Hierarchical ensemble methods for gene/protein function prediction

Giorgio Valentini
valentini@di.unimi.it

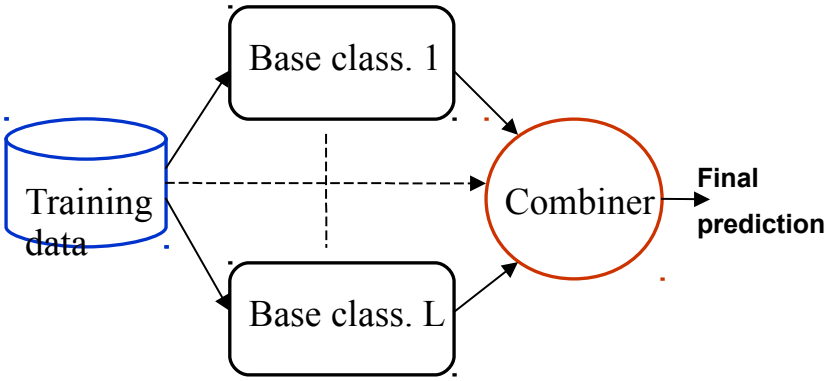
DI - Dipartimento di Informatica

Università degli Studi di Milano

A brief introduction to ensemble methods

Ensembles are sets of learning machines that work together to solve a machine learning problem

E.g.:



Ensemble methods are one of the main topics in machine learning research

Why should we use ensembles?

From *empirical studies* : ensembles are often much more accurate than individual learning machines (Freund & Schapire (1995), Bauer & Kohavi (1999), Dietterich (2000), …)

Different *theoretical explanations* proposed to justify their effectiveness (Kittler (1998), Schapire et al. (1998), Kleinberg(2000), Allwein et al. (2000), …).

Very fast development of *computer technology*: availability of very fast computers and networks of workstations at a relatively low cost.

An example: majority voting ensembles

A dichotomic classification problem and L classifiers with error < 0.5

The resulting majority voting ensemble has an error lower than the single classifier

For instance, 21 classifiers, p<0.3, probability of error of each classifier

$$P_{error} = \sum_{i=\lceil L/2 \rceil}^{L} \binom{L}{i} p^i (1-p)^{L-i} \Rightarrow P_{error} = 0.026 << p$$

Condorcet Jury Theorem (XVIII century) : the judgment of a committee is superior to those of individuals, (if their competence is reasonable, e.g. p<0.5)

A lot of methods …

- Majority and weighted voting (Perrone and Cooper, 1993, Lam & Sue, 1997)
- Minimum, maximum, average and OWA aggregating operators (Kittler, 1998, Kuncheva, 1997)
- Bayesian (Naïve-Bayes) decision rule (Xu, 1992)
- Fuzzy aggregation (Cho & Kim, 1995, Wang et al., 1998)
- Decision templates (Kuncheva et al., 2001)
- Meta-learning techniques (Chan & Stolfo, 1993, Wolpert, 1994, Prodromidis et al., 1999)
- Bagging (Breiman, 1998)
- Boosting (Freund & Schapire, 1998)
- Random forests (Breiman, 2001)
- ECOC ensembles (Dietterich and Bakiri, 1995)

See ***L. Kuncheva Combining Pattern Classifiers, Wiley, 2004*** for a good review book on ensemble methods

Hierarchical ensemble methods

They are in general characterized by a two-step strategy:

1. Flat learning of the protein function on a per-term basis (a set of independent classification problems)
2. Combination of the predictions by exploiting the relationships between terms that govern the hierarchy of the functional classes.

The term *ensemble* raises from the fact that a set of learning machines in someway combine their output.

In principle any supervised learning algorithm can be used for step 1.

Step 2 requires a proper combination of the predictions made at step 1.

Hierarchical ensemble methods

- Bayesian network-based ensembles (*Barutcuoglu et al*. 2006, *Guan et al*. 2008)
- Hierarchical renconciliation methods (*Obozinski et al*. 2008)
- Hierarchical decision trees (*Vens et al*. 2008, *Schietgat et al* 2010)
- Hierarchical Bayesian cost-sensitive ensembles (*Cesa-Bianchi and Valentini*, 2010)
- True Path Rule Ensembles for trees (*Valentini*, 2011)
- True Path Rule Ensembles for DAGs (*Notaro et al*. 2017)

Hierarchical Bayesian network-based prediction of gene function

(Barutcuoglu, Schapire and Troyanskaya, 2006)

Main ideas:

- *Flat prediction* of each term/class (possibly inconsistent)
- *Bayesian hierarchical combination* scheme to allow collaborative error-correction over all nodes

Basic notation:

y_i : binary membership to class *i*

$\hat{y}_i$: classifier output for class *i*, $1 \le i \le N$

Bayesian correction of classifier outputs

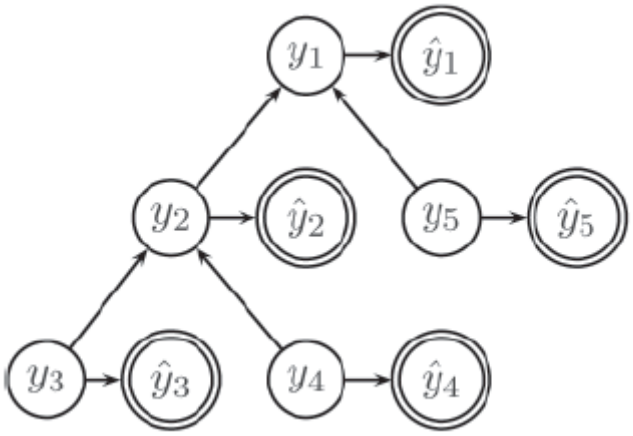
Goal: given a set of (possibly inconsistent) $\hat{y}_i$
find the set of consistent y_i that maximize:

$$P(y_1,\ldots,y_N \mid \hat{y}_1,\ldots,\hat{y}_N) = \frac{P(\hat{y}_1,\ldots,\hat{y}_N \mid y_1,\ldots,y_N)P(y_1,\ldots,y_N)}{Z}$$

Direct solution is too hard … (exponential in time w.r.t to the number of nodes)

Proposed solution: *a Bayesian network structure that exploits the relationships between functional classes*.

The proposed Bayesian network



1. y_i nodes conditioned to their children (structure constraints)
2. $\hat{y}_i$ nodes conditioned on their label y_i (Bayes rule)
3. $\hat{y}_i$ are independent from both $\hat{y}_j, j \neq i$ and $y_j, j \neq i$ given y_i

This allows us to simplify the Bayesian equation:

from 1: $$P(y_1, \ldots, y_N) = \prod_{i=1}^{N} P(y_i \mid ch(y_i))$$

from 2,3: $$P(\hat{y}_1, \ldots, \hat{y}_N, \mid y_1, \ldots, y_N) = \prod_{i=1}^{N} P(\hat{y}_i \mid y_i)$$

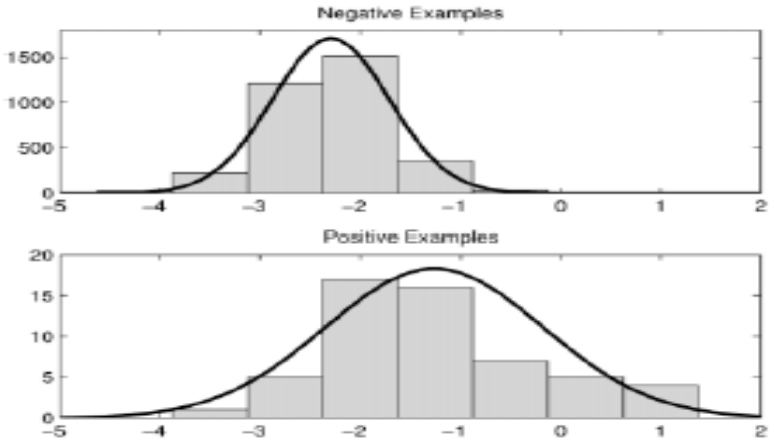
Estimation of the probabilities

Estimation of $P(y_1, \ldots, y_N) = \prod_{i=1}^{N} P(y_i \mid ch(y_i))$

Can be inferred from training labels by counting

Estimation of $P(\hat{y}_1, \ldots, \hat{y}_N, \mid y_1, \ldots, y_N) = \prod_{i=1}^{N} P(\hat{y}_i \mid y_i)$

Can be inferred by validation during training, by modeling the distribution of $\hat{y}_i$ outputs over positive and negative examples.
E.g.: a parametric gaussian model:



Implementation of the method

- *Bagged ensemble of SVMs* (10 SVMs) trained at each node (see next slide …)
- Median values of their outputs on out-of-bag examples have been used to *estimate means and variances for each class*.
- Mean and variances have been used as parameters of the *gaussian models used to estimate the conditional probabilities* $P(\hat{y}_i | y_i = 1)$ and $P(\hat{y}_i | y_i = 0)$

The prediction of the label for each class i is then computed as follows:

$$P(y_1,\ldots,y_N \mid \hat{y}_1,\ldots,\hat{y}_N) = \frac{\prod_{i=1}^{N} P(\hat{y}_i \mid y_i) P(y_i \mid child(y_i))}{Z}$$

Bagging (**B**ootstrap **agg**regat**ing**)

(Breiman, 1996)

Input: $Z=\langle (x_1,y_1),\ldots,(x_m,y_m)\rangle$ a base learner: *LearnAlg*

Do for t=1 to T:

1. Bootstrap replicate Z_t from Z
   (random sampling with replacement)
2. Get back an hypothesis h_t:X ->Y
   h_t = LearnAlg(Z_t)

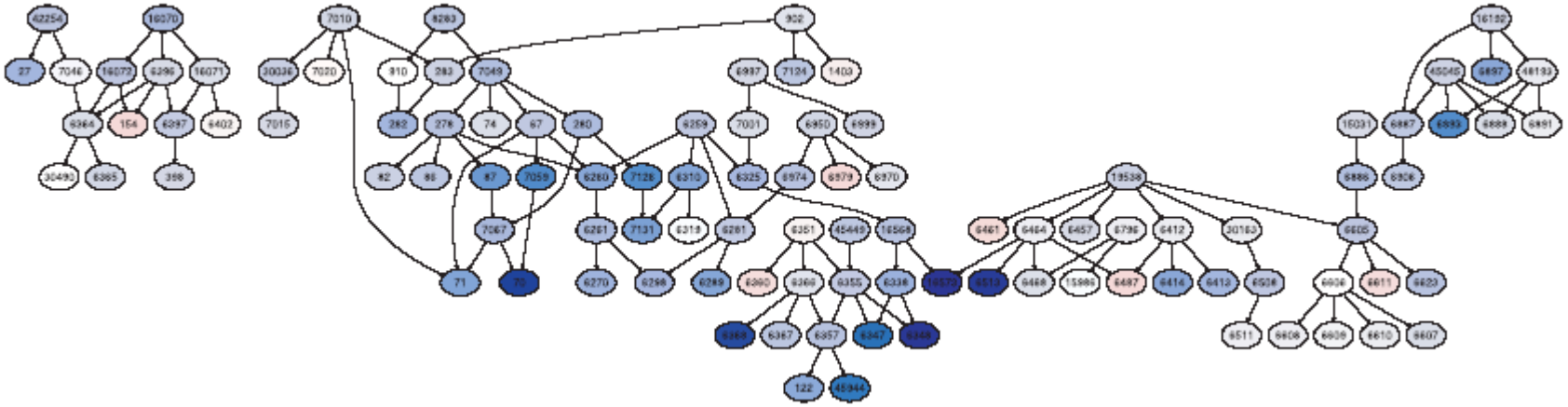
end for

Output the final hypothesis by aggregation and majority voting**:**

$$\sum_{t=1}^{T} \begin{cases} 1 & if \quad h_t(x)=y \\ 0 & otherwise \end{cases}$$

- Effective with unstable algorithms
- It reduces the variance component of the error

Results on a sub-hierarchy of the BP GO ontology



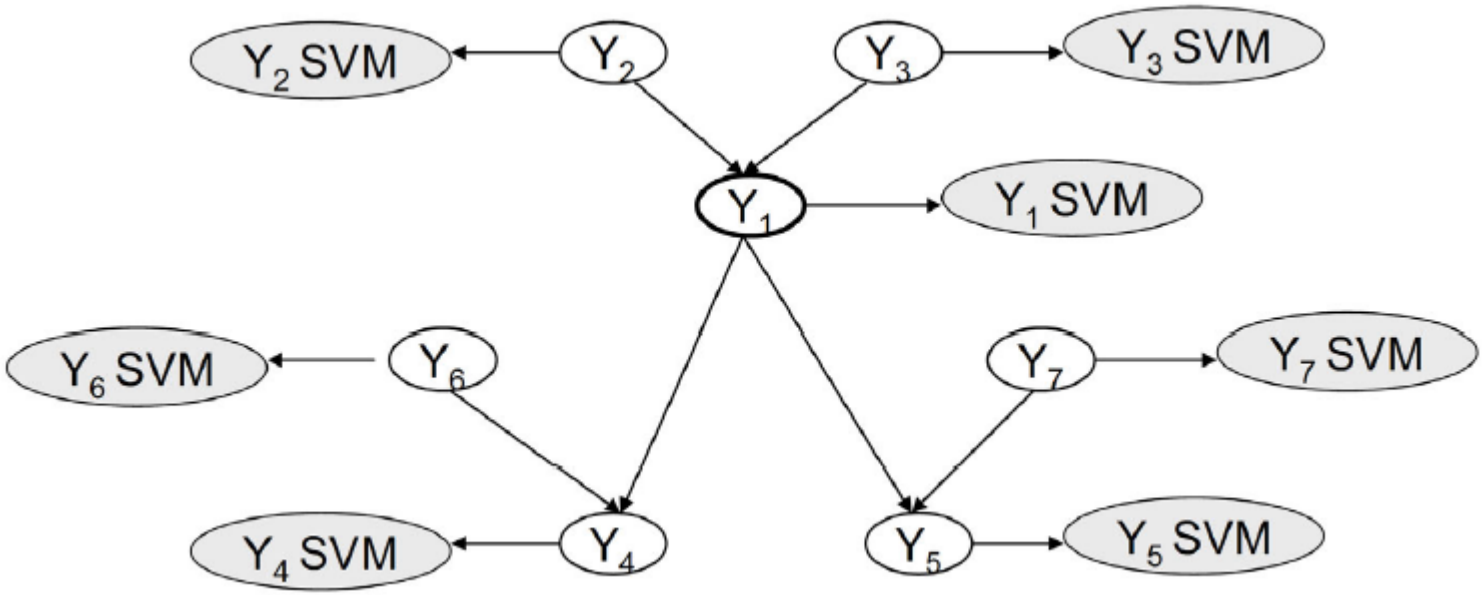
- 105 terms/nodes of the GO BP (model organism *S.cerevisiae*)
- 4 types of data integrated through Vector Space Integration
- Hierarchical approach improves AUC results on 93 of the 105 GO terms
- Darker blue: improvements; darker red: deterioration; white: no change.

Improvements of the algorithm

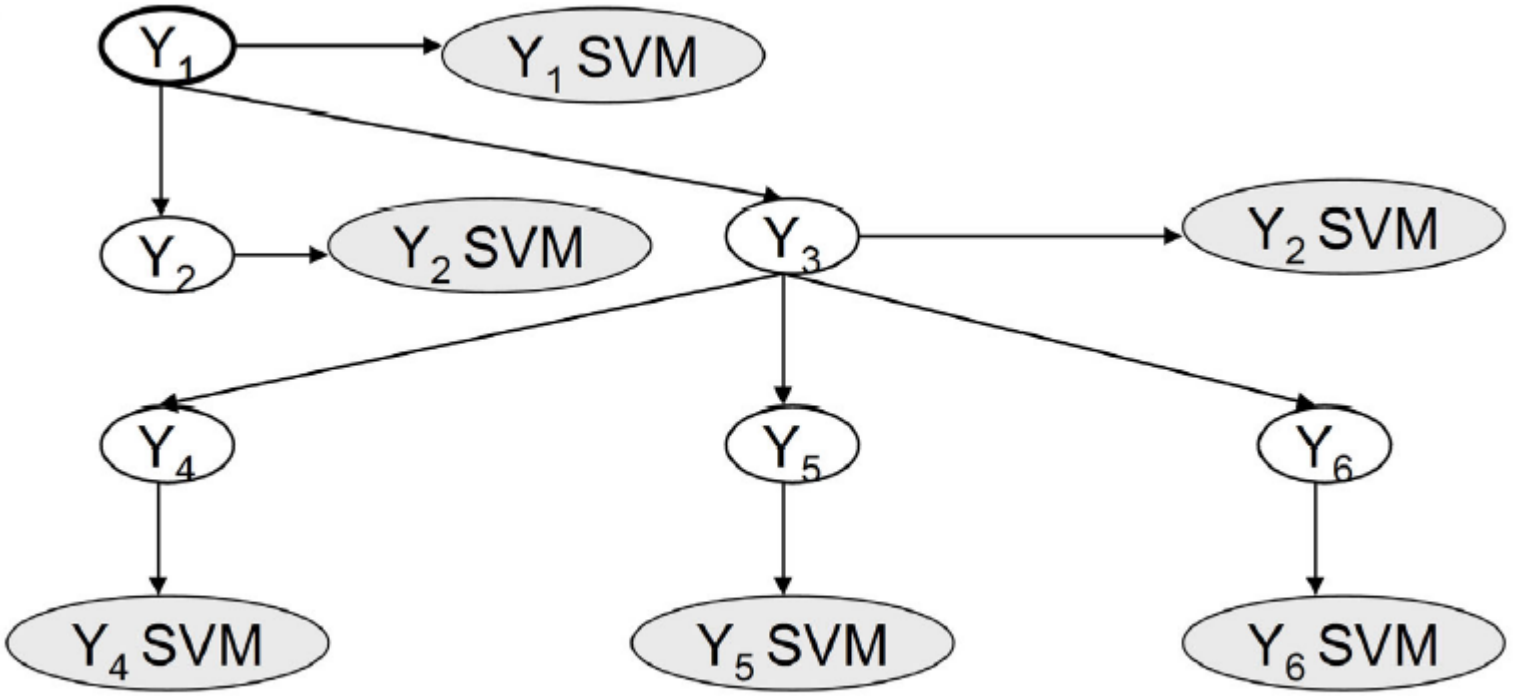
Guan, Myers, Hess, Barutucuoglu, Caudy and Troyanskaya, 2008

- Two variants of the Bayesian integration:
  - HIER-MB: Hierarchical Bayesian combination involving nodes in the Markov Blanket
  - HIER-BFS: Hierarchical Bayesian combination involving nodes the 30 first nodes visited through a Breadth-First-Search (BFS) in the GO graph
- Integration of 3 classifiers selected through held-out examples
- Application to the prediction of *M. musculus* (mouse) gene functions

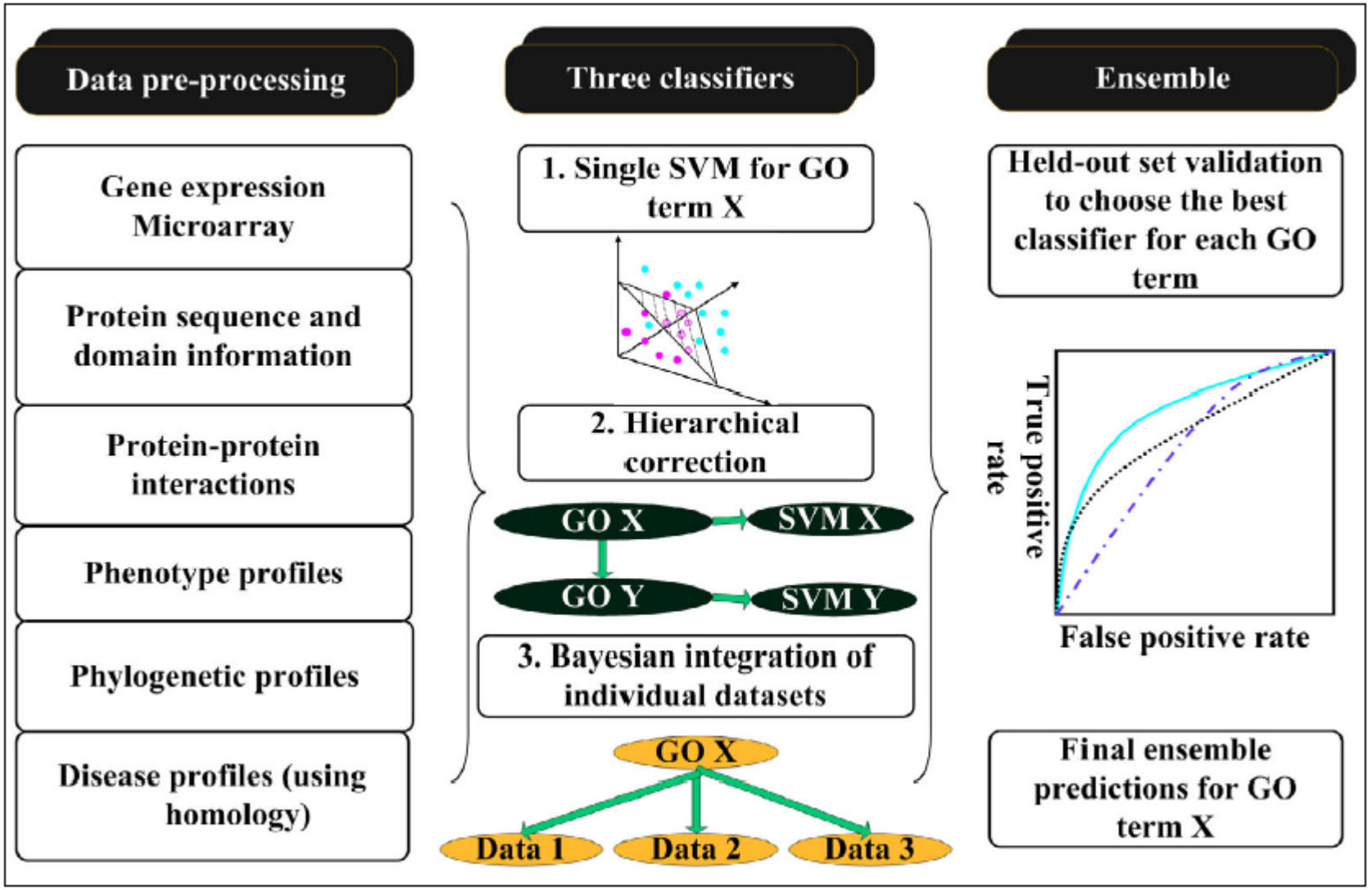
HIER-MB: Hierarchical Bayesian combination involving nodes in the Markov Blanket



HIER-BFS: Hierarchical Bayesian combination using the first 30 BFS nodes



Ensemble of 3 classifiers selected through held-out examples



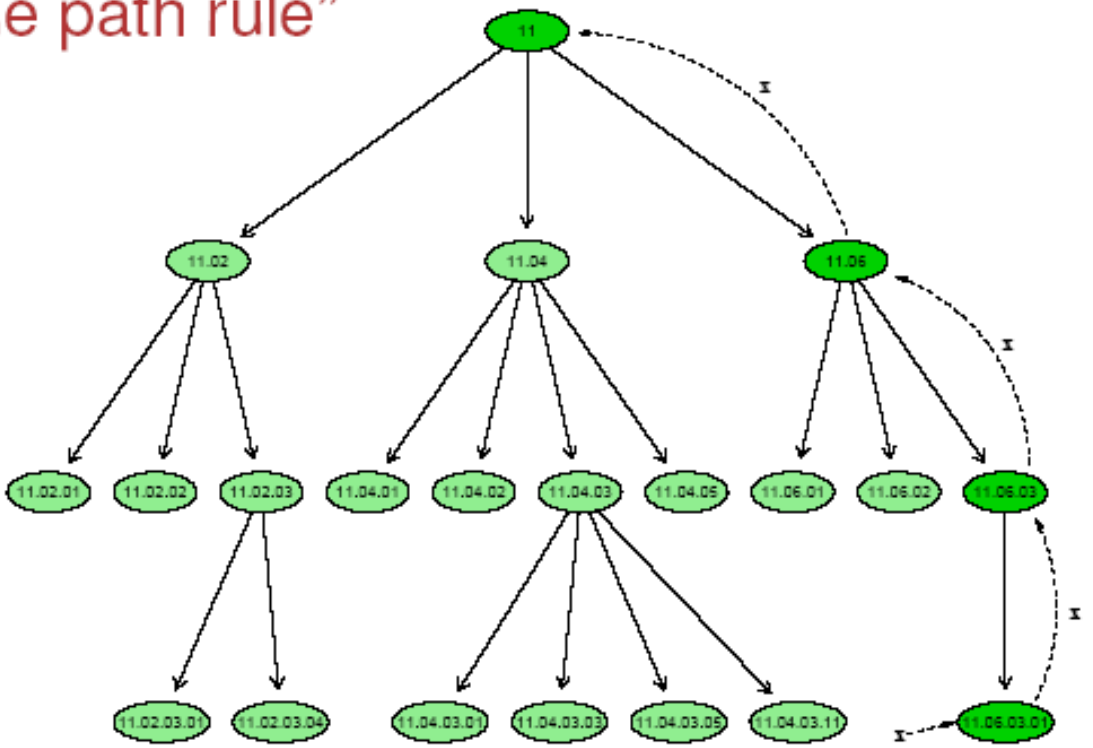
Main limitations of the Princeton group approach

Main drawbacks:

- Hierarchical integration is local (limited to the Markov blanket and the first 30 BFS nodes)
- Integration strategy: other works showed that methods other than VSI work better (e.g. Kernel fusion (*Lanckriet et al., 2004*), ensemble methods (*Re and Valentini, 2010*)).
- The approach does not take into account the unbalance between positive and negative examples.

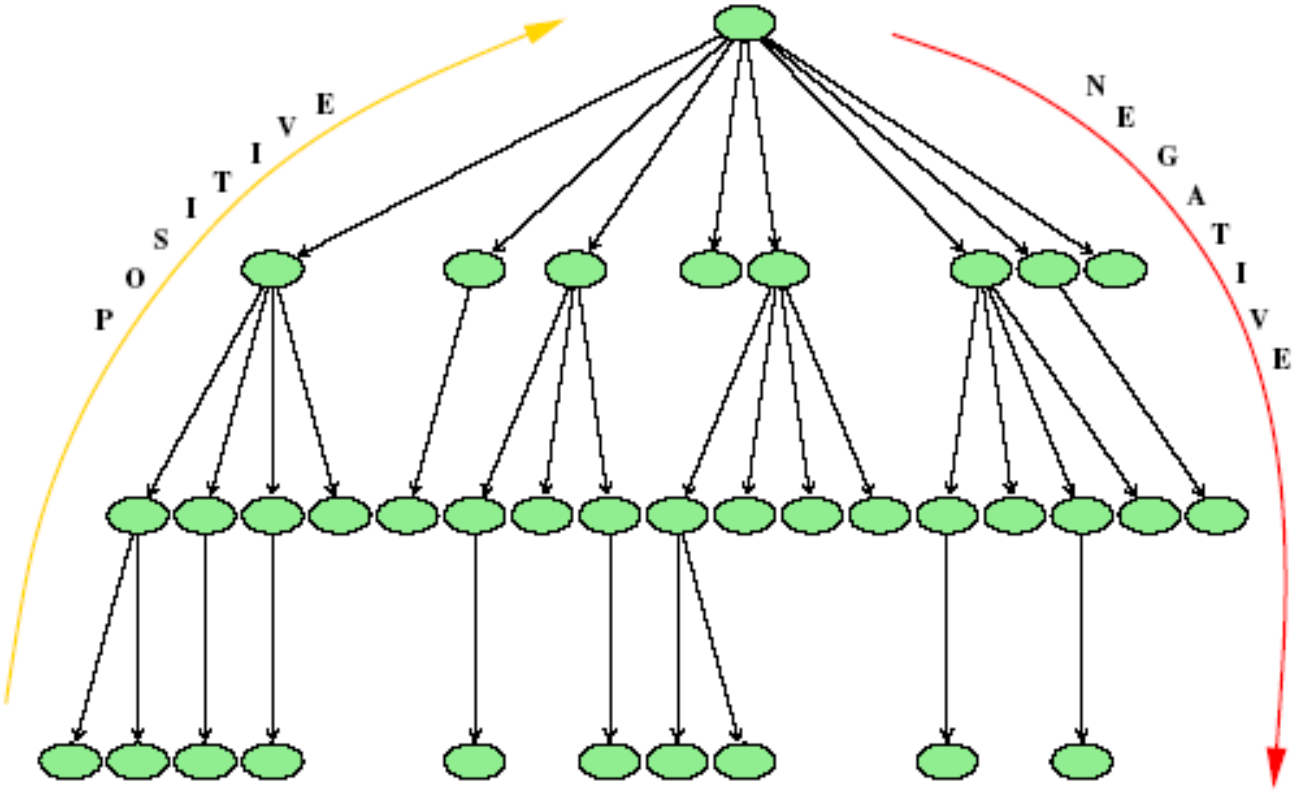
An approach based on the “true path rule”

The “true path rule”



“An annotation for a class in the hierarchy is automatically transferred to its ancestors, while genes unannotated for a class cannot be annotated for its descendants”.

True Path Rule ensembles (*Valentini*, 2011): an asymmetric flow of information



From bottom to top : positive predictions influence ancestor nodes/classifiers

From top to bottom : negative predictions influence descendant nodes/classifiers

TPR ensemble for tree-structured ontologies: more in the next lecture ...

References (1)

Astikainen, K., Holm, L., Pitkanen, E., Szedmak, S., and Rousu, J. (2008). Towards structured output prediction of enzyme function. *BMC Proceedings*, 2(Suppl 4:S2).

Barutcuoglu, Z., Schapire, R., and Troyanskaya, O. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7), 830–836

Belkin, M, Matveeva, I, Niyogi, P. (2004) Regularization and semi-supervised learning on large graphs. In COLT

Bengio, Y., Delalleau, O., and Le Roux, N. (2006). Label Propagation and Quadratic Criterion. In O. Chapelle, B. Scholkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press.

Bertoni, A., Frasca, M., Valentini G. (2011) COSNet: a Cost Sensitive Neural Network for Semi-supervised Learning in Graphs., *European Conference on Machine Learning 2011, Athens, Lecture Notes in Computer Science, Springer*

Cesa-Bianchi, N. and Valentini, G. (2010). Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *Journal of Machine Learning Research, W&C Proceedings, Machine Learning in Systems Biology*, 8, 14–29.

Cesa-Bianchi, N., Re, M., and Valentini, G. (2010). Functional inference in FunCat through the combination of hierarchical ensembles with data fusion methods. In *ICML-MLD 2nd International Workshop on learning from Multi-Label Data*, pages 13–20, Haifa, Israel.

Delalleau, O., Bengio, Y, Le oux, N (2005) Efficient non-parametric function induction in semi-supervised learning. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005.

Deng, M., Chen, T., and Sun, F. (2004). An integrated probabilistic model for functional prediction of proteins. *J. Comput. Biol.*, 11, 463–475.

Friedberg, I. (2006). Automated protein function prediction-the genomic challenge. *Brief. Bioinformatics*, 7, 225–242

Guan, Y., Myers, C., Hess, D., Barutcuoglu, Z., Caudy, A., and Troyanskaya, O. (2008). Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, 9(S2).

Karaoz, U. *et al.* (2004). Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc. Natl Acad. Sci. USA*, 101, 2888–2893.

References (2)

Marcotte, E., Pellegrini, M., Thompson, M., Yeates, T., and Eisenberg, D. (1999). A combined algorithm for genome-wide prediction of protein function. *Nature*, 402, 83–86.

Mostafavi, S. and Morris, Q. (2010). Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics*, 26(14), 1759–1765.

Mostafavi, S., Ray, D.,Warde-Farley, D., Grouios, C., and Morris, Q. (2008). GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9(S4).

Obozinski, G., Lanckriet, G., Grant, C., M., J., and Noble, W. (2008). Consistent probabilistic output for protein function prediction. *Genome Biology*, 9(S6).

Oliver, S. (2000). Guilt-by-association goes global. *Nature*, 403, 601–603.

Pavlidis, P.,Weston, J., Cai, J., and Noble,W. (2002). Learning gene functional classification from multiple data. *J. Comput. Biol.*, 9, 401–411.

Pena-Ca stillo, L., et al. (2008): A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. Genome Biology 9 S1

Re, M. and Valentini, G. (2010). Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction. *Journal of Machine Learning Research, W&C Proceedings, Machine Learning in Systems Biology*, 8, 98–111.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7, 1601–1626.

Schietgat, L., Vens, C., Struyf, J., Blockeel, H., and Dzeroski, S. (2010). Predicting gene function using hierarchical multilabel decision tree ensembles. *BMC Bioinformatics*, 11(2).

Sharan, R. Ulitsky, I.Shamir, R. (2007) Network-based prediction of protein function, Molecular Systems Biology 3:88

Sokolov, A. and Ben-Hur, A. (2010). Hierarchical classification of Gene Ontology terms using the GOstruct method. *Journal of Bioinformatics and Computational Biology*, 8(2), 357–376.

References (3)

Szummer, M Jaakkola, T. (2001) Partially labeled classication with markov random walks. In NIPS, volume 14.

Tsochantaridis, I., Joachims, T., Hoffman, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6, 1453–1484.

Tsuda, K., Shin, H., and Scholkopf, B. (2005). Fast protein classification with multiple networks. *Bioinformatics*, 21(Suppl 2), ii59–ii65.

Valentini, G. and Cesa-Bianchi, N. (2008). Hcgene: a software tool to support the hierarchical classification of genes. *Bioinformatics*, 24(5), 729–731.

Valentini, G. (2011), True Path Rule hierarchical ensembles for genome-wide gene function prediction, *IEEE ACM Transactions on Computational Biology and Bioinformatics*, 8(3), 832-847.

Vazquez, A., Flammini, A., Maritan, A., and Vespignani, A. (2003). Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 21, 697–700.

Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., and Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2), 185–214.

Zhou, D., et al. (2004) Learning with local and global consistency. In NIPS, volume 16

Zhu, X. Ghahramani, Z. , Laerty J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In ICML.