Algorithms in Bioinformatics

Mini project no. 01 2010
Pages 1–4

Stabilization matrix method - Mini project

Sigmar Stefànsson and Francesco Favero

Danmarks Tekniske Univeristet

22 October 2010

Supervisor: Morten Nielsen

ABSTRACT

Summary We look at Stabilization matrix methods using two different kinds of minimizations procedures [3] for predicting binding [1] affinity of immunogenic peptides to major histocompatibility complex [5] (MHC) molecules.

The testing data available was from 35 different MHC molecules from HLA-A and HLA-B, with number of peptides ranging from 59 to 3089.

Finally we studied the results of the Monte Carlo and the gradient decent implementations of the SMM, to find the optimal setting of the algorithms.

INTRODUCTION

In simple mathematical terms the SMM algorithm tries to minimize the difference between the predicted values and the measured values according to a norm.

$$||Hw - y_{meas}|| +^t w\Lambda w \rightarrow minimum \qquad (1)$$

In this case the L2 norm is used. The second term suppresses the effect of noise in the experimental data [3]. The regulation parameter λ is used to deal with over-fitting in the SMM algorithms [6]. In the context of the SMM algorithm, H is a matrix encoding the peptides, w is the prediction (position specific scoring) matrix and y_{meas} is vector containing the measured binding affinity of the peptides.

Over-fitting occurs when a statistical model describes random error or noise instead of the underlying relationship [2]. A model having too many degrees of freedom, in relation to the amount of data available is susceptible to over-fitting. A model which has been overfit will generally have poor predictive performance. Among common methods of avoiding over-fitting are early stopping and cross-validation.

In early stopping, the training is stopped based on test set performance (when it starts decaying). Cross-validation is about combining models. In K-fold cross-validation, the original sample is randomly partitioned into K sub samples. Of the K sub samples, a single sub sample is retained as the validation data for testing the model, and the remaining K 1 sub samples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K sub samples used exactly once as the validation data [4].

In terms of the SMM algorithm, the solution is not found by simple matrix inversion using 1. The system of linear equations is under- or overdetermined and an approximate solution is to be located on the solution hyper-surface that can contain local minima. Equation 2 shows how the error used in the minimization procedure is calculated. O and t being the prediction output and the measurement respectively.

$$E = \frac{1}{2}\sum_{i}(O_i - t_i)^2 + \lambda\sum_{l} w_i^2 \qquad (2)$$

MATERIAL AND METHODS

We were given semi-completed code for the implementation of the Stabilization matrix method. The minimization procedures were Gradient descent in one case and Monte Carlo in the other.

The testing data available was from 35 different MHC molecules from HLA-A and HLA-B, with number of peptides ranging from 59 to 3089. In addition to finishing the code we added some optimizations like skipping unnecessary memory allocations at performance critical locations.

We ran the matrix creation on each 35 MHC molecules to get the broadest perspective. The code did run fast enough for us to be able to search for optimal λ value with brute force. The λ values we tested were 0, 0.002, 0.005, 0.01, 0.02, 0.03, 0.05 and 0.1. For largest data sets the optimal lambda values were the ones close to zero, 0.002 and 0.005.

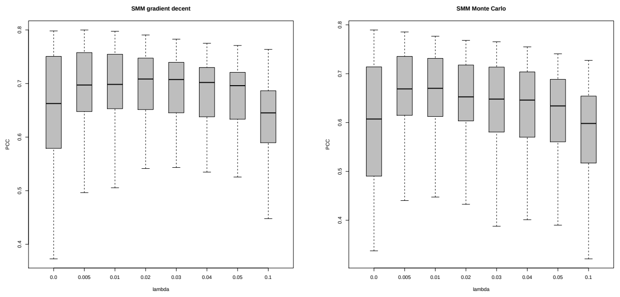
Regardless the size of the data, for the two implementations the optimal value of λ is different. In Figure 1 all the 35 Pearson correlation coefficient are include. As we can see the Monte Carlo is better with smaller value of λ (around 0.005 and 0.01 see Figure 1(a)), the gradient decent version have the optimal λ value slightly bigger (best λ around 0.02) see Figure 1(b).

The data for each MHC molecule was splitted into five parts, equally (or nearly equally) sized. Each part was then used for the validation by the K-fold cross validation method [4]. So each part was trained on the rest (4/5) of the data, resulting in 5 iterations to be used for the evaluation.

In order to find the optimal λ value we evaluate the correlation of all the 35 molecules with both algorithms with different λ parameter. To reduce the running time of the procedure we splitted the list of the 35 dataset in 7 different jobs taking care of 5 dataset each.

As noted in Figure 2 and in Figure 3 the use of the λ value suppresses the effect of noise in the measured data. In the case of zero value λ, the optimal entries for the weight vector *w* minimize the difference between predicted and measured values. Minimizing with a non-zero value for λ results in a shift of the optimal entries in *w* towards values closer to zero. It should be noted here that the λ

© Danmarks Tekniske Univeristet 2010.



(a) Gradient decent

(b) Monte Carlo

Fig. 1. Plot of Pearson correlation coefficients at progressive increasing of λ values, for the gradient decent 1(a) and for the Monte Carlo 1(b) implementation

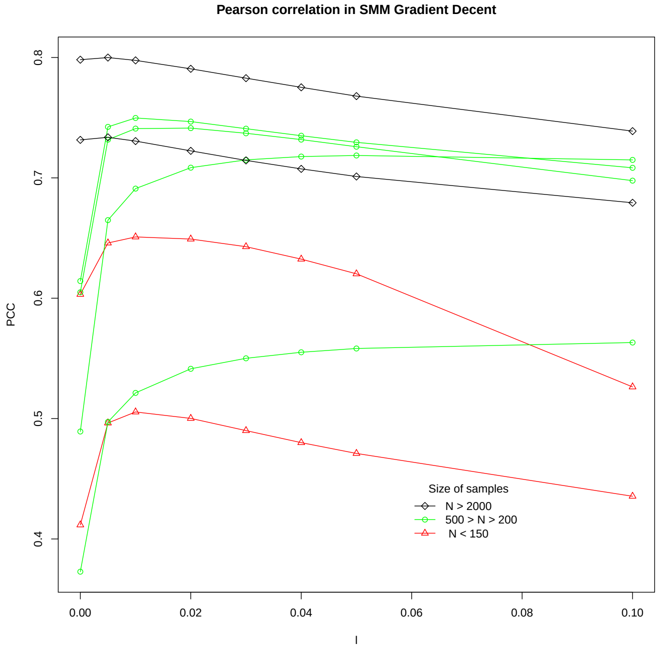


Fig. 2. Pearson's correlation coefficient from the SMM Gradient Decent,of different sample using increasing values of λ. The sample were grouped by the size of the dataset. We can see how dataset of similar size show a similar trend of the Pearson correlation λ depending.

value used as a parameter in the code is the per-target normalized λ value, not the global one.

We also evaluate each single group of peptide binding data. From the bar-plot in Figure 7 we can see how the 5-fold concatenate evaluation it is not better than all the single fold evaluation, but it is seems more an average of all them.

RESULTS

Figure 2 shows the trend for Pearson's correlation coefficient (PCC, prediction accuracy) using different λ values for various data sizes.

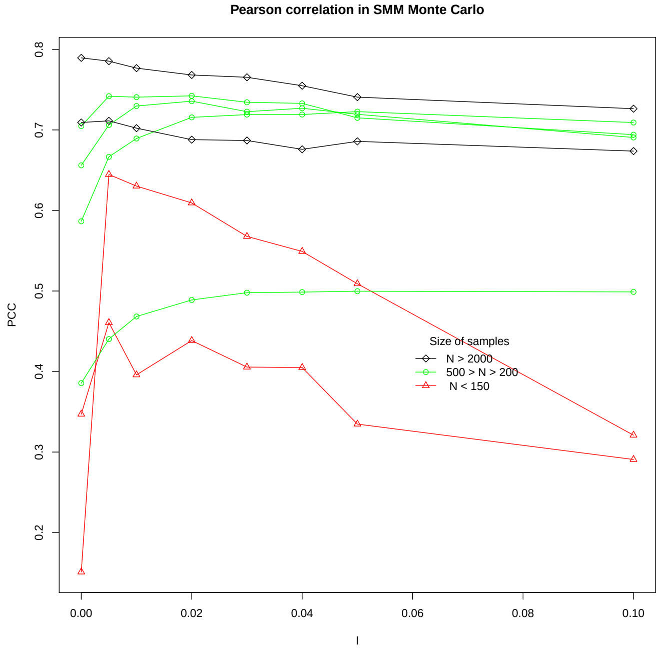


Fig. 3. Pearson's correlation coefficient from the SMM Monte Carlo, of different sample using increasing values of λ. The sample were grouped by the size of the dataset. We can see how dataset of similar size show a similar trend of the Pearson correlation λ depending.

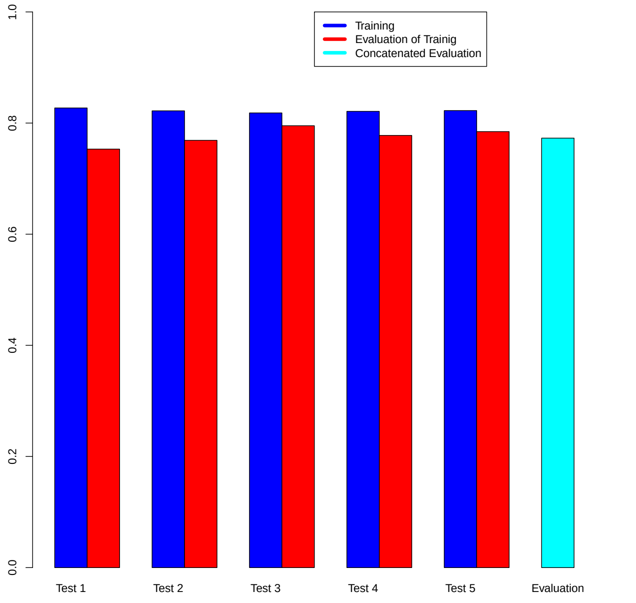


Fig. 4. Bar-plot showing the Pearson Correlation for each training set, for each evaluation on the test set and the correlation of the concatenate evaluation among the 5 test set

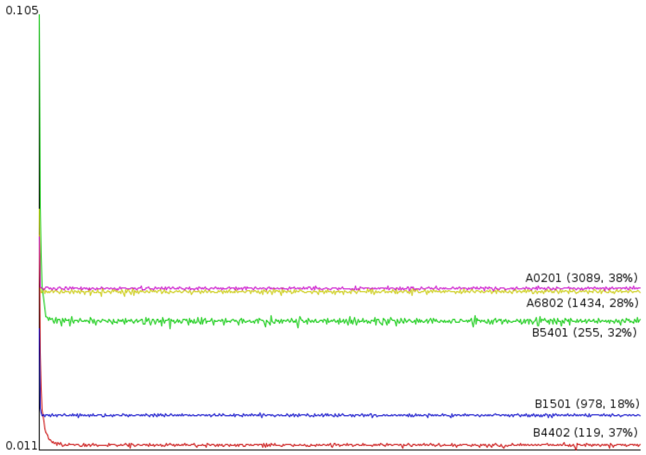


Fig. 5. Error trend for dataset of different size in function of the number of cycle of the SMM gradient decent algorithm

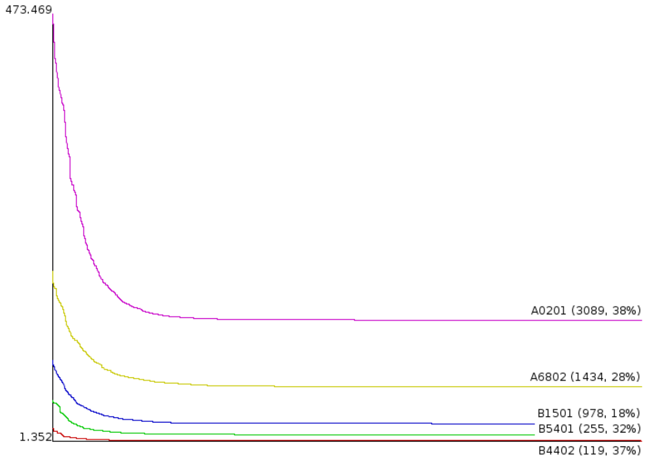


Fig. 6. Error trend for dataset of different size in function of the number of cycle of the SMM Monte Carlo algorithm

As a general rule, as the data size gets larger the less important the effect of a non-zero λ becomes. For large data sets a λ value of zero seems to give the best results. A large dataset should average out the noise. Similar graph showing results from the Monte Carlo version says the same story (Figure 3).

For the small datasets an optimal λ value is less than 0.01. Next we run the SMM algorithms on a data from five different kinds of MHC molecules (with different data sizes) showing the error as a function of iterations. For the SMM algorithm using gradient descent and 500 iterations, a default lambda value of 0.05 is used (Figure 5). Similar chart is shown for the Monte Carlo version of the algorithm in Figure 6 showing 12000 iterations. The reason for showing different data sizes on the same chart is just for showing the trends in the two different algorithms. The gradient descent method converges much faster to an optimal value than the Monte Carlo method. The data sizes and percentage of number of binding peptides are shown in the parentheses right by the names of the lines representing each MHC molecule.

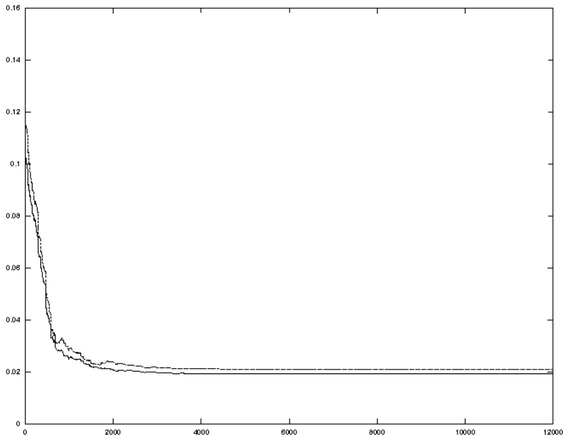


Fig. 7. When training the Monte Carlo version of the algorithm with a λ of 0.05, the mean square error for the testing seems to follow the training error pretty well. In this case an early stopping to prevent overfitting is unnecessary

The gradient descent version of the SMM algorithm is both performing faster and better than the Monte Carlo version. The speed of the convergence is determined by the derived slope in gradient descent. The Monte Carlo version lowers the temperature not in the context of a slope.

DISCUSSION

We generally can assume that the gradient decent implementation performs better than the Monte Carlo SMM implementation.

From Figure 8 we can easily see that even if performed with the optimal λ value we have lower Pearson correlation coefficients values in the Monte Carlo implementation than in the gradient decent.

We made some adjustments to the c code to be able to compare the mean square error (MSE) of the training and testing results in each cycle. The gradient descent version converged in most cases in just one or two iterations. In Monte Carlo the testing MSE did not seem to decay with continuing iterations, indicating that preventing overfitting with early stopping would not be practical in that case Figure 7.

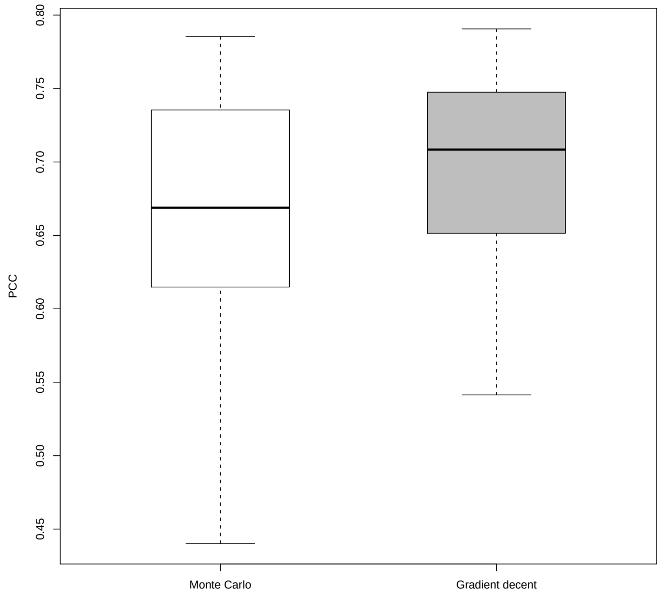


Fig. 8. Boxplot of the evaluate Pearson correlations obtained with SMM Monte Carlo and Gradient decent at the respective best λ values for all the 35 datasets.

REFERENCES

[1]C. Lundegaard, K. Lamberth, M. Harndahl, S. Buus, O. Lund, and M. Nielsen. NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8-11. *Nucleic Acids Res*, 36(Web Server issue):W509–12, 2008.

[2]N. Morten. Dealing with sequence redundancy. *Class lecture*, 2010.

[3]B. Peters and A. Sette. Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. *BMC Bioinformatics*, 6(132), 2005.

[4]Wikipedia. Cross-validation (statistics), 2010. [Online; accessed 28-September-2010].

[5]Wikipedia. Major histocompatibility complex, 2010. [Online; accessed 18-October-2010].

[6]Wikipedia. Overfitting, 2010. [Online; accessed 15-October-2010].