Feature Selection for Microarray Gene Expression Data using Simulated Annealing guided by the Multivariate Joint Entropy

Félix Fernando González-Navarro*, Lluís A. Belanche-Muñoz†

*Instituto de Ingeniería
Universidad Autónoma de Baja California, Mexicali, México
fernando.gonzalez@uabc.edu.mx

†Dept. de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya, Barcelona, Spain
belanche@lsi.upc.edu

Abstract—In this work a new way to calculate the multivariate joint entropy is presented. This measure is the basis for a fast information-theoretic based evaluation of gene relevance in a Microarray Gene Expression data context. Its low complexity is based on the reuse of previous computations to calculate current feature relevance.

The μ-TAFS algorithm –named as such to differentiate it from previous TAFS algorithms– implements a simulated annealing technique specially designed for feature subset selection. The algorithm is applied to the maximization of gene subset relevance in several public-domain microarray data sets. The experimental results show a notoriously high classification performance and low size subsets formed by biologically meaningful genes.

Keywords—Feature Selection; Microarray Gene Expression Data; Multivariate Joint Entropy; Simulated Annealing.

I. INTRODUCTION

In cancer diagnosis, classification of the different tumor types is of great importance. An accurate prediction of different tumor types provides better treatment and toxicity minimization on patients. Traditional methods of tackling this situation are primarily based on morphological characteristics of tumorous tissue [1]. These conventional methods are reported to have several diagnosis limitations. In order to analyze the problem of cancer classification using gene expression data, more systematic approaches have been developed [2].

Pioneering work in cancer classification by gene expression using DNA microarray showed the possibility to help the diagnosis by means of Machine Learning or more generally Data Mining methods [3], which are now extensively used for this task [4]. However, in this setting gene expression data analysis entails a heavy computational consumption of resources, due to the extreme sparseness compared to standard data sets in classification tasks [5].

Typically, a gene expression data set may consist of dozens of observations but with thousands or even tens of thousands of genes. Classifying cancer types using this very high ratio between number of variables and number of observations is a delicate process. As a result, dimensionality reduction and in particular feature subset selection (FSS) techniques may be very useful. Finding small subsets of very relevant genes among a huge quantity could derive in much specific and efficient treatments.

This work addresses the problem of selecting a subset of features by using the TAFS (Thermodynamic Algorithms for Feature Selection) family of methods for the FSS problem. Given a suitable objective function, the algorithm makes uses of an special-purpose simulated annealing (SA) technique to find a good subset of features that maximizes the objective function. A distinctive characteristic of TAFS over other search algorithms for FSS is its probabilistic capability to accept momentarily worse solutions, which in the end may result in better hypotheses. Despite their powerful optimization capability, SA-based search algorithms usually lack execution speed, involving long convergence times. In consequence, they have been generally excluded as an option in FSS problems, let alone in highly complex domains such as microarray gene expression data. A few contributions using the classical SA algorithm for FSS are found in prostate protein mass spectrometry data [6], marketing applications [7], or parameter optimization in clustering gene expression analysis [8].

Our answer to these computational problems is twofold. First, we use a filter objective function for FSS (thus avoiding the development of a predictive model for every subset evaluation). Second, the objective function itself is evaluated very efficiently based on the reutilization of previous computations.

Specifically, a new way to calculate the multivariate joint entropy for categorical variables is presented that is both exact and very efficient. This measure is then used by a SA-based TAFS algorithm to search for small subsets of highly relevant genes in five public domain microarray data sets. Classification experiments yield some of the best
results reported so far for these data sets and offer a drastic reduction in subset sizes.

The paper is organized as follows: section II briefly reviews the Simulated Annealing technique; section III reviews and motivates the previous Thermodynamic Algorithms for feature subset selection; section IV develops the information-theoretic measure for feature relevance and its efficient implementation; section V describes the data sets and the experimental settings; section VI presents the results and their interpretation. The paper ends with the conclusions and directions for future work.

II. SIMULATED ANNEALING

Simulated Annealing (SA) is a stochastic technique inspired on statistical mechanics for finding (near) globally optimal solutions to large optimization problems. SA is a weak method in that it needs almost no information about the structure of the search space. The algorithm works by assuming that some parts of the current solution belong to a potentially better one, and thus these parts should be retained by exploring neighbors of the current solution. Assuming the objective function is to be minimized, then SA would jump from hill to hill and hence escape or simply avoid sub-optimal solutions.

When a system \( S \) (considered as a set of possible states) is in thermal equilibrium (at a given temperature \( T \)), the probability that it is in a certain state \( s \), called \( P_T(s) \), depends on \( T \) and on the energy \( E(s) \) of the state \( s \). This probability follows a Boltzmann distribution:

\[
P_T(s) = \frac{\exp \left( \frac{-E(s)}{kT} \right)}{Z}, \text{ with } Z = \sum_{s \in S} \exp \left( \frac{-E(s)}{kT} \right)
\]

where \( k \) is the Boltzmann constant and \( Z \) acts as a normalization factor. Metropolis and his co-workers developed a stochastic relaxation method that works by simulating the behavior of a system at a given temperature \( T \) [9]. Being \( s \) the current state and \( s' \) a neighboring state, the probability of making a transition from \( s \) to \( s' \) is the ratio \( P_T(s \rightarrow s') \) between the probability of being in \( s \) and the probability of being in \( s' \):

\[
P_T(s \rightarrow s') = \frac{P_T(s')}{P_T(s)} = \exp \left( -\frac{\Delta E}{kT} \right)
\]

where we have defined \( \Delta E = E(s') - E(s) \). Therefore, the acceptance or rejection of \( s' \) as the new state depends on the difference of the energies of both states at temperature \( T \). If \( P_T(s') \geq P_T(s) \) then the “move” is always accepted. It \( P_T(s') < P_T(s) \) then it is accepted with probability \( P_T(s, s') < 1 \) (this situation corresponds to a transition to a higher-energy state).

Note that this probability depends upon the current temperature \( T \) and decreases as \( T \) does. In the end, there will be a value of \( T \) low enough (the freezing point), wherein these transitions will be very unlikely and the system will be considered frozen. In order to maximize the probability of finding states of minimal energy at every value of \( T \), thermal equilibrium must be reached. To do this, according to Metropolis, an annealing schedule is designed to prevent the process from getting stuck at a local minimum. The SA algorithm introduced in [10] consists in using the Metropolis idea at each temperature \( T \) for a finite amount of time. In this algorithm \( T \) is first set at a initially high value, spending enough time at it so to approximate thermal equilibrium. Then a small decrement of \( T \) is performed and the process is iterated until the system is considered frozen.

If the cooling schedule is well designed, the final reached state may be considered a near-optimal solution. However, the whole process is inherently slow, mainly because of the thermal equilibrium requirement at every temperature \( T \).

III. THERMODYNAMIC ALGORITHMS FOR FSS

In this section we review TAFS (Thermodynamic Algorithm for Feature Selection) and eTAFS, two algorithms for FSS that were originally designed for problems of moderate feature size (up to one hundred) [11]. If we consider FSS as a search of possible feature subsets of the full feature set \( \mathcal{X} \), then SA acts as a combinatorial optimization process [12]. In this sense, TAFS and eTAFS find a subset of features that optimize the value of a given objective function \( J : \mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R} \). From now on, we assume that this function is to be maximized and that \( J \geq 0 \).

To this end, a special-purpose forward/backward mechanism is embedded into an SA algorithm, taking advantage of its most distinctive characteristic, the probabilistic acceptance of worse scenarios over a finite time. This characteristic is enhanced by the notion of an \( \epsilon \)-improvement: a feature \( \epsilon \)-improves a current solution if it has a higher value of the objective function or a value not worse than \( \epsilon \% \). This mechanism is intended to account for noise in the evaluation of the objective function (caused either by the finiteness of the data set or introduced by the chosen resampling method). The pseudo-code of TAFS is depicted in Algorithm 1.

Algorithm 1. The algorithm consists of two major loops. The outer loop waits for the inner loop to finish and then updates \( T \) according to the chosen cooling schedule. When this loop reaches \( T_{\text{min}} \), the algorithm halts. It keeps track of the best solution found (which is not necessarily the current one).

The inner loop is the core of the algorithm and is composed of two interleaved procedures: Forward and Backward, that iterate until an equilibrium point is found. These procedures work independently of each other, but share information about the results of their respective search in the form of the current solution. Within them, FSS takes

\[1\]This is the case for accuracy, mutual information, distances and many other useful measures.
place and the mechanism to escape from local minima starts working. These procedures iteratively add or remove features one at a time in such a way that an $\epsilon$-improvement is accepted unconditionally, whereas a non-$\epsilon$-improvement is accepted probabilistically. The pseudo-code for Forward and Backward, and $\epsilon$-improvement is outlined in Algorithms 2, 3 and 4. When Forward and Backward finish their respective tasks, TAFS checks if the current solution is the same as it was prior to their execution. If this is the case, then we consider that thermal equilibrium has been reached and $T$ is adjusted, according to the cooling schedule. If it is not, another loop of Forward and Backward is carried out.

A. $\epsilon$TAFS: an Enhanced TAFS Algorithm

A modification to Algorithm 1 aimed at speeding up relaxation time is presented in this section. The algorithm – named $\epsilon$TAFS, see Algorithms 5 and 6 – is endowed with a feature search window (of size $l$) in the backward step, as follows. In forward steps always the best feature is added (by looking all possible additions). In backward steps this search is limited to $l$ tries at random (without replacement). The value of $l$ is incremented by one at every thermal-equilibrium point. This mechanism is an additional source of non-determinism and a bias towards adding a feature only when it is the best option available. On the contrary, to remove one, it suffices that its removal $\epsilon$-improves the current solution. Another direct consequence is of course a considerable speed-up of the algorithm. Note that the design of $\epsilon$TAFS is such that it grows more and more deterministic, informed and costly as it converges towards the final configuration.

**Algorithm 1: TAFS algorithm for feature selection**

```plaintext
input : $X$: Full Feature set \{X_1...X_n\}
J(): Objective Function 
$\alpha()$: Cooling Schedule 
$\epsilon$: Epsilon T0: Initial Temperature Tmin: Final Temperature 

1 $X_{cur} \leftarrow \emptyset$ Initial current subset 
2 $J_{cur} \leftarrow 0$ Initial objective function value 
3 $T \leftarrow T0$ Initial temperature 
4 while $T > T_{min}$ do 
5 repeat 
6 $Y \leftarrow X_{cur}$ 
7 $\text{Forward (}X_{cur},J_{cur}\text{)}$ 
8 $\text{Backward (}X_{cur},J_{cur}\text{)}$ 
9 until $Y = X_{cur}$ 
10 $T \leftarrow \alpha(T)$ 
```

**Algorithm 2: Procedure Forward ($Z, J_Z$ are modified)**

```plaintext
input : $Z, J_Z$

1 repeat 
2 $x \leftarrow \text{argmax} J(Z \cup \{X_i\})_{X_i \in Z}$ 
3 if $>_\epsilon (Z, x, true)$ then 
4 accept $\leftarrow true$ 
5 else 
6 $\Delta J \leftarrow J(Z \cup \{x\}) - J(Z)$ 
7 accept $\leftarrow \text{rand}(0,1) < e^{\frac{\Delta J}{T}}$ 
8 if accept then 
9 $Z \leftarrow Z \cup \{x\}$ 
10 if $J(Z) > J_{cur}$ then 
11 $J_Z \leftarrow J(Z)$ 
12 until not accept 
```

**Algorithm 3: Procedure Backward ($Z, J_Z$ are modified)**

```plaintext
input : $Z, J_Z$

1 repeat 
2 $x \leftarrow \text{argmax} J(Z \setminus \{X_i\})_{X_i \in Z}$ 
3 if $>_\epsilon (Z, x, false)$ then 
4 accept $\leftarrow true$ 
5 else 
6 $\Delta J \leftarrow J(Z \setminus \{x\}) - J(Z)$ 
7 accept $\leftarrow \text{rand}(0,1) < e^{\frac{\Delta J}{T}}$ 
8 if accept then 
9 $Z \leftarrow Z \setminus \{x\}$ 
10 if $J(Z) > J_{cur}$ then 
11 $J_Z \leftarrow J(Z)$ 
12 until not accept 
```

**Algorithm 4: Function $>\epsilon$**

```plaintext
input : $Z, x, d$ 
output: boolean 

1 if $d$ then 
2 $Z' \leftarrow Z \cup \{x\}$ 
3 else 
4 $Z' \leftarrow Z \setminus \{x\}$ 
5 $\Delta x \leftarrow J(Z') - J(Z)$ 
6 if $\Delta x > 0$ then 
7 return true 
8 else 
9 return $\frac{\Delta x}{J(Z)} < \epsilon$ 
```

**Algorithm 5: $\epsilon$TAFS algorithm for feature selection**

```plaintext
input : $X$: Full Feature set \{X_1...X_n\} 
J(): Objective Function 
$\alpha()$: Cooling Schedule 
$\epsilon$: Epsilon T0 Initial Temperature T_{min} Final Temperature 

1 $X_{cur} \leftarrow \emptyset$ Initial current subset 
2 $J_{cur} \leftarrow 0$ Initial objective function value 
3 $T \leftarrow T_{0}$ Initial temperature 
4 $l \leftarrow 2$ Window size (for backward steps) 
5 while $T > T_{min}$ do 
6 repeat 
7 $Y \leftarrow X_{cur}$ 
8 $\text{Forward (}X_{cur},J_{cur},l\text{)}$ 
9 $\text{Backward (}X_{cur},J_{cur},l\text{)}$ 
10 until $Y = X_{cur}$ 
11 $T \leftarrow \alpha(T)$ 
12 $l \leftarrow l + 1$ 
```


Algorithm 6: eTAFS Backward procedure ($Z, J_Z$ are modified). Note that $X_0$ can be efficiently computed while in the for loop.

\begin{tabular}{|c|}
\hline
\textbf{input} : $Z, J_Z$; \\
\textbf{1} $A \leftarrow \emptyset$; $A_B \leftarrow \emptyset$ \hfill \\
\textbf{2} repeat \hfill \\
\hspace{1cm} for $i := 1$ to $\min(l, |Z|)$ do \hfill \\
\hspace{2cm} Select $x \in Z \setminus A_B$ randomly \hfill \\
\hspace{2cm} if $>\varepsilon$ (Z, x; false) then \hfill \\
\hspace{3cm} $A \leftarrow A \cup \{x\}$ \hfill \\
\hspace{3cm} $A_B \leftarrow A_B \cup \{x\}$ \hfill \\
\hspace{2cm} $X_0 \leftarrow \arg\max_{X \in A_B} (J(Z \setminus \{X\}))$ \hfill \\
\hspace{2cm} if $X_0 \in A$ then \hfill \\
\hspace{3cm} accept $\leftarrow$ true \hfill \\
\hspace{2cm} else \hfill \\
\hspace{3cm} $\Delta J \leftarrow J(Z \setminus \{X_0\}) - J(Z)$ \hfill \\
\hspace{3cm} accept $\leftarrow \text{rand}(0,1) < e^{\Delta J}$ \hfill \\
\hspace{2cm} if accept then \hfill \\
\hspace{3cm} $Z \leftarrow Z \setminus \{X_0\}$ \hfill \\
\hspace{2cm} if $J(Z) > J_Z$ then \hfill \\
\hspace{3cm} $J_Z \leftarrow J(Z)$ \hfill \\
\textbf{3} until not accept \hfill \\
\hline
\end{tabular}

IV. INFORMATION THEORETIC FEATURE RELEVANCE
A. Entropy definitions

Entropy, a main concept in information theory [13], can be seen as an average of uncertainty in a random variable. If $X$ is a discrete random variable with probability mass function $p$, its entropy is defined by$^2$:

$$H(X) = - \sum_x p(x) \log p(x) = -E_X[\log p(X)] \quad (2)$$

being $E[]$ the expectation operator. If a variable ($X$) is known and another one ($Y$) is not, the conditional entropy of $Y$ with respect to $X$: is the mutual entropy with respect to the corresponding conditional distribution:

$$H(Y|X) = - \sum_x \sum_y p(x, y) \log p(y|x) \quad (3)$$

From these two definitions another concept is build, the mutual information (MI), which can be interpreted as a measure of the information that a random variable has or explains about another one:

$$I(X; Y) = H(Y) - H(Y|X) = E_{X,Y}[\log p(x, y)/p(x)p(y)] \quad (4)$$

The computation of the MI can be extended from the bivariate to the multivariate case of a number $n \geq 2$ of variables, against another one:

$$I(X_1, \ldots, X_n; Y) = \sum_{i=1}^n I(X_i; Y|X_1, \ldots, X_{i-1}) = H(Y) - H(Y|X_1, \ldots, X_n). \quad (5)$$

Conditional MI is expressed in the natural way, by conditioning in (4):

$$I(X; Y|Z) = H(Y|Z) - H(Y|X, Z) \quad (6)$$

The MI has been used with success as for feature selection in machine learning tasks. Currently there is no agreed-upon definition of the general multivariate mutual information $I(X_1; \ldots; X_n)$. An existent proposal is the interaction information, described e.g. in [14] which, for the case of three variables $X, Y, Z$, is defined as $I(X; Y; Z) = I(X; Y|Z) - I(X; Y)$. The extension to the multivariate case is in terms of the marginal entropies and is given by:

$$I(X_1; \ldots; X_n) = - \sum_{\tau \subseteq \{X_1, \ldots, X_n\}} (-1)^{n-|\tau|}H(\tau).$$

This definition is impractical due to its exponential character. In the next section, the objective function $J$ takes the form of an information-theoretic index of relevance based on the multivariate joint entropy, which has already been used elsewhere [15]. One of the contributions of this paper resides in a fast implementation of the calculation and its application to microarray gene expression data.

B. Incremental Multivariate Joint Entropy

For a random variable $X$, it is known that the joint entropy obeys the following property:

$$H(X, Y) \geq H(X) \quad (7)$$

This property says that joint entropy is always at least equal to the entropies of the original system: adding a new variable can never reduce the available uncertainty. If we rewrite (7) as an equation:

$$H(X, Y) = H(X) + \Delta_X(Y), \quad (8)$$

then $\Delta_X(Y) \geq 0$ represents the increment in entropy due to the addition of the variable $Y$ to the system. In a feature selection setting, given $Z$ a class variable, $\tau \subset X$ the current subset and $H(\tau)$ its joint entropy, if a new feature $X_i \in X \setminus \tau$ is considered for possible inclusion in the current subset:

$$H(Z, \tau \cup \{X_i\}) = H(Z, \tau) + \Delta_{Z, \tau}(X_i) \quad (9)$$

It turns out that, to obtain the next calculation, it is computationally far more advantageous to store $H(Z, \tau)$ and calculate the quantity $\Delta_{Z, \tau}(X_i)$ than to compute the full

$^2$All log are to base 2.
The notation \( H(Z, \tau) \) calculates starting joint entropy as well as its first MES for two binary variables \( X_1 \) and \( X_2 \), into two entropy values: \( H(X_1) \) contribution is produced by the action of \( X_1 \), and \( H(X_2) \) for two binary variables \( X_1 \) and \( X_2 \), its corresponding entropy value. An example of a MES table with its unique forming patterns is indicated in the left part of Table I. The two unique values and their entropies \( H(X_1 = 0) = 0.481 \) and \( H(X_1 = 1) = 0.515 \) are calculated. Let us suppose that a feature \( X_2 \) is to be evaluated w.r.t the current subset \( \tau = \{ X_1 \} \). The MES table with its unique forming patterns are indicated in the right part of Table I. We can see that by introducing \( X_2 \) to the current subset \( \tau \), four partitions are generated for each unique value of \( X_1 \): \{00, 01, 10, 11\}. In the particular case of \( X_1 = 0 \), a change in its entropy contribution is produced by the action of \( X_2 \) by splitting it into two entropy values: \( H(X_1 = 0, X_2 = 0) = 0.488 \) and \( H(X_1 = 0, X_2 = 1) = 0.523 \), for a total entropy of \( H(X_1 = 0, X_2) = 1.011 \). The increment in entropy \( \Delta_{\tau} \) is obtained as the difference between the current MES (considering the addition of \( X_2 \)) and the previous scheme without it —see Table II.

\[
\begin{array}{c|c|c|c}
\hline
\Delta_{\tau}(X_1) & H(X_1, X_2) & -P(X_1) \log P(X_1) & \text{difference} \\
\hline
\Delta_{\tau}(X_1 = 0) & 1.011 & 0.481 & 0.531 \\
\Delta_{\tau}(X_1 = 1) & 0.939 & 0.515 & 0.424 \\
\hline
\end{array}
\]

Table II

\[ \Delta_{\tau} \] computations from the Marginal Entropy Scheme —see Table I.

Finally, this last value is applied to eq. (9) to obtain the joint entropy \( H(X_1, X_2) = H(X_1) + \Delta_{\tau}(X_2) = 0.996 + 0.954 = 1.950 \). The listings in Algorithms 7 and 8 show the pseudo-code to compute the procedure explained above. The notation \( D|\tau \) stands for the restriction of the dataset \( D \) to the features in \( \tau \).

Initial entropy is evaluated in lines 2-5. This first step calculates starting joint entropy as well as its first MES (lines 4-5), which will be taken as input to the next computation. Note that these two lines can be efficiently implemented as one function and using only one loop-cycle, with complexity \( \theta(|D|) \), where \( |D| \) is the number of training instances.

In the else part of the if clause, the MES is calculated with the addition of \( X_1 \) to the current subset \( \tau \) (named \( E_{r+} \)). Taking into account that previous MES inherits the ordering sequence derived from a previous stage (because of lines 5 and 9), entropies generated by changes in the MES given by \( \tau \cup \{ X_1 \} \) are summed \( (E_{r+}) \) in groups (line 11) by the newly formed patterns, rendering a one-to-one correspondence between previous MES and current MES.

**Algorithm 7: Incremental Multivariate Joint Entropy**

**Algorithm 8: MarginalEntropyScheme Function**

Thus, the entropy contribution \( \Delta_{\tau}(X_1) \), showing the effect of adding \( X_1 \) to \( \tau \), is computed by the difference in both MESs (line 11), being finally added to the current entropy.
The implementation of lines 10-11 follows the same consideration as lines 4-5, and hence complexity is in the same order.

The incremental multivariate joint entropy is used to obtain an index of relevance (acting as the objective function) of a feature $X_i \in \mathcal{X}$ to a class $Z$ with respect to a subset $\tau \subset \mathcal{X} \setminus X_i$ and is defined by:

$$J(X_i; Z|\tau) = \frac{H(Z) + H(\tau, X_i) - H(\tau, Z, X_i)}{H(Z)}$$  \hspace{1cm} (10)

Note the denominator acts as a normalization factor, such that $J \in [0, 1]$, with $J = 1$ corresponding to the highest relevance. The reward of using this objective function by a TAFS-like algorithm consists in the possibility of testing it in highly complex domains such as microarray data sets. We name the combination of $\epsilon$TAFS and the objective function in eq. (10) as the $\mu$-TAFS algorithm.

V. EXPERIMENTAL WORK

To compute the necessary entropies described in previous section, a discretization process is needed. This change of representation does not often result in a significant loss of accuracy (sometimes significantly improves it [16], [17]); it also offers reductions in learning time [18]. In this work, the CAIM algorithm was selected for two reasons: it is designed to work with supervised data, and does not require the user to define a specific number of intervals [19].

A. Data sets

Five public-domain microarray gene expression data sets are used to test and validate the approach proposed in this work: Colon Tumor: 62 observations of colon tissue, of which 40 are tumorous and 22 normal, 2,000 genes [20]. Leukemia: 72 bone marrow observations and 7,129 probes: 6,817 human genes and 312 control genes [3]. The goal is to tell acute myeloid leukemia (AML) from acute lymphoblastic leukemia (ALL). Lung Cancer: distinction between malignant pleural mesothelioma and adenocarcinoma of lung [21]; 181 observations with 12,533 genes. Prostate Cancer: used in [22] to analyze differences in pathological features of prostate cancer and to identify genes that might anticipate its clinical behavior; 136 observations and 12,600 genes. Breast Cancer: 97 patients with primary invasive breast carcinoma; 12,600 genes analyzed [23].

B. Settings

Provided that the core nature of the $\mu$-TAFS algorithm, and even many other algorithms –e.g. Genetic Algorithms, Neural Networks–, resides in their stochasticity, several runs have to be performed, in order to better assess the average behaviour of the methods.

The experimental design to test $\mu$-TAFS algorithm measures performance by carrying out $m = 100$ different independent runs. In each run, $\mu$-TAFS is executed on the corresponding dataset and returns the set of all those feature subsets reaching the best found performance function (maximum relevance, in this case). To overcome the existence of many solutions, the subset that offers the lowest mutual information (MI) among its elements – i.e. the less redundancy– is taken as the subset delivered in this run. After completing the $m$ execution runs, the obtained subsets can be ordered from minimum to maximum MI value.

The $\mu$-TAFS parameters are as follows: $\epsilon = 0.01, T_0 = 0.1$ and $T_{\text{min}} = 0.0001$. These settings were chosen after preliminary fine-tuning and are kept constant for all the problems [11]. The cooling function was chosen to be geometric $\alpha(t) = 0.9t$, following recommendations in the literature [12].

<table>
<thead>
<tr>
<th>Data set</th>
<th>Time</th>
<th>Jeval</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colon Tumor</td>
<td>6.41</td>
<td>505,901</td>
<td>6.93 ± 0.06</td>
</tr>
<tr>
<td>Leukemia</td>
<td>6.51</td>
<td>506,489</td>
<td>3.36 ± 0.06</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>7.45</td>
<td>560,972</td>
<td>2.58 ± 0.04</td>
</tr>
<tr>
<td>Prostate Cancer</td>
<td>98.74</td>
<td>7,119,800</td>
<td>9.85 ± 0.05</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>136.93</td>
<td>10,943,628</td>
<td>9.62 ± 0.03</td>
</tr>
</tbody>
</table>

Table III

VI. EXPERIMENTAL RESULTS

A. $\mu$-TAFS performance results

The evolution of $\mu$-TAFS from a high temperature state to a frozen point is depicted in Fig. 1. Highly unstable –i.e. high temperature condition– readings are observed at the initial stages in each of the datasets. As soon as the algorithm becomes more relaxed due to eq. (1), worse solutions are avoided. The frozen condition is observed at the final stages of each execution, where $J$ values consecutively reach the maximum possible value ($J = 1$) in all cases.

The running performance of $\mu$-TAFS is summarized in Table III. The results show that $\mu$-TAFS yields subsets of considerably low size and also low variability. Notorious readings correspond to Leukemia and Lung Cancer. It is conjectured that such sizes respond to the nature of the proposed information theoretic model on discretized data sets, in the sense that only a few genes significantly contribute to increase the index of relevance given by eq. (10). On the one hand, working with continuous features, the index would tend to vary smoothly –i.e. generating small increments; as a consequence, more features are added-deleted. On the other hand, discrete features variations are normalized by their discretization scheme, so small increments in the real-value are merged into a single discrete value. Therefore, mostly significant increments are truly reflected in its addition-deletion from the current subset.
The obtained solutions are displayed in Table IV. Among the eight classifiers used to test the solutions, only the final model is presented. Lung Cancer, Leukemia and Prostate Cancer reach remarkably high accuracies, while Colon Tumor and specially Breast Cancer show lower 10x10 CV readings. In all cases, the subset that delivers this performance is considerable small, having 7 genes or less (and only 3 genes in the Leukemia data set). Moreover, all Wilcoxon test \( p \)-values signal significant differences \((p < 0.05)\) between the best method and all other methods in the corresponding data set, except for the lSVM vs. LR in Colon Tumor \((p = 0.312)\).

C. Discussion of the results

It is a common practice to compare to similar works in the literature. Unfortunately, the methodological steps are in general very different, especially concerning resampling techniques, making an accurate comparison a delicate undertaking. Nonetheless, such a comparison is presented in Table V. Seven references which are illustrative of recent work are indicated, including previous work from the authors. In this table the validation method, the best classifier and the best reported result are detailed (final accuracy and number of genes involved).

The Colon Tumor data set presents difficulties in classification, never reaching 90\%. The solution delivered by \( \mu \)-TAFS is comparable with the best known (that of BGS\(^3\) [25]); however, it uses 5 genes against the 9 used by BGS\(^3\).

---

**Table IV**

<table>
<thead>
<tr>
<th>Data set</th>
<th>Classifier</th>
<th>10x10 CV size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colon Tumor</td>
<td>lSVM ((C = 2^3))</td>
<td>89.19±0.38 5</td>
</tr>
<tr>
<td>Leukemia</td>
<td>lSVM ((C = 2^3))</td>
<td>99.89±0.07 4</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>LR</td>
<td>99.62±0.27 3</td>
</tr>
<tr>
<td>Prostate Cancer</td>
<td>NN (6)</td>
<td>95.66±0.21 7</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>rSVM ((C = 2^3, \gamma = 2^{-1}))</td>
<td>86.90±0.48 6</td>
</tr>
</tbody>
</table>

---

3For the experiments, we use a MATLAB implementation; specifically, for the SVMs we use the MATLAB interface to LIBSVM [24]. All tests are run on on a regular x86 workstation.

4The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test for the analysis of two related samples, or repeated measurements on a single sample. It can be used as an alternative to the paired Student’s t-test when the population cannot be assumed to be normally distributed. It should therefore be used whenever the distributional assumptions that underlie the t-test cannot be satisfied.
The other difficult problem seems to be Breast Cancer. In this data set, µ-TAFS gives the best result among the references consulted, using also less genes and in front of solutions that employ a pure wrapper strategy. For the other three problems, µ-TAFS is also able to yield better solutions compared to other approaches, many of them using a much bigger gene subset.

Expression levels for each model in the five data sets are given in Fig. 2. It is seen that each model posses genes that are visually identified as the ones that present irregular expression levels: Colon Tumor genes M76378 and T51288; Leukemia genes AFFX-CreX-5.at and L09209; Lung Cancer gene 37157.at; Prostate Cancer genes 38322.at and 37639.at; and Breast Cancer genes Contig14882_RC, Contig53822_RC and Contig57657_RC.

<table>
<thead>
<tr>
<th>Work</th>
<th>Validation</th>
<th>Colon Tumor</th>
<th>Leukemia</th>
<th>Lung Cancer</th>
<th>Breast Cancer</th>
<th>Prostate Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="F">25</a></td>
<td>10x10CV</td>
<td>89.36</td>
<td>97.89</td>
<td>98.84</td>
<td>83.37</td>
<td>93.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(9.3NN)</td>
<td>(2,NB)</td>
<td>(4,LR)</td>
<td>(12.1SVM)</td>
<td>(3,10NN)</td>
</tr>
<tr>
<td><a href="F">26</a></td>
<td>200-B.632</td>
<td>88.75</td>
<td>98.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(14.1SVM)</td>
<td>(23.1SVM)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><a href="W">27</a></td>
<td>10x10CV</td>
<td>85.48</td>
<td>93.40</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3,NB)</td>
<td>(2,NB)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><a href="W">28</a></td>
<td>100-RS</td>
<td>87.31</td>
<td>94.50</td>
<td>-</td>
<td>72.20</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(94.1SVM)</td>
<td>-</td>
<td>(23.1SVM)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><a href="W">29</a></td>
<td>50-HO</td>
<td>77.00</td>
<td>96.00</td>
<td>99.00</td>
<td>79.00</td>
<td>93.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(33.1SVM)</td>
<td>(30.1SVM)</td>
<td>(38.1SVM)</td>
<td>(46.1SVM)</td>
<td>(47.1SVM)</td>
</tr>
<tr>
<td><a href="FW">30</a></td>
<td>10x10CV</td>
<td>-</td>
<td>-</td>
<td>99.40</td>
<td>96.30</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>(135.5NN)</td>
<td>(79.5NN)</td>
<td>-</td>
</tr>
<tr>
<td><a href="F">31</a></td>
<td>10CV</td>
<td>-</td>
<td>98.6</td>
<td>99.45</td>
<td>68.04</td>
<td>91.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>(2,1SVM)</td>
<td>(5,1SVM)</td>
<td>(8,1SVM)</td>
<td>(6,1SVM)</td>
</tr>
</tbody>
</table>

Table V

The genes corresponding to the solutions displayed in Table IV are detailed in Table VI. In the following, known biological evidence is presented about the effect of gene expressions in each cancer disease. This evidence is assembled by examining recent relevant medical literature.

Colon Tumor:

- **M76378 CRSP1-Cysteine and glycine-rich protein 1.** This gene encodes a member of the cysteine-rich protein (CRSP) family. It may be involved in regulatory processes important for development and cellular differentiation. Hypomethylation, a decrease in the epigenetic methylation of cytosine and adenosine residues in DNA, of CRSP1 and other genes were confirmed in the cancer cells by bisulfite sequencing [32].

- **H08393 COL11A2-collagen, type XI, alpha 2 (Homo sapiens).** Two alpha chains of type XI collagen, a minor fibrillar collagen are encoded by this gene [33]. Up-regulation of this gene in the mucosa stromal cells of both familial adenomatosis polyposis and sporadic colorectal cancer has been detected [34].

- **T51849 EPHB1-Tyrosine-protein kinase receptor elk precursor.** EphB1 is a member of receptor tyrosine kinases of the EphB subfamily and has been positively identified in the development, progress and prognosis of colorectal cancers [35].

- **M19131 CALM2-calmodulin 2 (phosphorylase kinase, delta).** Caml2 plays an important role in intracellular calcium signaling, which regulates a variety of cellular processes, such as cell proliferation and gene transcription [36]. Increased expression levels of this gene were found in anaplastic large cell lymphoma cell lines [37].

- **T51288 ASS1-argininosuccinate synthase (human).** Arginine, a semi-essential amino acid in humans, is critical for the growth of human cancers as in primary ovarian, stomach and colorectal cancer, whose expression levels read high values [38].

Leukemia:

- **AFFX-CREX-5_AT** NOT IDENTIFIED.
It is activated during amino-acid deprivation and has been associated with neurodegenerative diseases, cancer, pathogen infections and myopathies [44].

- 33221_at PAXIP1-PAX interacting (with transcription-activation domain) protein 1. Member of the paired box (PAX) gene family, this gene plays a critical role in maintaining genome stability by protecting cells from DNA damage [39], [45]. Analysis of pulmonary adenocarcinomas in experiment GDS1650 in [33] records shows over-expression levels of this gene.

- 40790_at BHLHE40-basic helix-loop-helix family, member e40. This gene encodes a basic helix-loop-helix protein expressed in various tissues, which is may be involved in the control of cell differentiation [33]. Experiments suggest that loss of DEC1 expression is an early event in the development of lung cancer [46].

- 107_at RAB40A-member RAS oncogene family. This gene encodes a member of the Rab40 subfamily of Rab small GTP-binding proteins that contains a C-terminal suppressors of cytokine signaling box [39]. No medical evidence was found in literature about its role in cancer.

Prostate Cancer:

- 1230_at MTMR11-myotubularin related protein 11. Experiments on patients with acute lymphoblastic leukemia and with Burkitt lymphoma, three putative oncogenes or tumor suppressor genes were found, one of them was the MTMR11 [47].

- 38322_at PAGE4-P antigen family, member 4 (prostate associated). This gene is strongly expressed in prostate and prostate cancer; and also expressed in other tissues as in testis, fallopian tube, uterus, placenta, as well as in testicular cancer and uterine cancer [39].

- 37639_at HPN-Hepsin. Hepsin is a cell surface serine protease and plays an essential role in cell growth and maintenance of cell morphology and it is highly related with prostate cancer, benign prostatic hyperplasia [39], [48].

- 660_at CYP24A1-cytochrome P450, family 24, subfamily A, polypeptide 1. This gene encodes a member of the cytochrome P450 superfamily of enzymes. The cytochrome P450 proteins catalyze many reactions involved in drug metabolism and synthesis of cholesterol, steroids and other lipids [39]. This gene has been reported as responsible for degradation of the active vitamin D metabolite 1,25-dihydroxyvitamin D3 which
is known to be antimitotic in prostate cancer cells [49].

- **35998_at** Hypothetical protein LOC284244 (LOC284244). No evidence found.
- **34107_at** PFKFB2-6-phosphofructo-2-kinase/fructose-2,6-bisphosphatase 2. The protein encoded by this gene is involved in the synthesis and degradation of fructose-2,6-bisphosphate, a regulatory molecule that controls glycolysis in eukaryotes [39]. It has been suggested that the induction of **de novo** lipid synthesis—process that protects cancer cells from reactive oxygen species—by androgen requires the up-regulation of HK2 and PFKFB2 [50].

**Breast Cancer:**

- **AB014543** CLUAP1-clusterin associated protein 1 (Homo sapiens). This gene is highly expressed in osteosarcoma, ovarian, colon, and lung cancers [51].
- **Contig57657_RC** PAK1-p21 protein (Cdc42/Rac)-activated kinase 1 (Homo sapiens). This gene encodes a family member of serine/threonine p21-activating kinases, known as PAK proteins, whose role is the regulation of cell motility and morphology [33]. Pak1 is directly related with the Etk/Bmx protein, acting this later as a control to the proliferation and tumorigenic growth of mammary epithelial cancer cells [52].
- **NM_006191** PA2G4-Proliferation-associated 2G4, 38kDa (PA2G4). Also known as EBP1, this gene encodes an RNA-binding protein that is involved in growth regulation [39]. The EBP1 has been shown to be a transcriptional corepressor that inhibits the growth of human breast cancer cell lines [53].
- **Contig14882_RC, Contig53822_RC, Contig53713_RC** NOT IDENTIFIED.

**VII. CONCLUSIONS**

A new algorithm for feature selection using Simulated Annealing guided by the discrete multivariate joint entropy has been introduced and evaluated. Our experimental results concern the search for small subsets of highly relevant genes in five public domain Microarray Gene Expression data samples. The very promising results indicate that the algorithm offers a promising and general framework for feature selection in very high dimensional data sets.

The entropic relevance measure has shown to be a good candidate as the objective function to be optimized by the algorithm. The reported classification results are competitive to current standards in analyzing microarray gene expression data with a very affordable execution time. This last aspect should not be overlooked, since database size is constantly growing and the complexity of optimization scenarios (that make extensive use of resampling methods) is ever greater.

**VIII. ACKNOWLEDGMENTS**

Authors wish to thank to Spanish CICyT Project no. CGL2004-04702-C02-02; CONACyT and UABC for supporting this research from its beginning.

**REFERENCES**


