

# Comparison of Naïve Bayes, Logistic Regression and Markov Model for protein-DNA interaction prediction

Michelle Ruse, Oksana Yakhnenko  
CS 672 Final Project

May 4, 2006

## Abstract

Protein-DNA prediction is a challenging problem solving which can help in developing numerous medicines for treatment of different diseases. In this project we compare several methods for protein-DNA interaction prediction: Naïve Bayes, Markov Model of order  $k-1$  and Logistic Regression. Although Naïve Bayes is a fairly popular method that was shown to have good results in the past, Logistic Regression shows slightly higher correlation coefficient, higher sensitivity and it is fairly fast to train. We also show that adding sequence order and interaction between the neighbors (i.e. as captured by Markov Model) does not improve classification results, and in fact performs worse than Naïve Bayes.

## 1 Introduction

Will an individual develop a certain disease given the presence or absence of certain protein-DNA interactions? What purpose do certain cells perform? Which medicines will interact the best to remedy cell damage or eradicate foreign substances within a cellular body? The bio-medical implications from the study of protein-DNA interaction are numerous.

From a given sequence of amino acids, we want to predict classes of each of these amino acids in a sequence. Such class predictions applications would include protein-protein interaction, protein-DNA interaction, or surface residue prediction, for instance. Limiting our class to a protein-DNA interaction gives a binary classifier of  $\{0,1\}$  for {"has protein-DNA interaction", "does not have protein-DNA interaction"}, respectively.

## 2 Protein-DNA Interaction

Determining protein-DNA interaction of a given molecule tells us information pertaining to the molecule interaction's interaction with other substances, pos-

sibly helping in the design of medications. Also, information applies to gene repair in a cell, as well as gene expression.

The study of the regulation of expression of gene, the regulation of replication and other structural DNA functions to maintain cellular states involve the study of protein-DNA interactions of amino acids. X-ray crystallization, gel mobility shift assay and others have physical limitations. Today's computational models could allow for faster, more cost-efficient and most-likely accurate classification of protein-DNA interaction on given amino acid sequences. Technology's current state is one that produces faster machines at lower costs enabling ever increasingly expansive computational capability.

The data set used for this work is that used by Yan, et al. From the Protein Data Bank (PDB), data was extracted from structures of known protein-DNA complexes. In order to further sample the data so that a subset of relatively high PDB structure quality and/or mutual sequence identity served as the sample, culling was performed via PISCES (Protein Sequence Culling Server). This process left 171 DNA-binding proteins sequence with identity less than or equal to 30 percent with at least 40 amino acids per sequenceCHANGUI.

### 3 Previous Work

Work in this area has provided various varying solution ideas to improve classification. For example, Jones et al. used residue patches on the surface of DNA-binding proteins, using electrostatic potentials of residues for predicting DNA-binding sites[5]. Tsuchiya et al. used structure-based method to identify protein-DNA binding sites based on electrostatic potentials and surface shape[7]. Neural Network classifier to identify patches likely to be DNA-binding sites based on physical and chemical properties of the patches. Ahmad and Sarai proposed a sequence-based method for predicting DNA-binding residues that incorporates sequence alignment profiles into the input[1].

In a paper currently in preparation by Changui Yan et al, the following results were obtained using a Naïve Bayes classifier to identify DNA-binding residues based on sequence information. The data set was trained and tested using leave-one-out cross validation, using a window size of nine with the center of the window the target amino acid. The accuracy was 71%, the correlation coefficient 0.24, specificity 53% and sensitivity 53%. Further experiments used the sequence entropy of the target residue as additional input, which is the entropy of the corresponding column in multiple alignment obtained by aligning the target sequence with its sequence homologs. This further experimentation yielded accuracy of 78%, correlation coefficient 0.28, specificity 44% and sensitivity 41% [9].

#### 3.1 Data Representation

We further manipulated the data with formatting issues for our algorithm and parsing technique. A sliding window of size nine was used, with the center

the target amino acid of the sequence. The neighboring four on both sides and the target itself were used for training to predict if the target amino acid had a protein-DNA surface interaction. Issues arise with this technique for the beginning and the ending of each sequence, when the target acids are in positions [1,4] or [n-4,n], where n is the length of our given sequence. Not all eight values were present for use by the learner. In these instances, we used unknown acids and interactions.

### 3.2 Naïve Bayes

A well known Machine Learning technique Naïve Bayes has been widely used in much computational research. Its simplicity along with its performance in prediction have kept in at the forefront and often used as a starting point for solving computational problems that extend beyond feasible experimentation.

Naïve Bayes assumes no dependencies between variables in predicting the class of an object, in other words there is class conditional independence. Derived from Bayes' Theorem of probability:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (1)$$

This equation is the calculation of the probability that the hypothesis H of X belonging to a class, given that X is the observed data. P(H|X) is the posterior probability of H conditioned on X. P(X) is the prior probability of X, P(H) is the apriori probability of H, that X belongs to class C. P(X|H) is the posterior probability of X conditioned on H.

Using Naïve Bayes for classification combines the above probabilistic model with a decision rule based on the training data. Commonly, selecting the most probable class serves as the rule:

$$classify(f_1, f_1, \dots) = \arg \max_{c_j \in C} P(C = c_j) \prod_{i=1}^n P(F_i = f_i | C = c_j) \quad (2)$$

This gives the class the maximum probability in a given instance as the rule for the observed data as the decision in the predicting phase, after learning has created the various decision rules.

The Naïve Bayes classifier predicts if a target residue in a protein sequence to identify DNA-binding residues using sequence information alone. The window selects a center target and uses the neighboring information to train and predict the binding residues in this local algorithm. Naïve Bayes is used in determining target and neighboring relationships and determining the rules for classification.

## 4 Methods

In our attempts to improve on the existing methods we have applied two methods that have been shown to improve upon Naïve Bayes. The first method,

Markov Model of order  $k-1$  (or NB $k$ ), takes into account the order of the elements in the sequence, as well as models dependencies between  $k$  neighboring elements. The second method, Logistic Regression, is a discriminative counterpart of Naïve Bayes in the parameters are chosen in order to maximize the class conditional log likelihood of the data. With sufficient data the second method of Logistic Regression was shown to improve the accuracy when compared to Naïve Bayes, generative case. In this section we describe basic ideas behind the Markov Model and Logistic Regression.

#### 4.1 Markov Model of order $k-1$ (NB $k$ )

The following section comes from [8]

"Markov Models for sequence classification have been used with success by many researchers in a variety of applications [3] [10] [2]. We start with a brief review of the basic ideas behind MM( $k-1$ ).

Let  $S$  be a sequence,  $s_i$  be the value of an element of  $S$  at the position  $i$ , and  $\Sigma$  be the alphabet over which the sequence values range.

A sequence can be modeled as a graph in which each sequence element is represented by a node, and a direct dependency between two neighboring elements is represented by an edge in the graph. Generally, it is the case that two or more neighboring elements in the sequence will be dependent on each other. Figure 1 shows several directed dependency models for: a) Naïve Bayes which corresponds to assuming no dependence between neighboring elements of a sequence, b) a dependency model of the first order i.e., one that considers dependencies between pairs of neighboring elements of a sequence ( $k=2$ ) and c) a dependency model of the second order i.e., one that considers dependencies among three adjacent elements of a sequence. More generally, Markov Models of order  $k-1$ , capture the dependency between the current element  $s_k$  and its  $k-1$  preceding elements  $[s_{k-1} \dots s_1]$  in a sequence. MM( $k-1$ ) family of generative models offer the needed expressive power to model increasingly complex dependencies among neighboring elements of a sequence.

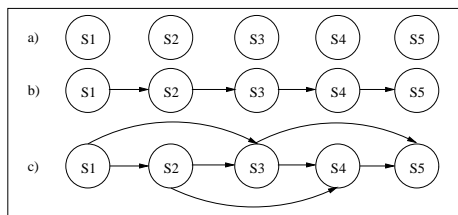


Figure 1: Markov Model representation of dependencies of order  $k-1$  dependency. a) Naïve Bayes model; b) Markov Model of order 1 c) Markov Model of order 2

The joint probability distribution for the MM( $k-1$ ) follows directly from

the Junction Tree Theorem [4] and the definition of conditional probability:

$$\begin{aligned} P(S = s_1 s_2 \dots s_n, c_j) &= \frac{\prod_{i=1}^n P(S = s_i \dots s_{i+k-1}, c_j)}{\prod_{i=1}^n P(S = s_i \dots s_{i+k-2}, c_j)} \\ &= P(S = s_1 \dots s_{k-1}, c_j) \\ &\quad \prod_{i=k}^n P(S = s_i | s_{i-1} \dots s_{i-k+1}, c_j) \end{aligned}$$

The probabilities  $P(S = s_i | s_{i-1} \dots s_{i-k+1}, c_j)$  can be readily estimated from data using the counts of the subsequences  $s_i \dots s_{i-k+1}, c_j$  and  $s_{i-1} \dots s_{i-k+1}, c_j$  as sufficient statistics. With the generative model in place, a sequence  $S$  to be classified is assigned to the most likely class based on the generative models for each class. That is,  $class(S = s_1 \dots s_n) = \arg \max_{c_j \in C} P(S = s_1 \dots s_{k-1}, c_j) \prod_{i=k}^n P(S = s_i | s_{i-1} \dots s_{i-k+1}, c_j)$ . Markov Models of order  $k - 1$  (where  $k > 1$ ) have been shown to have consistent improvement in accuracy over Naïve Bayes (which is equivalent to a Markov Model of order 0), with the classification accuracy typically increasing with the increase in  $k$  (until we run out of data to reliably estimate the increased number of parameters) [2]. "

## 4.2 Logistic Regression

Logistic regression is a generative counterpart of Naïve Bayes. It assumes that all features are independent given class label, however the parameters are chosen to maximize the probability of class given the sequence  $P(C = c_j | S)$ . The model assumes parametric form for the distribution, and then estimates the parameters directly from data. If  $Y = \{0, 1\}$  is the class, and  $X = [x_1 \dots x_n]$  is the observation, then the class conditional likelihood

$$P(Y = 1 | X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_i x_i)}$$

and

$$P(Y = 0 | X) = \frac{\exp(w_0 + \sum_{i=1}^n w_i x_i)}{1 + \exp(w_0 + \sum_{i=1}^n w_i x_i)}$$

where  $w_i$  is a weight associated with a feature  $x_i$  of the observation. The derivative of the likelihood has the form

$$\frac{\partial P}{\partial w_i} = \sum_l x_i^l (Y^l - P(Y^l | X^l, w))$$

and the parameters  $w_i$  are learnt using gradient descent:

$$w_i^{t+1} \leftarrow w_i^t + \eta \frac{\partial P}{\partial w_i}$$

where  $0 < \eta \leq 1$  is the learning rate.

With sufficient data, Logistic Regression was shown to have improved accuracy over Naïve Bayes [6].

## 5 Results

We have applied Logistic Regression (implemented in Weka) using the same representation as described in section 3.1, and NBk (airIDM implementation) to the dataset, using Naïve Bayes as a baseline for comparison. We have used ROC curves, accuracy, correlation coefficient to evaluate the results of these algorithms. We have used 5-fold cross-validation and we did not adjust the trade-off parameter for the final results.

### 5.1 Performance Measures

We have used accuracy, sensitivity, correlation coefficient and ROC curves to measure the performance. We define these measures as  $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ ,  $sensitivity = \frac{TP}{TP+FN}$  and  $CC = \frac{TPTN-FPFN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$ . ROC curves capture the trade-off between true positive rate and false positive rate as we vary the threshold  $\theta$  when the decision rule is  $class(s) = c_i$  if  $\frac{P(s,c_i)}{P(s,c_j)} > \theta$  where  $P(s, c_i)$  is a score assigned by a classifier if the assigned class is  $c_i$ .

### 5.2 NBk

To our surprise, adding dependencies in the model did not help and did not show improvement. Figure 2 shows the ROC curves obtained for NBk trained for  $k=1$  (Naïve Bayes) plotted in navy,  $k=2$  plotted in red,  $k=3$  plotted in light blue, and  $k=4$  plotted in green. The model becomes worse as the number of the dependencies increase. There are two possible explanations for this. One is that the data is too sparse, and with the increasing value of  $k$  the number of parameters needed to be estimated increases. We may not have had enough data to accurately estimate these parameters. Another possible explanation is that with the representation of the target residue with four neighbors on the left and on the right the sequence, we do not have enough information. We possibly need to use a larger window size in this representation to obtain some improvement.

### 5.3 Logistic Regression

The performance of the logistic regression was similar to that of Naïve Bayes. Figure 3 shows ROC curves for Naïve Bayes and Logistic Regression. The logistic regression showed a slightly higher ROC curve and it was able to identify more residues correctly than Naïve Bayes.

When comparing correlation coefficient, accuracy and sensitivity of Logistic Regression and Naïve Bayes we have not adjusted the  $\theta$  parameter to trade-off between the true positive and false positive rate. Table 1 shows these results for Naïve Bayes and Logistic Regression. Both algorithms have the exact same accuracy, however Logistic Regression shows a slightly higher correlation coefficient and was able to predict 20 more residues involving in protein-DNA

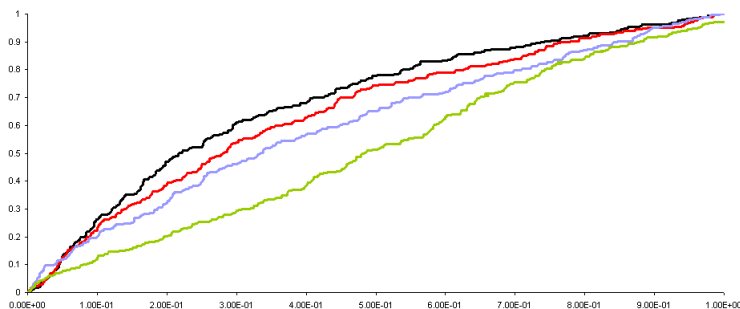


Figure 2: ROC curves for NBk. Navy k=1, red k=2, blue k=3, green k=4. The performance of the model decreases as the number of the dependencies increases.

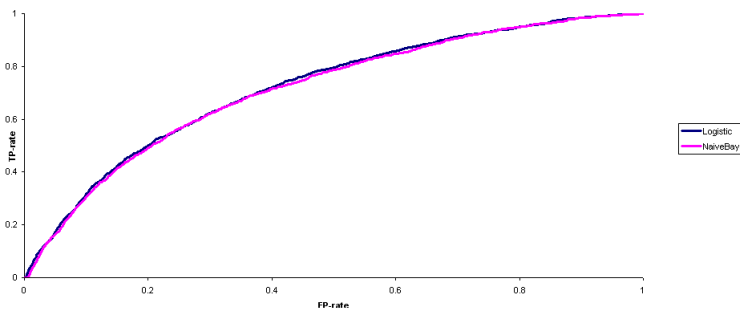


Figure 3: ROC curves for Naïve Bayes and Logistic Regression

interaction than Naïve Bayes, which is very important due to the unbalanced nature of the dataset.

## 6 Conclusion and Future Work

In this project we have attempted to improve on protein-DNA interaction prediction model from sequence. The base model uses four elements to the right and the left of the target residue and Naïve Bayes to predict whether a residue is an interaction residue or not. We have applied Markov Model of order  $k - 1$  that shows no improvement, and in fact worsens as the number of interaction increases, and Logistic Regression that have shown some improvement in the performance.

We would like to experment with larger window size in the representation, which may show some improvement in  $MM(k - 1)$ . We would also like to directly compare Logistic Regression and Naïve Bayes by using leave-one-out

Algorithm	Accuracy	CC	Sensitivity
<b>Naïve Bayes</b>	0.86	0.11	0.51
<b>Logistic Regression</b>	0.85	0.13	0.63

Table 1: Accuracy, correlation coefficient and sensitivity of Naïve Bayes and Logistic Regression after 5-fold cross-validation

cross validation, tuning the  $\theta$  parameter on the ROC curve to trade-off between true positive rate and false positive rate, and include additional information in the input, such as entropy, or solvent accessibility area.

It would also be of interest to develop a graphical model that would include the interaction directly throughout the sequence as was in the initial idea (that did not happen due to one of the author’s computer failure).

## References

- [1] S. Ahmad, M. Gromiha, and A. Sarai. Analysis and prediction of dna-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics*, 2004.
- [2] C. Andorf, A. Silvescu, D. Dobbs, and V. Honavar. Learning classifiers for assigning protein sequences to gene ontology (GO) functional families. In *Proceedings of the Fifth International Conference On Knowledge Based Computer Systems (KBCS)*, 2004.
- [3] E. Charniak. *Statistical Language Learning (Language, Speech, and Communication)*. MIT Press, 1996.
- [4] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. Probabilistic networks and expert systems. *Springer*, 1999.
- [5] S. JoneS, H. Shanahan, H. Berman, and J. Thornton. Using electrostatic potentials to predict dna-binding sites on dna-binding proteins. *Bioinformatics*, 2003.
- [6] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naïve bayes. 2002.
- [7] Y. Tsuchiya, K. Kinoshita, and H. Nakamura. Structure-based prediction of dna-binding sites on proteins using the empirical preference of electrostatic potential and the shape of molecular surfaces. *Proteins*, 2004.
- [8] O. Yakhnenko, A. Silvescu, and V. Honavar. Discriminatively trained markov model for sequence classification. 2005.

- [9] C. Yan, M. Terribilini, F. Wu, R. L. Jernigan, D. Dobbs, and V. Honavar. Predicting dna-binding sites of proteins from amino acid sequence. *In preparation*, 2006.
- [10] Z. Yuan. Prediction of protein subcellular locations using Markov chain models. *FEBS Letters*, 451(1):23–6, 1999.