Extending an Algorithm for Clustering Gene Expression Time Series

Mikko Korpela and Jaakko Hollmén

Helsinki University of Technology, Laboratory of Computer and
Information Science, P.O. Box 5400, FI-02015 HUT, Finland
mvkorpel@cis.hut.fi

1 Introduction

The development of microarray technology has enabled simultaneous expression measurements from tens of thousands of genes [1]. Many gene expression experiments produce time series data with only a few (around 5) time points, due to the high measurement costs. The time series usually represent the dynamic response of an organism to a change in conditions, e.g. application of some drug or other treatment. Here we share some of the experiences we gained while analyzing such data sets, originating from a collaborative project. More information about that work can be found in [2].

We focus on a particular clustering algorithm designed for short time series data [3]. We found that some inaccuracies in the original presentation of the algorithm need to be addressed. In addition to providing corrections for the problems, we also present an extension to the algorithm.

2 The Clustering Algorithm

The clustering algorithm discussed here was first introduced in [3]. Roughly speaking, the algorithm consists of two phases: selection of model profiles (cluster prototypes) and clustering itself. The clustering phase also includes assessment of the statistical significance of each cluster. After reviewing some benefits and disadvantages of the clustering algorithm as a whole, we briefly introduce the original profile selection algorithm. Then we propose some changes to it.

The clustering algorithm has basically two benefits compared to traditional algorithms like k-means. First, the cluster prototypes are chosen to be distinct, in other words as different as possible. Traditional algorithms might use several similar cluster prototypes to represent typical patterns in the data, and neglect less typical patterns. The second advantage of the clustering algorithm is tied to the first one. Namely, as shown in [3], the statistical test used in the clustering algorithm is able to detect the significance of some small clusters that would go unnoticed with a less sophisticated method.

The statistical significance test used in the clustering algorithm is computationally demanding. Therefore, the algorithm is only suitable for short time series. With today's computer technology, the practical upper limit for the algorithm is probably something less than ten time points.

2.1 The Original Profile Selection Algorithm

The purpose of the profile selection phase is to select m distinct model profiles from a set of candidate profiles P. The set P is constructed by fixing the value at the first time point to zero, and allowing the change between values at consecutive time points to be anything in the range of $-c, \ldots, +c$ discrete units. That is, the change can be at most c units either way, up or down. When the number of time points is n, the set P contains $(2c+1)^{n-1}$ profiles. The set of distinct profiles R is selected with Alg. 1. The algorithm is a greedy approximation to the problem of finding the set R that satisfies

$$\mathop{\arg\max}_{R:R\subset P,|R|=m} \min_{p_1,p_2\in R} d(p_1,p_2) \ . \tag{1}$$

The distance measure used in (1) is

$$d(\boldsymbol{x},\boldsymbol{y}) = 1 - \rho(\boldsymbol{x},\boldsymbol{y}) \ , \tag{2}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors representing model profiles, and $\rho(\boldsymbol{x},\boldsymbol{y})$ is the correlation coefficient (Pearson's correlation) between the vectors. [3]

Algorithm 1 SelectVectorsMaxMinDist A greedy algorithm for choosing m distinct profiles (appeared in [3])

```
SelectVectorsMaxMinDist($d, P, m$)
let $p_1 \in P$ be the profile that always goes down one unit between time points
$R \leftarrow \{p_1\}$
$L \leftarrow P \setminus \{p_1\}$
for $i \leftarrow 2$ to $m$
        do let $p \in L$ be the profile that maximizes $\mathbf{min}_{p_1 \in R} d(p, p_1)$
           $R \leftarrow R \cup \{p\}$
           $L \leftarrow L \setminus \{p\}$
return $R$
```

2.2 Extensions

While implementing the clustering algorithm, some aspects about the profile selection phase aroused our attention. As can be seen from the definition of the correlation coefficient ρ,

$$\rho(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \ , \quad a_i = x_i - \overline{x},\ b_i = y_i - \overline{y} \ , \tag{3}$$

it is not defined when $\boldsymbol{x}$ or $\boldsymbol{y}$ is the constant zero profile, where the value at each time point is zero. Therefore, the zero profile must be removed from the set P

before profile selection, or the profile selection algorithm will fail. The number of profiles remaining in P is $(2c+1)^{n-1}-1$.

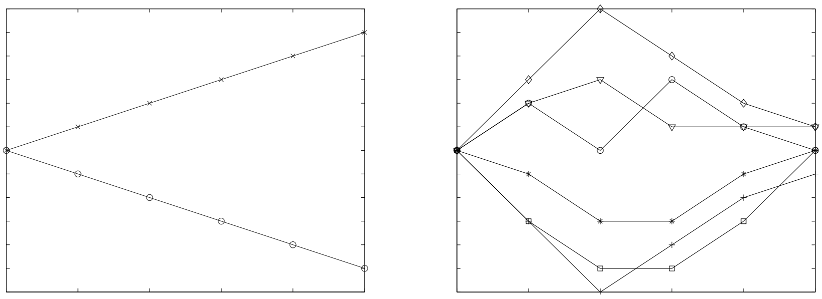
In addition to removing the zero profile, we introduce a procedure to further reduce the amount of profiles in P. The procedure is based on the fact that some profiles are equal with respect to distance measure (2). From a set of equal profiles, only one profile is needed. We choose to keep the "basic" profile and remove its multiples. This is done in Alg. 2. The removal of redundant profiles takes time, but also speeds up profile selection. The procedure also has a cosmetic side: the "simplest" possible profile always represents each equivalence class of profiles.

Algorithm 2 RemoveRedundant A simple algorithm for removing redundant model profiles

```
RemoveRedundant($P, c, n$)
R ← {}
let Primes be the set of all prime numbers up to and including $c$
while $|P| > 0$
    do let $p$ be any profile in $P$
       $P \leftarrow P \setminus \{p\}$
       nonredundant ← TRUE
       let $p_i$, $i \in 1, \ldots, n$, be the values at each time point of $p$
       for each prime in Primes
           do if each $p_i$ is divisible by prime
                then nonredundant ← FALSE
                     break
       if nonredundant
         then $R \leftarrow R \cup \{p\}$
return $R$
```

Algorithm 1 is greedy in the sense that it selects one locally optimal profile at a time. However, this approach fails at times. Figure 1 represents profile selection with parameters $n = 6, c = 3$, when the redundant profiles have been removed with Alg. 2. The two first selected profiles are in Fig. 1a. When choosing the third profile, there are 180 profiles that are equally good optimal choices in the greedy sense. Six of these are shown in Fig. 1b.

Our updated profile selection procedure is a randomized algorithm [4] that simply chooses one of the "equally good" profiles by random. The algorithm (Alg. 3) also has a user-specifiable parameter *repeats*, that adjusts the level of compromise between running time and the quality of the approximative solution to (1). Preliminary experiments with our extended algorithm indicate an improvement in the minimum distance between selected model profiles. With a large number of *repeats* it pays off to reduce the search space by removing the redundant profiles with Alg. 2.



(a) The two first profiles

(b) Six of the equally good options when choosing the third profile

Fig. 1: Ambiguity in greedy profile selection ($n = 6, c = 3$). Algorithm 1 fails when there is no single best choice.

Algorithm 3 SelectVectorsMaxMinDistRandom A randomized greedy algorithm for choosing m distinct profiles

```
SelectVectorsMaxMinDistRandom($d, P, m, repeats$)
$dist_{best} \leftarrow -\infty$
for $i \leftarrow 1$ to $repeats$
        do $R_t \leftarrow$ SelectHelper($d, P, m$)
            $dist_{temp} \leftarrow \mathbf{min}_{(p_1,p_2)\in R_t \times R_t} d(p_1, p_2)$
            if $dist_{temp} > dist_{best}$
               then $dist_{best} \leftarrow dist_{temp}$
                       $R \leftarrow R_t$
return $R$

SelectHelper($d, P, m$)
let $p_1 \in P$ be the profile that always goes down one unit between time points
$R \leftarrow \{p_1\}$
$L \leftarrow P \setminus \{p_1\}$
for $i \leftarrow 2$ to $m$
        do let $p \in L$ randomly be one of the profiles that maximize $\mathbf{min}_{p_1 \in R} d(p, p_1)$
            $R \leftarrow R \cup \{p\}$
            $L \leftarrow L \setminus \{p\}$
return $R$
```

3 Summary

We examined some shortcomings of the profile selection phase of the clustering algorithm introduced in [3]. The removal of the constant zero profile is a mandatory step. The removal of redundant profiles is optional, and has the potential of reducing the time required for profile selection. Finally, Alg. 3 fixes the problem with the original profile selection algorithm and multiple "equally good" profiles. It also includes a possibility of improved results with multiple *repeats*.

Acknowledgements

We would like to thank our collaborators in University of Helsinki and Finnish Institution of Occupational Health for providing us with interesting real-world data.

References

1. Kohane, I.S., Kho, A.T., Butte, A.J.: Microarrays for an Integrative Genomics. The MIT Press, Cambridge, MA, USA (2003)
2. Korpela, M.: Analysis of changes in gene expression time series data. Master's thesis, Helsinki University of Technology, Finland (February 2006)
3. Ernst, J., Nau, G.J., Bar-Joseph, Z.: Clustering short time series gene expression data. Bioinformatics **21**(Suppl. 1) (2005) i159–168
4. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge, UK (1995)