String Transformation-Based Bayesian Classification or Proteins

Timothy Meekhof
University of Idaho
Moscow, Idaho, USA
timothym@cs.uidaho.edu

Gary W. Daughdrill
University of Idaho
Moscow, Idaho, USA
gdaugh@uidaho.edu

Robert B. Heckendorn
University of Idaho
Moscow, Idaho, USA
heckendo@uidaho.edu

ABSTRACT

We describe a Markov chain Bayesian classification tool, SCS, that can perform data-driven classification of proteins and protein segments. Training data for interesting classification problems is often limited; thus, SCS uses string transformation functions to change the encoding of proteins to reduce problem perplexity and improve classification. A wrapper-based genetic algorithm is used to search the space of possible string transformation functions to find functions that improve classification.

Categories and Subject Descriptors

I.5.1 [**Computing Methodologies**]: Pattern Recognition Models Statistical

Keywords

Bioinformatics, Classifier Systems

General Terms

Algorithms

1. MARKOV-BASED CLASSIFICATION

SCS is a probabilistic classifier, meaning that it uses a generative probability model for each possible class. For any sample protein segment, the class whose probability model assigns the highest probability to the protein becomes associated with that protein.

A Markov chain model is used to estimate the necessary probabilities, but needs training data to do so. Despite that Markov chain models have been successfuly used for problems such as speech recognition [1] and language translation [2], our initial attempts to apply Markov chain Bayesian classifier protein classification performed poorly.

Lack of training data forced us to use Markov chain models of very low order, despite using order interpolation techniques as outlined in [1] and [3]. Also, we found that our Markov models were limited regarding any phenomena that are larger Markov history window.

2. STRING TRANSFORMATIONS

SCS uses a string transformation function in front of the Markov chain Bayesian classifier, to mitigate these problems. The string transformation converts the language of proteins into a coded language that is simpler and tractable for Markov Modeling, and it also encodes some information from the entire string.

In SCS, there are two forms of string transformation: language reduction and summary features. Language reduction involves a process of transforming the 20-character language of amino acids into an alternative language that has lower perplexity. Language reduction consists of a table one-to-one and many-to-one mappings. Using English as an example we might have the following:

$\tau \rightarrow$ the	*the* becomes τ.
$a \rightarrow e$	any *e* is replaced by *a*.
$a \rightarrow i$	any *i* is replaced by *a*.
$q \rightarrow qu$	any *qu* is replaced by *q*.
$\omega \rightarrow$ t%r	*t* followed by any character followed by *r* becomes ω.

A language reduction function is a list of one-to-one and many-to-one mappings like the ones listed above. Strings are transformed by scanning. At each token of the input it the transformation list is searched for the first pattern that matches the input token(s). The mapped token is output and the current position in the input string will advances by the length of the matched tokens. If no pattern matches, then the current input token is output and the scan position advanced by one. The rules above produce the following transformations:

π(theory)	$\rightarrow$	τory
π(happy)	$\rightarrow$	happy
π(torture)	$\rightarrow$	$\omega\omega$a
π(thedogisquiet)	$\rightarrow$	τadagasqaat

SCS adds summary feature tokens to the beginning of the input string. These features can possibly measure overall aspects of the input string, including the coded entropy of the input sequence, the rate of change of composition, dominance of a certain subset of amino acids, or the modal frequency of composition changes over the sequence.

The specific parameters of each feature are as important as the selection of the features themselves. For instance, it would be very interesting to note that hydrophobicity oscillates at a certain rate over the input. To accomplish this, a Fourier-based feature is used. The parameters of the feature include the set of amino acids which are hydrophobic and the thresholds of oscillation which should be given imporance. The space of summary features and their parameters are part of the string transformation search process of the genetic algorithm (GA).

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

	True Pos.	True Neg.	G. Mean
Baseline	76.3%	94%	84.7%
Best w/o Summary Features	77.8%	94.4%	85.7%
Best w/o String Reduction	91.4%	85.6%	88.5%
Overall Best (After GA search)	91.3%	94.2%	93.6%

Table 1: Helix Experiment Summary. The baseline is the Markov classifier without GA search or any string transformations. The overall best classifier was found using GA search and both string reduction and summary feature transformations.

3. GENETIC ALGORITHM SEARCH

The space of string transformation functions is infinite. Furthermore, there is no guarantee that any particular string transformation function is useful, even if it has lower perplexity. Transforming our training data to a form which permits better Markov chain models is useless for classification, if the transformation destroys the very information on which the classification is based.

SCS uses genetic algorithm search as a wrapper, as defined in [4]. Each member of the GA population is a transformation, made up of a sequence of summary feature definitions and string reductions. Our implementation uses steady state GA with a population of 8000 members. Crossover is implemented as block-uniform crossover. The mutation rate is 25%, and the crossover rate is 75%. Selection is Rank based with exponentially decreasing probability. The GA search continues until it has reached at least 100,000 iterations and the program has iterated twice as long as required to find the last best member of the population seen so far.

4. HELIX RECOGNITION

Table 1 gives a few of the interesting data points for several different models. The first (Baseline) result is achieved by SCS with no string transformation, just the Markov model classification. Without string transformations, SCS achieved 76.3% true positive helix recognition rate and a 6.0% false positive rate (94% true negative.) The second report in Table 1 shows the overall best result of the GA search and we see a 91.3% true positive rate with 5.8% false positive rate. This is a dramatic improvement over the base performance of the algorithm.

Figure 1 shows a plot of every GA iteration and the performance of the algorithm in terms of both the ratio of accurately labeled helix segments and incorrectly labeled nonhelix segments. Clearly from this plot, we see that many of the GA population members were substantially worse than the baseline case above. The "overall best" reported above corresponds to a single point on this graph, specifically the point closest to the upper left hand corner of the graph.

Figure 1 is similar to a Receiver Operating Characteristic (ROC) graph. Each point in the plot corresponds to an individual iteration of the GA. As is standard with ROC graphs, the best possible point would be a point at the upper left hand corner.

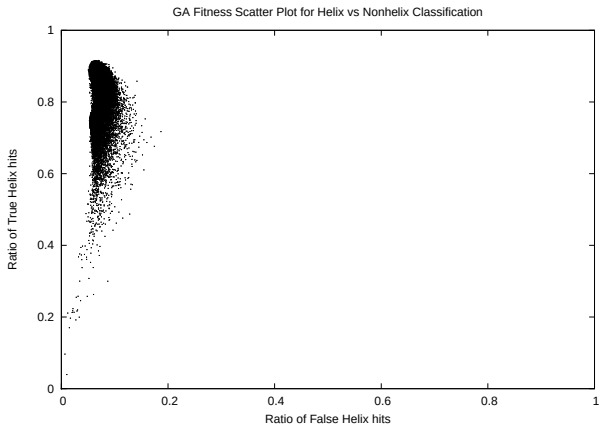

Figure 1: GA Scatter Plot for Helix Recognition (Upper left isideal.)

5. CONCLUSION

We have described SCS, a string classification system for proteins that uses Markov chain Bayesian classification. To improve classification performance, the input data for the classifier is first run through a string transformation system to reduce the perplexity of the string and to encode wide phenomena.

The particular choice of string transformation function is made by using a genetic algorithm to search the space of all string transformations to find one that improves the performance of the classifier on held-out portions of its training data.

In our experiment, we have shown helix versus nonhelix classification of protein segments. We are also looking more tasks such as disorder detection, and toxin recognition.

Acknowledgments

This research was supported by NIH Grant #P20 RR16448 from the COBRE Program of the National Center for Research Resources.

6. REFERENCES

[1] L. R. Bahl, F. Jelinek, and R. L. Mercer. Likelihood approach to continuous speech recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 2:179–190, 1983.

[2] P. F. Brown, J. Cocke, S. J. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[3] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, Computer and Information Science, Philadelphia, Pennsylvania, 1999.

[4] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13:44–49, 1998.