

Generalizability of Machine Learning Methods in Bioinformatics

Jeremy Purvis

April 27, 2006

It appears that the only true requirement for a learning machine is that at some point in time, under some conditions, for some particular data set—the method must have worked at least once. This observation might explain why the number of machine learning methods has become so enormous, or, perhaps more optimistically, indicates that a variety of approaches will be necessary to deal with the computational problems facing bioinformaticians. In this paper, I consider some of the more creative applications of machine learning to bioinformatics published within the past year. It is by no means a comprehensive survey; instead, the focus will be on the *generalizability* of the methods examined. That is, after training the model, how well does it handle new data [1].

Supervised Methods. A major limitation of supervised learning methods is that they are trained on small or unrepresentative data sets. Take microRNAs as an example. To date, there are about 500 known microRNA (miRNA) sequences, yet this number is expected to increase to several thousand in short time [2]. As the search for more miRNA sequences continues, it is likely that predictions will be biased toward sequences that resemble already established miRNA types. To make matters worse, many algorithms limit their search to conserved genomic regions, or reduce the sensitivity of their detection in order to control the number of false detections. Both of these approaches neglect valuable hypothesis space and thereby decrease generalizability. Yousef *et al* have dealt with this problem by integrating data from multiple species in hopes of constructing a more general detection model [3]. They used a naïve Bayes classifier to minimize the likelihood $P(X|C) = P(x_1, \dots, x_n|C)$, where $P(X|C)$ is the product of the likelihood of each datum in the feature vector (x_1, \dots, x_n) belonging to class C . Compared

to other approaches, their model shows improved generalization by (i) considering a variety of diverse sequence features, such as length, bulges, loops, and motifs; (ii) carefully choosing negative examples with which to train the classifier (here, they used cross-species putative miRNAs that failed exactly one structural feature test); and (iii) combining multiple sets of miRNAs in the training data.

Another improvement to feature space was made by Lee *et al* [4], who improved the prediction of protein secondary structure by incorporating evolutionary information into their calculations. They compared the calculated homolog-averaged amino acid composition from each protein $P_i(X) = (\sum_{j=1}^{L_i} S_i(j, X))/L_i$, gathered from multiple alignments, to the background probability of each amino acid $P_0(X) = (\sum(N_k(X)/L_k))/3352$. Multiple linear regression was applied to the computed results to produce a training data set, which was then used to train an artificial neural network. Finally, classification was performed using a support vector machine with dot and radial kernel functions. An impressive error rate of 3.3% suggests that incorporating homologous sequence information can improve the generalizing capacity of secondary structure prediction machines.

The support vector machine (SVM) method has become a fashionable way to classify biological data [5, 6]. The power of SVMs lies in their ability to map features into arbitrarily complex spatial dimensions to find the optimal margin of separation. While the basic SVM discriminates quite well, Nguyen and colleagues have built upon this capability by combining multiple kernel functions that are customized to solve a particular classification problem [7]. During the training phase, an evolutionary algorithm is used to optimize the decision model, whose fitness is evaluated by measuring the number of hits the SVM would generate under the selected model. Increasingly fit models are evolved in this manner to produce an ‘optimal decision model’, which is used to build an SVM for the classification of novel samples. This hybrid approach affords not only an optimized kernel function, but an optimized feature space. When classifying expression data sets from leukemia and colon cancers, their method consistently produced a more stable and higher recognition rate than that achieved with a single kernel function.

An unfortunate downside to SVMs is that they often warp the feature space in ways that are *incomprehensible*, or difficult to relate intuitively. He *et al* stress the importance of preserving comprehensibility in computational schemes, claiming it is necessary for ‘advanced deduction’ and driving laboratory experimentation [8]. To address this concern, they propose a method that proceeds by first training an SVM and then using the output

to train a decision tree learning system. Specifically, a training data set $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ is divided into N similarly distributed subsets, of which all but one are used to train an SVM, with the missing subset used as test data. This is done in turn for each subset, and successful test data are then compiled into a new data set S_{SVM}^i , which is used to train the decision tree. Their method was used to predict transmembrane segments [8] and protein secondary structure [9]. An average of 90% prediction accuracy show that this distilled data set is better indicator for rule generation than decision trees alone, and comparable to that of SVMs. By utilizing this combined approach, they have produced a more comprehensible learning scheme while retaining the superior generalizing ability of SVMs.

As an aside, one should note that incomprehensibility refers to an *apparent* lack of rationality in a machine learning process. If scientists had nothing more to learn about biological systems, then perhaps we could dismiss these incomprehensible schemes as frivolous or nonsensical (or computationally lucky). Rather, one can imagine a situation in which the logistics of an optimized SVM mapping could be examined *post facto* in order to learn something about the nature of the feature space, or about its relation to the classification problem at hand. Particularly for low-criteria supervised problems, such as the one described above for microRNAs, incomprehensible methods like SVM may be the only feasible method for generalizable detection, especially when precious little positive training data exists.

Unsupervised Methods. A common drawback to machine learning approaches is that they often provide *too many* solutions. When the hypothesis space \mathcal{H} is large, and particularly when supervised criteria are small, an unsupervised method may generate multiple solutions to a problem that cannot all be correct [10]. In a recent paper appearing in *IEEE/ACM Transactions* [11], Kaski *et al* sidestepped this problem by looking at commonalities *within* respective data sets. By symmetric dependency modeling, they were able to form associative clusters (AC) between two data sets in such a way that dependencies between sample pairs was maximized. Data pairs (x_k, y_k) were partitioned into clusters so to maximize the Bayes factor,

$$BF = \frac{p(n_{ij}|M_D)}{p(n_{ij}|M_I)}, \quad (1)$$

where n_{ij} are co-occurring frequencies and M_D and M_I are the dependent and independent margins, respectively. This calculation was applied to expression profiles of orthologous man-mouse gene pairs to identify similarities

and differences in regulatory patterns. Although K-means clustering on individual data sets produced more clusters, AC provided better cluster homogeneity, which is measured by intracluster variance. This method promises high generalizability because of its minimal set of initial assumptions and parameterizations.

Clustering methods are also useful for sorting out crude patterns among noisy data. The words ‘crude’ and ‘noisy’ hold particular prominence in the world of gene expression data where, not surprisingly, clustering methods are employed religiously. Although clustering is inherently generalizable (given a decent measure of feature distance), the most commonly used algorithms often require a predetermined choice of clusters to operate effectively, or cannot handle large or incomplete data sets. By imposing a formal preprocessing step (non-linear principal component analysis) on yeast microarray data, Amato *et al* were able to filter out noise and extract the dominant correlation features from their expression data [12]. The results were then subject to Negentropy clustering, which chooses the number of clusters based on the differential entropy of a set of vectors $\mathbf{y} = (y_1, \dots, y_n)^T$ with density $f(\mathbf{y})$, defined as

$$H(y) = - \int f(y) \log(f(y)) dy. \quad (2)$$

In terms of generalizability, the most encouraging finding from this study was the uniqueness of clusters. Large gene clusters were discovered that belonged to separate cellular processes (as assessed by yeast Gene Ontology Database), including some sets that showed no gene overlap at all.

As seen, extracting features from gene expression data is a quantitative necessity because of the formidably high dimensionality of the technology. Pal *et al* sought to make feature selection of expression data an ‘online’ task—something that can change *while* the machine learns the classification task [13]. Their approach is interesting. Input expression data is fed to a multilayer perceptron (MLP) with the following property: input nodes are initially kept near-closed (gate function $F(m) \approx 0$) unless an input feature can reduce the error term for that node. Thus, only potentially useful features are identified and weighted according to their usefulness. When they applied this scheme to discriminate expression data from two forms of leukemia, they were able to reduce the list of relevant features to just five genes. These five inputs were then fed into a standard three-layer MLP. Their results (100% accuracy on the training set and 97.1% accuracy on the test data) show that the classification is highly dependent on feature selection. Furthermore, the minimal assumptions inherent in this model (all initial MLP weights are approximately zero) make it highly generalizable.

Alternative Learning Schemes. Computational biology has borrowed a great deal of machine learning methodology from other academic disciplines. In a recent study by Cheng *et al*, the problem of identifying trans-membrane helix boundaries in G-protein coupled receptors (GPCRs) was tackled using a text-inspired learning approach [14]. Following a linguistic model, the intracellular, extracellular, and helical portions of each GPCR were treated as separate ‘topics’ that were distinguished by discontinuities against a background model. This created segmentation in sequences, which, in Cheng’s case, were comprised of an interpolation of six basic probability models. Each model consisted of a lexicon of ‘words’ which represented constant-sized (one to ten residues) strings of amino acids. To construct the most likely models, a running average of the log probabilities of these strings was determined, with the requirement that the next segment must outperform the current segment model at every position to be labeled a segment boundary. The method was performed on the full set of known GPCRs and evaluated by ten-fold cross validation. Although average offset was about two residues from the ‘true’ boundary [15], a significant cluster of large offsets was observed for about 10% of the proteins. Thus, while the unsupervised method is generalizable to the majority of GPCRs, there is a significant fraction that are not well parsed by this machine text approach.

A common theme in these studies is the combination of two or more machine learning methods in a single experiment. Moreover, individual methods themselves are often borrowed or adapted from other academic disciplines or applications. As the strengths and weaknesses of each method become clearer, an appropriate learning method can be chosen for analysis during a particular phase of experimentation. For example, we have seen that a preprocessing phase can be helpful in reducing noise or removing extraneous data. For this purpose, one might use principal component analysis or a neural network. Subsequently, another learning machine, such as SVM or clustering, could be used for classification or regression of the filtered data. It appears that machine learning, like its human counterpart, will use a blend of diverse, unusual, and often unintuitive schemes to carry out the task of intelligent function.

References

- [1] T. POGGIO, R. RIFKIN, S. MUKHERJEE, and P. NIYOGI, *Nature* **428**, 419 (2004).

- [2] V. N. KIMA and J.-W. NAM, *Trends in Genetics* **22**, 165 (2006).
- [3] M. YOUSEF, M. NEBOZHYN, H. SHATKAY, S. KANTERAKIS, L. C. SHOWE, and M. K. SHOWE, *Bioinformatics Advanced Access* (2006).
- [4] K. D. LEE S, LEE BC, *Proteins* **62**, 1107 (2006).
- [5] F. ROSSI and N. VILLA, *Neurocomputing* **69**, 730 (2006).
- [6] M. BHASIN, E. L. REINHERZ, and P. A. RECHE, *Journal of Computational Biology* **13**, 102 (2006).
- [7] H.-N. NGUYEN, S.-Y. OHN, J. PARK, and K.-S. PARK, Combined Kernel Function Approach in SVM for Diagnosis of Cancer, in *ICNC (1)*, pp. 1017–1026, 2005.
- [8] J. HE, H. J. HU, R. HARRISON, P. C. TAI, and Y. PAN, *Expert Systems with Applications* **30**, 64 (2006).
- [9] J. Y. HE, H. J. HU, R. HARRISON, P. C. TAI, and Y. PAN, *IEEE Transactions On Nanobioscience* **5**, 46 (2006).
- [10] N. FRIEDMAN, M. LINIAL, I. NACHMAN, and D. PE'ER, *Journal of Computational Biology* **7**, 601 (2000).
- [11] S. KASKI, J. NIKKILÄ, J. SINKKONEN, L. LAHTI, J. KNUUTTILA, , and C. ROOS, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **2**, 203 (2005).
- [12] R. AMATO, A. CIARAMELLA, N. DENISKINA, C. D. MONDO, D. DI BERNARDO, C. DONALEK, G. LONGO, G. MANGANO, G. MIELE, G. RAICONI, A. STAIANO, and R. TAGLIAFERRI, *Bioinformatics* **22**, 589 (2006).
- [13] N. R. PAL, A. SHARMA, S. K. SANADHYA, and KARMESHU, *International Journal of Intelligent Systems* **21**, 453 (2006).
- [14] B. Y. M. CHENG, J. G. CARBONELL, and J. KLEIN-SEETHARAMAN, A Machine Text-Inspired Machine Learning Approach for Identification of Transmembrane Helix Boundaries., in *ISMIS*, pp. 29–37, 2005.
- [15] F. HORN, J. WEARE, M. W. BEUKERS, S. HORSCH, A. BAIROCH, W. CHEN, O. EDVARDBSEN, F. CAMPAGNE, and G. VRIEND, *Nucleic Acids Research* **26**, 275 (1998).