A Wrapper Feature Selection Method Based on Simulated Annealing Algorithm for Prostate Protein Mass Spectrometry Data

Yifeng Li, Yihui Liu

Abstract—**Protein mass spectrometry is an integration of mass spectrometry and biological chip techniques, and it shows great potential for exploration of biomarkers and diagnosis of diseases. But the curse of dimensionality inherently from mass spectrometry data makes the dimensionality reduction a necessary phase of proteomic pattern recognition before classification. This paper presents a simulated annealing algorithm to select discriminant feature subsets. Experiments indicate that this wrapper feature selection method performs well and outperforms the other reported methods.**

I. Introduction

High-throughput mass spectrometry is a significant tool for researching proteomics, and it has shown great potential for biomarker identification and detection of early-stage cancer. The changing of condition of organism can be directly reflected by proteome, so mass spectra are generated from the circulating proteome (e.g. serum, plasma, and nipple fluid). Nevertheless the immense amount of dimensionality of the spectra containing redundancy and information irrelevant to a particular disease poses considerable challenges to existing classification algorithms. Hence feature selection is a crucial phase before conducting classification. The process of disease diagnosis using proteomic patterns [1] is that: 1) Proteomic patterns are produced by mass spectrometer coupled with protein chips which sample from body fluid; 2)pattern recognition, including preprocessing, feature selection, and classification, is conducted on these mass spectra; 3) diseases are diagnosed by using identified biomarkers or built pattern classifier.

Feature selection classically aims to select a subset of m features from a set of n features, such that the value of criterion function is optimized over all subset of size m [2], and the between-class distance is enlarged while the within-class variance is narrowed in feature vector subspace. There are four basic steps in a typical feature selection approach [3]: a generation procedure generates the next candidate feature subset for evaluation; an evaluation function evaluates the subset under examination produced by the generation procedure and holds the best hitherto; a stopping criterion decides when to end; and a validation procedure checks whether the subset is valid by conducting classification and comparison, this phase is called validation classification in this study. Different feature selection methods can be grouped into two broad groups (i.e., filter and wrapper) based on their dependence on the validation classification algorithm that will finally use the selected subset. Filter methods are independent of the validation classification algorithm, whereas wrapper methods use the validation classification algorithm as the evaluation function [4].

There have been many feature selection and feature extraction methods to reduce dimensionality for mass spectra. For filter methods, there are Student-t test (T-test) [5], Kolmogorov-Smirnov test (KS-test) [5], [6], P-test [5], Receiver Operating Characteristics (ROC) [7], and more sophisticated wavelets techniques [8]. For wrapper method, there are sequential forward selection (SFS) [5], sequential backward selection (SBS) [5], and boosting [5], [9] approaches. For feature extraction methods, there are principal components analysis (PCA) [5], [10] and principal components analysis coupled with linear discriminant analysis (PCA/LDA) [5]. For intelligent optimization method, genetic algorithm (GA) is also employed in [11].

For prostate cancer studies, in [7], Adam *et al.* used ROC to select features and decision tree to classify, and yielded a sensitivity of 81%, a specificity of 97%, and a balanced accuracy (BACC) of 89% on a single test set of PC-IMAC-Cu dataset. In [9], Qu *et al.* utilized decision stumps together with AdaBoost (BDS) and Boosted Decision Stump Feature Selection (BDSFS) to select features. The BDS achieved an average sensitivity of 98.5%, an average specificity of 97.9%, and an overall BACC of 98%, whereas the BDSFS obtained a sensitivity of 91.1% and a specificity of 94.3% using a randomized 10-fold cross-validation on PC-IMAC-Cu dataset. In [12], Michael *et al.* employed a filter-based ANOVA F-statistic to rank features, and employed k-nearest-neighbors (kNN), linear/quadratic discriminant analysis (LDA/QDA), and support vector machines (SVM) to classify PC-IMAC-Cu dataset by using 100-fold randomized cross-validation strategy. The best accuracy of 91% is achieved by linear SVM. In [10], Lilien *et al.* used PCA for dimensionality reduction and LDA for classification on PC-IMAC-Cu dataset, when training sets were larger than 75% of the total sample size, an average accuracy of 88% was obtained, but it dropped to 86% when only 50% of the dataset was used for training. Petricoin *et al.* [13] and Wulfkuhle *et al.* [14] used GA's for feature selection and performed self organizing Maps (SOM's) for classification of PC-H4 dataset. They achieved a specificity of 95%, a sensitivity of 71%, and an average accuracy of 83%. In [5], Levner performed PCA, PCA/LDA, SFS, SBS, P-test, T-test, KS-test, nearest

Manuscript received March 31, 2008.

Yifeng Li is with the Institute of Intelligence Information Processing, School of Information Science and Technology, Shandong Institute of Light Industry, Jinan, Shandong 250353 China (corresponding author; phone: 86-531-89631256; fax: 86-531-89631251; e-mail: bolirenyifeng@yahoo.com.cn).

Yihui Liu is with the Institute of Intelligence Information Processing, School of Information Science and Technology, Shandong Institute of Light Industry, Jinan, Shandong 250353 China (e-mail: yxl@sdili.edu.cn).

shrunken centroid (NSC), and boosted feature extraction (BoostedFE) for feature selection or extraction and nearest centroid classifier for classification. A wrapper-based BoostedFE method achieved the best performance, an accuracy of 96%.

II. THEORIES

A. Simulated Annealing Algorithm

Simulated Annealing Algorithm (SAA) is a random global optimization methodology regardless of the differentiability and the multimodality of the objective function. It mimics the annealing process of physical system in statistical mechanics. Its basic concepts were presented by Metropolis *et al.* in [15] in 1953, but it was not until three decades later that this theory was advertised by Kirpatrick *et al.* who published an appealing paper on "Science", namely "Optimization by simulated annealing" [16] in which SAA was employed to solve large combinatorial optimization problems. From then on, many variants of SAA have been presented and applied to fairly large amounts of mathematical and engineering problems in many different domains.

The physical process is that: the substance is heated to molten state and subsequently it cools sufficiently slowly to ensure to reach thermal equilibrium at each temperature, finally, when the temperature is equal to zero, it gets to crystalline lattices of minimal energy (i.e. ground state). If the highest temperature is less than melting point or it anneals too quickly, it will solidify into a sub-optimal configuration which does not contain minimal energy. Metropolis *et al.* [15] modeled the evolution of a substance at a constant temperature based on Monte Carlo techniques. Given the current state i of the substance with energy E_i, the subsequent state j with energy E_j is generated by a small random perturbation to state i. If E_j is less than or equal to E_i, the state j can be definitely accepted as current state. Otherwise, the state j is accepted with a probability given by

$$\exp\left(\left(E_i - E_j\right)/\left(k_B T\right)\right) \tag{1}$$

where T is the current temperature and k_B is Boltzmann constant. The acceptance rule described above is called Metropolis criterion, and the algorithm that goes with it is known as the Metropolis algorithm. As temperature is decreasing sufficiently slowly, the substance reaches thermal equilibrium at each temperature. The thermal equilibrium can be characterized by Boltzmann distribution which gives the probability of the substance being in state i with energy E_i at temperature T. That is

$$P_T\left(X = i\right) = \exp\left(-E_i/\left(k_B T\right)\right) \Big/ \sum_j \exp\left(-E_j/\left(k_B T\right)\right) \tag{2}$$

where the denominator is the sum of energy of all possible states at temperature T.

SAA draws an analogy between the cooling of a material to search for minimal energy state and the solving of an optimization problem. The solutions in an optimization problem are equivalent to the states of the substance. The value of objective function for a feasible solution is equivalent to the energy of a state. The minimum of the objective corresponds to the energy of the ground state. SAA sets a control parameter to play the role of temperature. Essentially, SAA executes the Metropolis algorithm iteratively as decreasing the control parameter which is analogous to the temperature in the Metropolis algorithm. SAA can be modeled mathematically by means of Markov chains [17], in which the process of simulated annealing is viewed as a serial of Markov chains, each of which corresponds to a temperature status. For an instance (S, f) of minimization problem and a neighborhood function, f is the objective function and S is the solution space, the pseudo-code using the standard SAA is described as the following:

Initialize (x_0, t_0, L_0); $k = 0$; $x_i = x_0$;
while (*stopcriteria*==*false*)
  {*for* (l=1; l<=L_k; l++)
    {*Generate* (x_j *from* S_i);
     if ($f(x_j) <= f(x_i)$) $x_i = x_j$;
     else if (exp($(f(x_i)-f(x_j))/t_k$)>*random*[0,1)) $x_i=x_j$;}
   k++;
   CalculateMarkovLen (t_k);
   CalculateControl (t_k);}

where x_k, t_k, and L_k denote the values of solution x, control parameter t, and Markov chain length L respectively in iteration k of the algorithm, function *Initialize* computes the initial values of the solution (generally generated randomly), the control parameter, and the length of Markov chain, function *Generate* choices a new solution x_j from the neighborhood of the current solution x_i, *CalculateMarkovLen* and *CalculateControl* update the Markov chain length and the control parameter respectively, and *stopcriteria* denotes the termination criteria of the algorithm, such as reaching a very low temperature, going beyond the iteration limit, the objective value in the last trial of a Markov chain remaining unchanged over a number of consecutive chains, and so on.

In a nutshell, the most significant and special nature of the algorithm is that it not only accepts all improvement solutions but also the deterioration solutions with a certain probability which is lowered along with the decreasing of the control parameter. By accepting deteriorations, the algorithm avoids being trapped in local minima, and is able to explore globally for more possible solutions. Given an instance of optimization problem and an appropriate neighborhood function, SAA can find global optimal solutions with probability 1 if it is allowed an infinite number of transitions [17], [18]. High-quality solution can be obtained in a finite-time implementation of it.

B. Linear Discriminant Analysis

In this study, the Linear Discriminant Analysis (LDA) classifier is designed to construct the evaluation function of feature subsets generated and valid the performance of the feature selection method. LDA, which stems from R. A. Fisher's classical and pioneering paper [19], aims to obtain an optimal projection direction (also called decision or discriminant hyperplane) so that the distances of samples from different classes are enlarged, while the distances of each sample in the same class are shortened to the best. Under this projection direction, the m-dimensional feature vectors of measurement samples are projected onto lower dimensional

subspace (generally one-dimensional space). In the purpose to separate data well, LDA considers maximizing the following objective:

$$J(w) = \left(w^T S_B w\right) / \left(w^T S_W w\right) \tag{3}$$

where w denotes the projection direction represented as a column vector, w^T is the transpose of w, S_B and S_W stand for the between classes scatter matrix and the within classes scatter matrix, respectively. They are:

$$S_B = \sum_{i=1}^{c} \left[P(c_i)(\mu_i - \mu)(\mu_i - \mu)^T\right] \tag{4}$$

$$S_W = \sum_{i=1}^{c} \left\{P(c_i) E\left[(x - \mu_i)(x - \mu_i)^T \mid c_i\right]\right\} \tag{5}$$

where $P(c_i)$ is the prior probability of class c_i, c is the number of classes, μ_i is the mean of class c_i, μ is the mean of all the data. For binary-class problem, the projection direction is

$$w^* = \arg\max_w [J(w)] = S_W^{-1}(\mu_1 - \mu_2). \tag{6}$$

It can be used as the weight vector of linear discriminant function

$$g(x) = \left(w^*\right)^T x + w_0 \tag{7}$$

where w_0 is known as threshold.

Assume that each class follows a multivariate normal distribution, the posterior probability of each sample in subspace can be calculated as

$$P(c_i \mid x) = P(x \mid c_i) P(c_i) \Big/ \sum_{j=1}^{c} P(x \mid c_j) \tag{8}$$

where prior probability $P(c_i)$ is difficult to obtain in fact. $P(c_i)$ can be approximated by n_i/N for random sampling and large sample sets, or it can be assigned to equal probabilities with other classes for small sample sets.

C. k-fold Cross-validation

In this study, training sets and testing sets are generated by k-fold cross-validation during the process of validation classification. In k-fold cross-validation, the samples in the initial dataset is distributed to k mutually exclusive and approximately equal-sized subsets or "folds" at random: D_1, D_2,..., D_k. Training and testing is conducted k times. In the *ith* time, subset D_i is reserved as the testing set, and the rest *k-1* subsets are collected as training set to train the classifier.

D. Evaluation Measurements of Classification Performance

In this study we use the following measurements to evaluate the classification performance on the dataset after feature selection:

Accuracy=(*TP*+*TN*)/(*TP*+*TN*+*FP*+*FN*)
Sensitivity=*TP*/(*TP*+*FN*)
Specificity=*TN*/(*TN*+*FP*) (9)
BACC=(*Sensitivity*+*Specificity*)/2
PPV=*TP*/(*TP*+*FP*)
NPV=*TN*/(*TN*+*FN*)

where *TP*, *TN*, *FP*, and *FN* stand for the numbers of true positive (cancer), true negative (normal), false positive, and false negative samples, respectively. BACC, PPV, and NPV are shorts for balanced accuracy, positive predictive value, and negative predictive value, respectively.

III. MATHEMATICAL MODELING

A. Problem of Prostate Protein Mass Spectra Feature Selection and Classification

In the case of feature selection and classification for prostate protein mass spectrometry data, samples generated from surface enhanced laser desorption/ionization time off light (SELDI-TOF) mass spectrometer are labeled into two classes, namely cancer class and normal (or healthy) class. Given cancer class is composed of n_1 samples, and normal class n_2 samples, and each sample is comprised of d features, the task is to search for a well-performed feature subset from d features that discriminate cancer samples form normal samples.

B. Feasible Solution and Solution Space

Assume that SAA wants to select m features from the original d features, the feasible solution vector of this problem can be represented as x=(x_1, x_2,..., x_m), different components of which must be mutually unequal, and each component of this vector is actually an index in the original feature set, so it must be a positive integer and belongs to interval [1, d]. The solution space is the set of any solution that satisfies the above constraint conditions. That is

$$S = \{(x_1, x_2, \cdots, x_m) \mid 1 \le \forall i \le m : 1 \le x_i \le d; \quad 1 \le \forall i, j \le m, i \ne j : x_i \ne x_j \ \}. \tag{10}$$

C. Objective Function

The objective function of SAA as an optimization approach corresponds to the conception of evaluation function in feature selection problem. As a wrapper method, the feature selection scheme adopts LDA to build classifier because of swiftness and simplicity. The output of the objective function, namely objective value, is represented by the classification error of testing samples and the posterior probabilities of training samples. Hence the objective function is defined as

$$f(x) = 100 \cdot e_c + e_p \tag{11}$$

where e_c is the classification error of the testing set, and e_p is defined as

$$e_p = 1 - \frac{1}{n_{train}} \left\{ \sum_{i=1}^{n_{train}} \max\left[P(c_1 \mid x_i), \cdots, P(c_c \mid x_i)\right] \right\} \tag{12}$$

Evidently, this is a minimization problem, and $f \in [0, 100 + 1 - 1/c]$. In this evaluation function, both training set and testing set employ the same data in order to guarantee the consistency of the output of the objective function for a given feature subset.

D. The Generator of New Solutions

Given the current solution x=(x_1, x_2,..., x_m), the process of generating new solution x' is depicted in the following:

$x' = x;$
$for(i = 1; i \le \lfloor t_k \rfloor + 1; i++)$
$\{r = randz(1, m); rx = randz(1, d); s.t.\ rx \ne x_i, 1 \le i \le m$
$x'_r = rx;\}$

where *randz*(a,b) denotes obtaining a random integer from the interval [a,b] under a uniform distribution.

E. Add a Memory Element

Since SAA can accept the deteriorations to a limited extent with a probability, it may have encountered the optimal solution during the iterative search, but discard it yet later, which lead the final solution is not the best ever encountered. So it is necessary to fix a memory element to hold the best-so-far result during searching. Finally, the value stored in the memory element is compared with the terminative solution, and the better of them is delivered up to the experimenters as the best-of-run solution.

F. Cooling Schedule

The Cooling schedule specified a finite sequence $\{t_k\}$, (k=0,1,2,...), and corresponding Markov chain with a finite number of transitions at each t_k. Cooling schedule vitally impacts the performance, even the convergence, of SAA. A suitable schedule can guarantee SAA to obtain high-quality solutions. In this study, a dynamic cooling schedule is presented.

1) The Initial Value of the Control Parameter t_0: Control parameter t_k limits the range of searching, determines the probability of accepting a worse solution, and dominates the extent of perturbation when generating a new solution based on current one. In order to analyze t_0, acceptance ratio $w(t_k)$ is defined as the ratio of the number of accepted transitions to the number of proposed transitions at temperature t_k. Value t_0 should lead $w(t_0) \approx 1$. There are several theoretical guidelines for setting t_0, but choosing appropriate t_0 needs trials and error until the acceptance ratio approximately approaches to 1.

2) The Decrement Function for Lowering the Value of Control Parameter: This model uses exponent decrement function. That is

$$t_{k+1} = \beta t_k, k = 0,1,2,\cdots \quad (13)$$

where t_k is the current control parameter, t_{k+1} is the next one, and β is a positive constant, typically $\beta \in [0.80, 0.99]$.

3) The Final Value of the Control Parameter Specified by Stop Criteria: The final value of the control parameter determines the accuracy of the final solution. The smaller the final value of the control parameter is, the better the final solution is. In theory, the final value of control parameter is equal to 0. In practice, the algorithm will terminate when lowering the control parameter to a sufficient small value, for example 10^{-4}. The algorithm will also end when the average change in best-so-far value of the objective function at the end of each Markov chain is less than a sufficient small value, for example 10^{-6}, over a number of consecutive Markov chains, for example 800.

4) The Finite Length of Each Markov Chain: Kirkpatrick *et al.* directly assigned the size of problem to the length of Markov chain L_k [16], and Johnson utilized a polynomial of the size of neighborhoods in the problem instance at hand [20]. We present:

$$L_k = a_1 t_k^{-1/6} + a_2 m^2 + a_3 m + a_4 \quad (14)$$

where a_1, a_2, a_3, a_4 are constant coefficients. The first term adjusts the length dynamically during runtime in terms of the current value of control parameter, the second and the third terms is the polynomial of the size of problem , and the last term avoid short chain when the size of problem is small.

IV. Experiments and Results

The prostate protein mass spectra dataset for the experiments in this study was acquired from the FDA-NCI Clinical Proteomics Program Databank (http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp). The spectra were collected utilizing the H4 protein chip, prepared manually using the recommended protocol, and a Ciphergen PBS1 SELDI-TOF mass spectrometer. This dataset is generally named as PC-H4 dataset in [5], [10]. PC-H4 contains 322 total samples, and each sample is composed of 15154 features. We combined 26 samples with prostate cancer with prostate-specific antigen (PSA) levels 4-10 and 43 samples with prostate cancer with PSA levels greater than 10 into normal class, while 190 samples with benign prostate hyperplasia with PSA levels greater than 4 and 63 samples with no evidence of disease with PSA level less than 1 into cancer class.

Firstly, we employed SAA to select 1, 2, 6, and 8 features respectively from the original 15154 features by using the full samples or 2/3 samples of PC-H4 data, respectively. At the beginning of executing SAA, we initialized the parameters of it with $L_k = 100 t_k^{-1/6} + m^2 + m + 100$, $t_0 = 150$, and $t_f = 10^{-4}$. Once the average change in the best-so-far value of the objective function at the end of each Markov chain is less than 10^{-6} over 800 consecutive Markov chains, the algorithm will also stop. Different-sized feature subsets selected by SAA are shown in TABLE I.

After feature selection, we fed these feature subsets to LDA classifier respectively to validate them. 3-fold cross-validation experiments are performed and TABLE II shows the results.

TABLE I
FEATURES SELECTED BY SAA

Size of subset [a]	M/Z Features (dalton)	value of objective function
SA(*a*,1)	*501.2661*	14.1677
SA(*p*,1)	*501.2661*	12.7302
SA(*a*,2)	232.8369, *501.2661*	10.7178
SA(*p*,2)	80.4418, *501.2661*	8.03594
SA(*a*,6)	125.6354, 346.3075, 378.6367, *501.2661*, 4104.6598, 4845.703	2.85727
SA(*p*,6)	114.992, 126.0541, 301.3277, *501.2661*, 749.7643, 4090.3226	1.91991
SA(*a*,8)	141.5938, 233.1218, 358.9246, *501.2661*, 683.8196, 4080.7784, 5319.2853, 19850.614	2.57743
SA(*p*,8)	99.9101, 125.4262, 321.4157, 384.1042, 3632.3908, 4095.0989, 4853.5019, 6185.017	1.43882

[a] SA(*a*,*m*) denotes selecting *m* features by using the overall samples from PC-H4 dataset. SA(*p*,*m*) means selecting *m* features by using 2/3 data of PC-H4 dataset

From TABLE I, we found the feature 501.2661 daltons repeatedly appeared in the feature subsets, and it achieved an accuracy of 85.71%, as shown in TABLE II. So we can infer that it is a rather significant feature to differentiate the prostate cancer samples from the healthy samples. Further

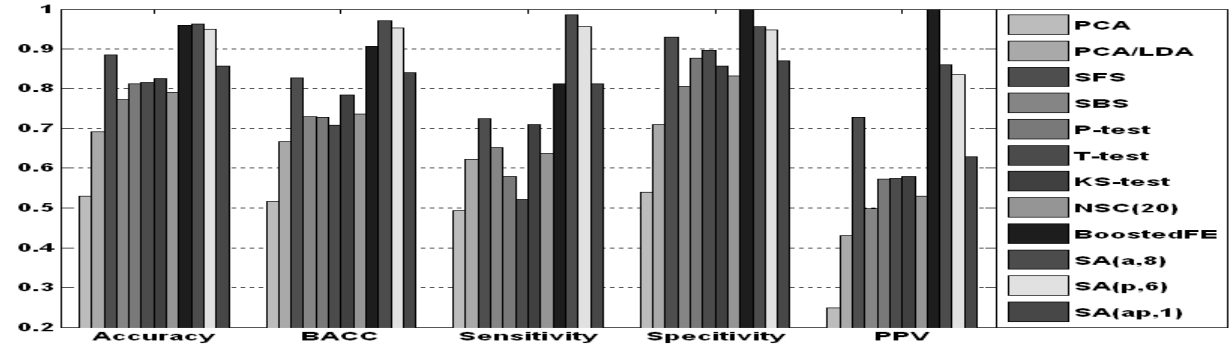

Fig. 1. Performance comparison for PC-H4 dataset. The first seven bars show the performance of algorithms in [5], and the last three bars show the performance of SAA. SA(*a*,*m*) denotes selecting *m* features with overall samples from PC-H4 dataset. SA(*p*,*m*) means selecting m features with 2/3 data of PC-H4 dataset. SA(a*p*,*m*) denotes using SA(*a*, *m*) or SA(p, *m*).

more, it can serve as a clue to find a biomarker in prostate proteomic pattern recognition. When selecting 6, and 8 features by using the overall data of PC-H4 dataset, an accuracy of 96.27% can be gained, whereas an accuracy of 95.03% can be obtained when selecting 6 features using 2/3 data of PC-H4 dataset. A sensitivity of 98.55% can be achieved by selecting 8 features with the full data of PC-H4 dataset, whereas a sensitivity of 95.65% can be obtained by selecting 6 features using 2/3 data of PC-H4. A specificity of 97.63% can be obtained through selecting 6 features using the entire data, whereas a specificity of 94.86% can be yielded by selecting the same number of features from 2/3 data of PC-H4 dataset.

Some filter methods and wrapper methods were presented and performed on PC-H4 dataset in [5]. We compared SAA with those approaches with respect to classification accuracy,

TABLE II
PERFORMANCE OF CLASSIFICATION STATISTICS FOR PC-H4 DATASET

Size of subset [a]	Accuracy	BACC	Sensitivity	Specificity	PPV	NPV
SA(*a*,1)	0.8571	0.8406	0.8116	0.8696	0.6292	0.9442
SA(*p*,1)	0.8571	0.8406	0.8116	0.8696	0.6292	0.9442
SA(*a*,2)	0.8820	0.8881	0.8986	0.8775	0.6667	0.9694
SA(*p*,2)	0.8665	0.8518	0.8261	0.8775	0.6477	0.9487
SA(*a*,6)	0.9627	0.9446	0.913	0.9763	0.9130	0.9763
SA(*p*,6)	0.9503	0.9526	0.9565	0.9486	0.8354	0.9877
SA(*a*,8)	0.9627	0.971	0.9855	0.9565	0.8608	0.9959
SA(*p*,8)	0.9379	0.9394	0.942	0.9368	0.8025	0.9834

[a] SA(*a*,*m*) denotes selecting *m* features by using the overall samples from PC-H4 dataset. SA(*p*,*m*) means selecting *m* features by using 2/3 data of PC-H4 dataset

BACC, sensitivity, specificity, and PPV, as shown in Fig. 1. We observed that SAA outperforms BoostedFE when both of them select 8 features in terms of accuracy, BACC, and sensitivity, and also outperforms the other methods except BoostEF for overall performance parameters. When selecting 1 and 6 features, SAA also achieves good performance.

V. CONCLUSION

Protein mass spectrometry is a promising technique to the detection of early-stage cancer. But the high dimensionality must be reduced before classification or prediction. This paper presents a wrapper approach based on simulated annealing algorithm. Results of experiments show that it performs well on PC-H4 dataset.

We suggest that biologists pay attention to and make further research on the proteins or peptides according to M/Z value 501.2661 daltons. The neighborhood function, the perturbation mechanism, and the new solution generator of SAA maybe need to be improved for a more effective performance for feature selection. And an adaptive cooling schedule should be explored in the future. Some class separability measures may be used to design SAA based filter algorithms, in order to make the process of feature selection faster. In addition, we should endeavor to develop the other intelligent optimization approaches for feature selection for mass spectra.

REFERENCES

[1] T. P. Conrads and T. D. Veenstra, "Diagnostic Proteomics," in *Business briefing: future drug discovery 2003*, 3rd ed. E. Boulton, Ed. London: Business Briefings Ltd., 2003, pp. 88-93.

[2] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature selection," *IEEE Trans. Computers*, vol. C-26(9), pp. 917–922, Sep. 1977.

[3] M. Dash and H. Liu, "Feature selection for classification", *Intelligent Data Analysis*, vol. 1, pp. 131-156, 1997.

[4] P. Langley, "Selection of relevant features in machine learning," in *Proc. the AAAI Fall Symposium on Relevance*, New Orleans, 1994, pp. 1-5.

[5] I. Levner, "Feature selection and nearest centroid classification for protein mass spectrometry," *BMC Bioinformatics*, vol. 6, Mar. 2005.

[6] J. S. Yu, S. Ongarello, R. Fiedler, X.W. Chen, G. Toffolo, C. Cobelli, , and Z. Trajanoski, "Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data," *Bioinformatics*, vol. 21, pp. 2200–2209, 2005.

[7] B. Adam, Y. Qu, J. M. Davis, M. D. Ward, M. A. Clements, L. H. Cazares, O. J. Semmes, P. F. Schellhammer, Y. Yasui, Z. Feng, and G. L. Jr. Wright, "Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men," *Cancer Research*, vol. 62, pp. 3609-3614, Jul. 2002.

[8] Y. Liu, "Feature Extraction for Mass Spectrometry Data," in *LSMS ,LNBI*, vol. 4689, K. Li, Ed. Berlin/Heidelberg: Springer-Verlag, 2007, pp. 188-196.

[9] Y. Qu, B. Adam, Y. Yasui,M. D. Ward, L. H. Cazares, P. F. Schellhammer, Z. Feng, O. J. Semmes, and G. L. Jr. Wright, "Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients," *Clinical Chemistry*, vol. 48(10), pp.1835-1843, Oct. 2002.

[10] R. H. Lilien, H. Farid, and B. R. Donald, "Probabilistic disease classification of expression-dependent proteomic data from mass

spectrometry of human serum," *Computational Biology*, vol. 10(6), pp. 925-946, Oct. 2003.
[11] N. O. Jeffries, "Performance of a genetic algorithm for mass spectrometry proteomics," *BMC Bioinformatics*, vol. 5, Nov. 2004.
[12] W. Michael, D. N. Naik,S. Kasukurti, A. Pothen,R. R. Devineni, B.L. Adam, O.J. Semmes, and G. L. Jr. Wright, "Computational protein biomarker prediction: a case study for prostate cancer," *BMC Bioinformatics*, vol. 5, 2004.
[13] E. F. Petricoin, D. K. Ornstein, C. P. Paweletz, A. Ardekani, P. S Hackett, B. A. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C. Simone, P. J. Levine, W. M. Linehan, M.R. Emmert-Buck, S. M. Steinberg, E.C. Kohn, and L. A. Liotta: "Serum proteomic patterns for detection of prostate cancer," *Journal of the National Cancer Institute*, vol. 94(20), pp.1576-1578, Oct. 2002.
[14] J. D. Wulfkuhle, L. A. Liotta, and E. F. Petricoin, "Proteomic applications for the early detection of cancer," *Nature Reviews*, vol. 3, pp. 267-275, Apr. 2003.
[15] M. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087-1092, Jun. 1953.
[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220(4598), pp. 671-680, May 1983.
[17] E. Aarts, J. Korst, and W. Michiels, "Simulated Annealing," in S*earch methodologies introductory tutorials in optimization and decision support techniques*, chapter 7, Ed. E. K. Burke and G. Kendall, Springer US, 2005, pp. 187-210.
[18] M. W.Trosset, "What is Simulated Annealing?," *Optimization and Engineering*, vol. 2, pp. 201-213, 2001.
[19] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp.179-188, 1936.
[20] D. S. Johnson , C. R. Aragon , L. A. McGeoch , and C. Schevon, "Optimization by simulated annealing: an experimental evaluation. Part I, graph partitioning," *Operations Research*, vol. 37(6), pp.865-892, Nov–Dec. 1989.