. To extract a graph out of the correlation matrix we use the graphical lasso [5] which estimates a sparse graph model by using L_1 (lasso) regularization on inverse covariance matrix, and is theoretically a better method than the simple thresholding of the covariance matrix, applied in [15]. Graphical lasso assumes multivariate Gaussian distribution over cell lines with mean μ and covariance matrix Σ . The inverse covariance matrix Σ^{-1} is a good indicator for conditional independencies [6], where variable i and j are conditional independent given other variables if the ij th entry of Σ^{-1} is zero. It imposes L_1 penalty during the estimation of Σ^{-1} to increase the sparsity of the resulted graph. The objective is to maximize the penalized log-likelihood

$$\log \det \Sigma^{-1} - \text{tr}(S \Sigma^{-1}) - \rho \|\Sigma^{-1}\|_1,$$

where tr is the trace of the matrix, S is empirical covariance matrix, and $\|\Sigma^{-1}\|_1$ is the L_1 norm of Σ^{-1} . Particularly in our application, we post processed the estimated sparse graph to be a tree-liked one.

¹ <http://discover.nci.nih.gov/cellminer/home.do>

3 Experiments

3.1 Data and Preprocessing

In this paper we use the NCI-Cancer dataset obtained through PubChem Bioassay² [17] data repository. The dataset, initiated by National Cancer Institute and National Institutes of Health (NCI/NIH), contains bioactivity information of large number of molecules against several human cancer cell lines in nine different tissue types including leukemia, melanoma and cancers of the lung, colon, brain, ovary, breast, prostate, and kidney. For each molecule tested against a certain cell line, the dataset provide a bioactivity outcome that we use as the classes (active, inactive).

Currently, there are 43197 molecules in the PubChem Bioassay database together with their activities information in 73 cancer cell lines. 60 cell lines have screen experimental results for most molecules and 4547 molecules have no missing data in these cell lines. Therefore these cell lines and molecules are selected and employed in our experiments. However, molecular activity data are highly biased over the 60 cell lines: Around 60% of molecules are inactive in all cell lines, while still a relatively large proportion of molecules are active against all cell lines. These molecules are less likely to be potential drug candidates than the ones in the middle part of the histogram.

To tackle the skewness problem, Su et al. [15] prepared three different versions of the datasets, which approach is also followed here:

Full Data. This dataset contains all 4547 molecules in the NCI-Cancer dataset that have their activity class (active vs. inactive) recorded against all 60 cancer cell lines.

No-Zero-Active. From full data, we removed all molecules that are not active towards any of the cell lines. The remaining 2303 molecules are all active against at least one cell line.

Middle-Active. Here, we followed the preprocessing suggested in [14], and selected molecules that are active in more than 10 cell lines and inactive in more than 10 cell lines. As a result, 545 molecules remained and were employed in our experiments.

3.2 Compared Methods

Three kinds of multi-task classifier ensembles are compared:

- SVM: Support vector machines (SVM) are used as the single-task non-ensemble baseline classifier.
- MMCRF-Glasso: An MMCRF model where the underlying graph connecting the tasks is built by graphical lasso from auxiliary data.
- MMCRF-EnsRT: An ensemble of 1-500 MMCRF models, where the graph connecting the tasks is built by a random spanning tree.

² <http://pubchem.ncbi.nlm.nih.gov>

- MMCRF-EnsRP: An ensemble of 1-500 MMCRF models, where the graph connecting the tasks is built by random pairing of the tasks.

In the tests by [15], a relatively hard margin ($C = 100$) emerged as the most favorable for SVM, while MMCRF proved to be quite insensitive as regarding margin softness. Here we used the same value for all compared classifiers.

3.3 Experiment Setup and Performance Measures

Because of the skewness of the multilabel distribution we used the following *stratified 5-fold cross-validation* scheme in all experiments reported such that we group the molecules in equivalence classes based on the number of cell lines they are active against. Then each group is randomly split among the five folds. The procedure ensures that also the smaller groups have representation in all folds. Besides overall classification accuracy, we also report microlabel F_1 score, the harmonic mean of precision and recall

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

In particular, we pool together individual microlabel predictions over all examples and all cell lines, and count accuracy and F_1 from the pool.

We generated hash fragments features from OpenBabel³ which is a chemical toolbox available in public domain. We used default value for enumerating all linear structures up to length seven. Then Tanimoto kernel was built based on hash fingerprints features and normalized.

3.4 Results

Figure 3 illustrates the performance of the compared methods on the three versions of the datasets. All models based on MMCRF are clearly more accurate than SVM. Among single models and small ensembles, MMCRF-Glasso is the most competitive method, showing that the auxiliary data contains information that can be successfully used to improve predictive performance.

Both the random pairing and random tree based ensembles steadily improve accuracy and F_1 score as the number of base models increases. SVM falls behind the random graph ensembles even on small ensemble sizes ($T < 5$). With larger ensemble sizes, both types of ensembles end up superior to MMCRF-Glasso in terms of classification accuracy. In terms of F_1 score, the best method depends on the dataset: on the Middle-Active dataset, the random tree ensemble outperform random pairing one, and MMCRF-Glasso is slightly behind. On No-zero-Active and Full data, random pairing ensemble ends up the best method. This result might reflect the sizes of the datasets: the Middle-Active dataset is significantly smaller than the other two, and perhaps the random pairing ensemble requires more data for best results.

³ <http://openbabel.org>

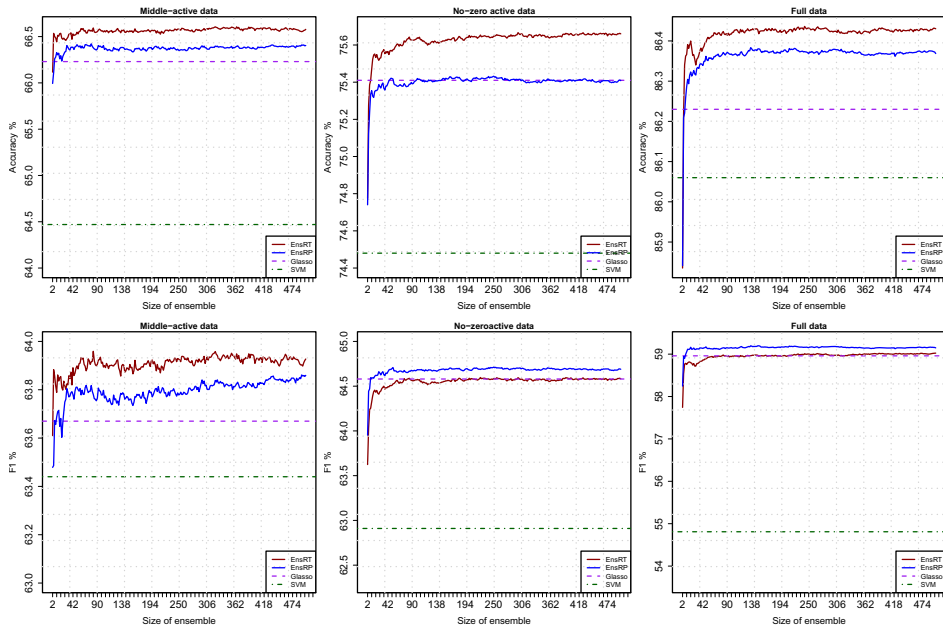


Fig. 3. Accuracy against number of individual classifiers in ensemble methods from different version of datasets. The red line corresponds to random tree ensemble, and blue line is random pairing ensemble. The performance of single models (SVM and MMCRF-Glasso) are depicted by the horizontal lines.

Table 1 shows the prediction performance from SVM, Glasso, EnsRP and EnsRT from three versions of the dataset. We performed two-tailed sign test to identify whether the differences in accuracy and F_1 score in individual cell lines are statistically significant. P -values for the difference over the worst classifier and the ones towards the best classifier are shown as asterisks and crosses. The result shows that, in terms of accuracy and F_1 the multi-task methods outperform SVM in all versions of datasets in a statistically significant manner. EnsRT outperforms Glasso in terms of accuracy in statistically very significant manner.

4 Discussion

The results of this paper show that ensemble methods can be effectively combined with a graph-based multi-task learner such as MMCRF. From machine learning point of view, perhaps the most surprising result obtained here is that in an ensemble, the base graph labeling models can be successfully learnt on random graphs, as opposed to using some auxiliary data or prior knowledge to extract graphs that aim to reflect statistical dependencies.

The present ensemble method differs from previous approaches in that the diversity among the base classifiers arises from the different random output

Table 1. Overall accuracy and microlabel F_1 scores. P -values for the differences over the worst classifier in each version of the dataset are marked with asterisks. P -values for the differences towards the best classifier are marked with crosses. Single, double and triple symbols correspond to p -value below 0.05, 0.01 and 0.001.

Dataset	Accuracy				F_1			
	SVM	Glasso	EnsRP	EnsRT	SVM	Glasso	EnsRP	EnsRT
Middle-Active	64.5% _{†††}	66.2% ^{****} _{†††}	66.5% ^{****} _†	66.6% ^{****}	63.4% _†	63.7%	63.9% [*]	63.9% [*]
No-Zero-Active	74.5% _{†††}	75.4% ^{****} _{†††}	75.4% ^{****} _{†††}	75.7% ^{****}	62.9% _{†††}	64.6% ^{****}	64.7% ^{****}	64.6% ^{****}
Full	86.1% _{†††}	86.2% ^{****} _{†††}	86.3% ^{****}	86.4% ^{****}	54.8% _{†††}	59.0% ^{****}	59.2% ^{****}	59.0% _{†††}

structures, we do not reweight training examples as in boosting and we do not resample the data like in bagging methods. At the same time, each weak learner is trained to discriminate between different multilabels as well as possible.

Another way to understand the phenomenon is to see the edges of the task network as 'experts', and the collection of edges adjacent to a node as a 'expert committee' voting on the node label, each from a different context. The random pairing of tasks then induces a random set of experts. Random tree of tasks, in addition, makes the experts to negotiate on all node labels in order to keep the tree labeled consistently. Our experiments suggest that enforcing this consistency also may be beneficial.

5 Conclusions

We presented an ensemble approach for multi-task classification of drug bioactivity. The base classifiers of the ensemble, learned by Max-Margin Conditional Random Field algorithm (MMCRF), predict a labeling of a graph coupling the tasks together. The predictive performance of two types of ensembles, one based on random pairing of tasks, another based on a random spanning tree of tasks, surpasses that of SVM as well as single MMCRF model where the underlying graph has been built from auxiliary data using graphical lasso.

Acknowledgements. The work was financially supported by Helsinki Doctoral Programme in Computer Science (Hecse), Academy of Finland grant 118653 (ALGODAN), and in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-2007-216886. This publication only reflects the authors' views.

References

1. Breiman, L.: Bagging predictors. *Machine Learning* 24, 123–140 (1996)
2. Ceroni, A., Costa, F., Frasconi, P.: Classification of small molecules by two- and three-dimensional decomposition kernels. *Bioinformatics* 23, 2038–2045 (2007)
3. Dietterich, T.: Ensemble methods in machine learning. *Multiple classifier systems*, 1–15 (2000)

4. Esuli, A., Fagni, T., Sebastiani, F.: Boosting multi-label hierarchical text categorization. *Information Retrieval* 11(4), 287–313 (2008)
5. Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3), 432–441 (2008)
6. Meinshausen, N., Bühlmann, P., Zürich, E.: High dimensional graphs and variable selection with the lasso. *Annals of Statistics* 34, 1436–1462 (2006)
7. Obrezanova, O., Segall, M.D.: Gaussian processes for classification: Qsar modeling of admet and target activity. *Journal of Chemical Information and Modeling* 50(6), 1053–1061 (2010)
8. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research* 11, 169–198 (1999)
9. Ralaivola, L., Swamidass, S., Saigo, H., Baldi, P.: Graph kernels for chemical informatics. *Neural Networks* 18, 1093–1110 (2005)
10. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-Based Learning of Hierarchical Multilabel Classification Models. *The Journal of Machine Learning Research* 7, 1601–1626 (2006)
11. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Efficient algorithms for max-margin structured classification. *Predicting Structured Data*, 105–129 (2007)
12. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. *Machine Learning* 39(2/3), 135–168 (2000)
13. Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., Džeroski, S.: Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC bioinformatics* 11(1), 2 (2010)
14. Shivakumar, P., Krauthammer, M.: Structural similarity assessment for drug sensitivity prediction in cancer. *Bioinformatics* 10, S17 (2009)
15. Su, H., Heinonen, M., Rousu, J.: Structured Output Prediction of Anti-cancer Drug Activity. In: Dijkstra, T.M.H., Tsivtsivadze, E., Marchiori, E., Heskes, T. (eds.) *PRIB 2010. LNCS*, vol. 6282, pp. 38–49. Springer, Heidelberg (2010)
16. Trotter, M., Buxton, M., Holden, S.: Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comp. and Chem.* 26, 1–20 (2001)
17. Wang, Y., Bolton, E., Dracheva, S., Karapetyan, K., Shoemaker, B., Suzek, T., Wang, J., Xiao, J., Zhang, J., Bryant, S.: An overview of the pubchem bioassay resource. *Nucleic Acids Research* 38, D255–D266 (2009)
18. Yan, R., Tesic, J., Smith, J.: Model-shared subspace boosting for multi-label classification. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 834–843. ACM (2007)