Combining Abstraction and SuperStructuring on Macromolecular Sequence Classification

Adrian Silvescu[1], Cornelia Caragea[2], Vasant Honavar[3]

1 Introduction

Representational commitments i.e., the choice of features or attributes that are used to describe the data presented to a learner, and the level of detail at which they describe the data, can have a major impact on the difficulty of learning, and the accuracy, complexity, and comprehensibility of the learned predictive model. The representation has to be rich enough to capture the distinctions that are relevant from the standpoint of learning, but not so rich as to make the task of learning infeasible due to overfitting.

We present an approach to exploiting the complementary strengths of feature construction (constructing complex features by combining existing features) and feature abstraction (grouping of similar features to generate a more abstract feature) or feature selection to adapt the data representation used by the learner. In particular, consider a special case of topologically constrained feature construction, namely, super-structuring. Super-structuring provides a way to increase the predictive accuracy of the learned models by enriching the data representation (and hence increasing the complexity of the learned model) whereas abstraction or selection help reduce the model size by simplifying the data representation.

2 Combining Abstraction and Super-structuring

The classification problem: Given a data set $\mathcal{D}= \{d^i = (s^i, c^i)\}_{i=1,n}$ where s^i is a sequence over a finite alphabet $G = \{g_1, \cdots, g_t\}$, that is $s^i \in G^*$ and c^i is the class associated with the sequence s^i that belongs to a finite set C, $c^i \in C$, the learning algorithm is asked to produce a model that is able to predict the class c for a novel sequence s [3].

Let s be a sequence over the alphabet G. The *super-structures* of sequence s are all the contiguous (potentially overlapping) sub-sequences of a certain length, a.k.a. k-grams [1].

The idea of our algorithm is to first create an expanded feature set (the k-gram set, $|\mathcal{KG}|= m$) from the sequences in $\mathcal{D}$ using *super-structuring* in order to improve the modeling performance, and then to shrink down this feature set in order to reduce the model size to a fixed number of features n_f. The reduction can be accomplished in two ways:

- (ABS) by *constructing abstractions* over the k-grams: partitioning the set of k-grams into n_f non-overlapping sets $ABS = \{a_1 : set_1, \cdots, a_{n_f} : set_{n_f}\}$ where a_i denotes the label for the i-th abstraction and set_i denotes the set of k-grams which are grouped together into the i-th abstraction.
- (FSEL) by *feature selection* over the k-grams: selecting a set of n_f k-grams $FSEL \subseteq \{kg_1, \cdots, kg_m\}$.

2.1 Constructing Abstractions

Our algorithm for *constructing abstractions* works as follows: we start by initializing each abstraction by a primary feature (k-gram). Then we recursively group the abstractions until we obtain n_f abstractions. Specifically, $m - n_f$ times we find a pair of abstractions that are most "similar" to each other, group them, add the abstraction resulting from their union to the set of abstractions ABS and delete the individual abstractions from ABS. After $m - n_f$ steps, the result of our algorithm is a set of n_f groups/abstractions.

In order to complete the description of our algorithm we need to show how to compute the similarity measure between two abstractions. Our general criteria for establishing similarity between items and thus deriving useful abstractions is based on the following *functionalist* claim: *similar items occur within similar contexts.* Thus, one way to define the similarity between two abstractions is to specify what we mean by the context of an abstraction a_j and then define a distance between these contexts.

[1]Yahoo! Labs. E-mail: silvescu@yahoo-inc.com
[2]Computer Science Department, Iowa State University, IA, USA. E-mail: cornelia@cs.iastate.edu
[3]Computer Science Department, Iowa State University, IA, USA. E-mail: honavar@cs.iastate.edu

Class Context for Abstractions. Given an abstraction $a_j = \{f_{j_1}, \cdots, f_{j_q}\}$ we define

$$CContext_D(a_j = \{f_{j_1,...,}f_{j_q}\}) := [\#a_j, P(C|a_j)] = \left[\sum_{l=1}^{q} \#f_{j_l}, \sum_{l=1}^{q} \pi_l P(C|f_{j_l})\right], \text{where } \pi_{_l} := \frac{\#f_{j_l}}{\sum_{l=1}^{q} \#f_{j_l}}$$

Thus, the Class Context of an abstraction $a_j = \{f_{j_1}, \cdots, f_{j_q}\}$ is the sum of the frequency counts of the features f_{j_l} in the data set $\mathcal{D}$ along with the weighted sum of the conditional probability of the class given the features f_{j_l}, $P(C|f_{j_l})$.

Distance Between Abstractions. Let a_i and a_j be two abstractions. We use the Weighted Jensen-Shannon Distance (WJS) [2] between two probability distributions to define the distance between two abstractions.

$$D(a_i, a_j) \quad := \quad WJS(CContext(a_i), CContext(a_j)) = WJS([\#a_i, P(C|a_i), [\#a_j, P(C|a_j)])$$

2.2 Feature Selection

Feature Selection is performed by selecting a set of nf features $FSEL \subseteq \{f_1, ..., f_m\}$, $|FSEL| = n_f$. The features are ranked according to a scoring function $Score$ and then the top n_f best ranked features are selected. We used information gain [3] between the probability of the class variable $P(C)$ and the probability of the feature f, which measures how dependent the two variables are.

3 Experiments and Results

We performed experiments that compare Naive Bayes Multinomial classifiers constructed using the original features with those constructed using feature selection, feature abstraction, and the combination of abstraction and super-structuring and feature selection and super-structuring on two datasets from the bioinformatics domain: **Eukaryotes** and **Prokaryotes**. The tasks are to classify sequences according to their subcellular localization.

The results of our experiments on both data sets show that adapting data representation by combining abstraction and super-structuring makes possible to construct predictve models that use significantly smaller (1-3 orders of magnitude) number of features than those that are obtained using super-structuring alone without sacrificing predictive accuracy (Figure 1).

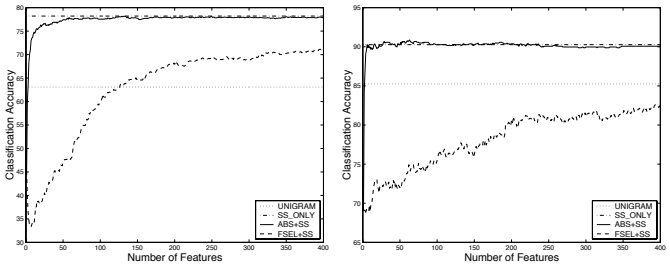

Figure 1: Comparison of Abstraction+SuperStructuring (ABS+SS) with Feature Selection+SuperStructuring (FSEL+SS), SuperStructuring only (SS_ONLY), and Unigram (UNIGRAM) on the **Eukaryotes** (left) and **Prokaryotes** (right) data sets using unigrams and 3-grams. For the same number of features used to train the classifiers, ABS+SS is superior in performance to FSEL+SS, and UNIGRAM. After a relatively small number of features, ABS+SS achieves the performance of SS_ONLY. For a small drop in performance on both **Eukaryotes** and **Prokaryotes**, we obtain a a reduction of model sizes by three orders of magnitude.

References

[1] Eugene Charniak. *Statistical Language Learning, Cambridge: 1993.* MIT Press, 1993.

[2] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37:145–151, 1991.

[3] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.