An SVM-based Method for Protein Structure Classification
Mark Verheggen

Abstract:
The goal of this project is to classify distantly-homologous proteins by their structure, without access to any information about the protein other than its amino acid sequence (primary structure). Protein folding is not a well-understood phenomenon. It is hard to define explicit sequence-based features for it. SVM is thus very appealing for this application because it only requires a kernel function be defined over the data points. The score of the optimal Smith-Waterman alignment between two sequences is proposed as this kernel. Smith-Waterman does not produce a correct Mercer kernel, however in practice it is close enough being positive definite to serve as one. Performance results were positive—a test error rate of about 1.5% was measured for binary classification problems, increasing to 35% on 16-classes. Attempts to transform the kernel into a positive definite one had no significant effect on performance.

Data set:
Training sequences were drawn from the SCOP (Structural Classification of Proteins) ontology (http://scop.mrc-lmb.cam.ac.uk/scop/). In contrast to most protein classification systems, SCOP has been manually compiled, with proteins categorized by known information about their structure. Its broadest categories are known as ‘folds’—classes that contain proteins of similar structures but not necessarily any other relation. These classes are adopted as the training categories. Protein sequences in the same fold often bear little resemblance to one another—making it a difficult categorization to learn. To ensure the dissimilarity of the training and test sets, all data points were drawn from a corpus of sequences with less than 40% identity to each other (obtained from ASTRAL: http://astral.berkeley.edu/). In practice though, the identity between sequences in the same class was often not greater than would be expected between two random strings.

The Kernel—Protein Sequence Alignment:
It is not clear how to take the inner product of two protein sequences. So instead, a rough similarity metric is used: Smith-Waterman alignments scores. Arguably the standard method for determining the similarity between biological sequences is the Smith-Waterman (or Needleman-Wunsch) algorithm. In brief, it is a dynamic programming for aligning two strings by introducing gaps, so as to maximize value of the matches between them. These matches are not scored uniformly but rather according to the probability of one amino acid being substituted for another via random mutation (obtained from the Blosum62 matrix). See [1] for full implementation details. A sample alignment is shown below.

```
FYANIQADATVATFFNGIDMPNQTNKTAALCGPNAWTGRNLKEVH----ANMGV
FY + D + FF +DM Q AFL A GG + + GR + + E H N G +
FYERVLQDDRICKHFFADVMAKQRAHQPFAFLTYAFAAGTDDKYGDRYMEAHKELVENHGL
```

Each alignment is associated with an overall score—corresponding to how much of the sequences matched—which is used for the kernel function for SVM.
Kernel Correctness:
Unfortunately, Smith-Waterman alignment does not produce a Mercer kernel. While it is symmetric, it is not a convex function—a fact which can be easily verified. Considered theoretically, the lack of a valid kernel has serious implications: SVM will not converge to the global maximum, nor is it guaranteed to reach a maximum at all. Practically speaking, however, it is not clear how much of an impact it will have on the performance of the classifier.

Part of the issue is that Blosum62 is not itself positive semi-definite. In theory, as a matrix of the probability of substitutions, it should be. But all of its entries are rounded to the nearest integer, introducing negative eigenvalues. Under the assumption that classifier performance would be improved if it were positive semi-definite, we deform it to force this property.

Three possible spectral transformations are considered:
- Eliminate negative eigenvalues by fixing them to zero (‘zeroed’ subsequently)
- Replace each eigenvalue with its absolute value (‘abs’)
- Subtract the most negative eigenvalue from the diagonal of the matrix (‘shift’)

Each of these transforms produces a positive semi-definite matrix, with varying degrees of loss of the original information. Because it is not apparent which approach is preferable, all three were tested.

Similarly, we can compute the kernel matrix for the entire data set, and then apply one of these transforms to it to generate a valid kernel. This approach was tested as well.

Multiple Classes:
As originally formulated, SVM performs binary classification. Two approaches for extending it to N classes are considered. In each, a classifier is trained for each pair-wise grouping of the N classes; there are 1/2*N*(N-1) total such classifiers. So for instance, given 3 classes, an SVM would be trained for classes 1-vs-2, 1-vs-3, and 2-vs-3. In the first approach, test points are presented to all of the classifiers, and given the label which receives the majority of votes.

A slightly more sophisticated approach—termed Directed Acyclic Graph SVM by its creators—is also taken [2]. For DAGSVM to classify a point, it first presents it to the 1-vs-N SVM model. If the model labels it as 1, the 1-vs-(N-1) classifier is subsequently applied to the point; if it labels it N, 2-vs-N is next used. More classifications follow. In general, the i-vs-j SVM is succeeded by the i-vs-(j-1) classifier if it returns i, or by the (i+1)-vs-j classifier if it returns j. The indexes i and j will reach equality after N-1 such classifications; that value is taken as the label of the data point. See the diagram at left for an example (from [2]).
**Results:**
To ensure roughly equal representation of data (and rapid training), fold classes were chosen randomly from those containing 25 to 50 such sequences. SVMs were trained with SMO, which in practice was so slow as to necessitate a C-based implementation. Mean 10-fold cross validation errors are reported here, with error bars denoting the standard deviation of the individual cross validation test errors. The first two graphs use DAGSVM.
Discussion:
DAGSVM clearly outperforms majority voting by a large margin. While according to [2] the two algorithms should be roughly equal, it seems that in this application classifiers will routinely give identical labels to all categories they were not trained on. So for instance, most n-v-m classifiers might label all points except those in class m as n, with the result that nearly all test points are given the labels n or m. This behavior does not adversely affect DAGSVM.

Continuing on, it would seem that the kernel being indefinite does not hinder performance; all but one of the attempted corrections had no significant impact on the results. Computing the kernel for the full 486 sequence training set (on which the worst performance was observed) yielded only 1 negative eigenvalue. So the topology of the SVM objective function is still relatively simple (it’s a saddle shape), even if it is not completely concave.

The following may also be relevant. It is provable under certain trivial conditions (met here) that any 2x2 kernel matrix (Kij = K(Xi,Xj)) of our kernel is positive definite—a fact which is confirmed by empirical observations. As SMO optimizes the objective over only a pair of coordinates at a time, it will therefore never decrease the value of the objective function.

Somewhat surprising is the failure of the diagonal shift when applied to the kernel. This particular approach for fixing indefinite kernels was explicitly recommended by [3].

Adjusting alignment parameters such as gap penalties had little effect as long as the scores were constrained to within reasonable values (e.g. not rewarding gaps). Such results are omitted here, as being too identical to those already included.

Overall, the results are positive, given the difficulty of the classification problem. Every data point was fairly unique—a fact which is evidenced by the large variance in the CV error. If nothing else, these results illustrate that string algorithms can successfully be used as kernels for SVM, despite their theoretical shortcomings.

References:
