Introduction to Computational Biology
Lecture # 28: More Gene Clustering

Yaacov Weiss
17/6/2007

1 Introduction

In the last lecture, we started discussing how to cluster a microarray. The two approaches were “top-down” and “bottom-up”. In this lecture, we develop a probabilistic model for the microarray data and use it to develop a probabilistic agglomerative (bottom-up) clustering algorithm. Finally we mention an EM algorithm using the same probabilistic model.

2 A Probabilistic Model

2.1 Data

Our input data are the (processed) results of microarray measurements. We get a number for each gene of every array. We will identify these numbers as $X_{ga}$, where $g$ specifies the gene and $a$ the array. Vectors using one dimension will be marked as $\vec{X}_g$ (or $\vec{X}_a$), and the complete matrix will be marked as $\vec{\vec{X}}$.

2.2 Notation

We are trying to cluster the genes (or arrays), thus we need to have a vector with the clustering placements. This vector is of the length of the number of genes, and its type is that of a plain identifier. Obviously, there is never any need to identify a larger number of clusters than genes. We may also limit the number of clusters at different stages.

The cluster vector will be marked as $\vec{C}$, while specific elements will be marked as $C_g$ (or $c_g$) which means the cluster in which $g$ is. When referring to the set of clusters, we will refer to the set $\mathcal{C}$, whose elements are $(c_1, c_2, ... c_n)$. When we refer to the sets of genes and arrays as sets, we will use $\mathcal{G}$ and $\mathcal{A}$ respectively.

2.3 Assumptions

To continue (in the direction we wish to continue) we need to make three assumptions about the distributions.

1. Independence of genes given clustering data.
2. Independence of arrays within a gene.
3. Gaussian distributions: $P(X_{ga} \mid C_g) \sim N(\mu_{c_ga}, \sigma^2_{c_ga})$

Using these assumptions (in order), we find that: $P(\vec{X} \mid \vec{C}) = \prod_g P(\vec{X}_g \mid C_g) = \prod_g \prod_a P(X_{ga} \mid C_g)$

Since we assume Gaussian distributions, given the probability parameters, we get:

$P(X_{ga} \mid C_g, \mu_{c_ga}, \sigma^2_{c_ga}) \propto \frac{1}{\sqrt{2\pi\sigma_{c_ga}}} \exp\left(-\frac{1}{2} \left(\frac{X_{ga}-\mu_{c_ga}}{\sigma_{c_ga}}\right)^2\right)$
3 MLE Analysis

3.1 General Normal Distributions

Assume \( X = \{x_1, x_2, \ldots, x_N\} \) are chosen from random distribution \( N(\mu, \sigma) \) with unknown parameters. It is clear that the maximum likelihood method over this data will return:

\[
\hat{\mu} = \frac{1}{N} \sum_i x_i, \quad \text{and} \quad \hat{\sigma}^2 = \left( \frac{1}{N} \sum_i x_i^2 \right) - \hat{\mu}^2.
\]

Now that we can in theory calculate \( P(\vec{X} | \vec{C}) \), we want to make it practical. We return to the example above.

\[
P(X | \mu, \sigma) = \prod_i P(x_i | \mu, \sigma) \propto \prod_i \frac{1}{\sqrt{2\pi} \sigma} \exp \left( -\frac{1}{2} \left( \frac{x_i - \mu}{\sigma} \right)^2 \right).
\]

It should be noted that the “proportion factor” can be compared between different sets. This is important since we want to be able to compare probabilities of different available clustering options. We still do not know that this is comparable over different sized sets.

To make this tractable, we take log from the above (log is a monotonic function) and play around with it:

\[
l(\mu, \sigma^2) = \ln \prod_i \left[ \frac{1}{\sqrt{2\pi} \sigma} \exp \left( -\frac{1}{2} \left( \frac{x_i - \mu}{\sigma} \right)^2 \right) \right]
\]

\[
= \sum_i \ln \left[ \frac{1}{\sqrt{2\pi} \sigma} \exp \left( -\frac{1}{2} \left( \frac{x_i - \mu}{\sigma} \right)^2 \right) \right]
\]

\[
= -N \cdot \frac{1}{2} \left( \ln 2\pi + \ln \sigma^2 \right) - \frac{1}{2} \sum_i \left( \frac{x_i - \mu}{\sigma} \right)^2
\]

\[
= -\left[ \frac{N}{2} \ln 2\pi + \frac{N}{2} \ln \sigma^2 + \frac{1}{2\sigma^2} \left( \sum_i x_i^2 - 2\mu \sum_i x_i + \mu^2 \right) \right]
\]

Obviously: \( \max l(\mu, \sigma^2) = l(\hat{\mu}, \hat{\sigma}^2) \), so we can continue (with the definitions we found above):

\[
\max l(\mu, \sigma^2) = l(\hat{\mu}, \hat{\sigma}^2)
\]

\[
= -\left[ \frac{N}{2} \ln 2\pi + \frac{N}{2} \ln \hat{\sigma}^2 + \frac{1}{2\hat{\sigma}^2} \left( \sum_i x_i^2 - 2\hat{\mu} \sum_i x_i + \hat{\mu}^2 \right) \right]
\]

\[
= -\left[ \frac{N}{2} \ln 2\pi + \frac{N}{2} \ln \hat{\sigma}^2 + \frac{1}{2\hat{\sigma}^2} \sum_i (x_i - \hat{\mu})^2 \right]
\]

\[
= -\left[ \frac{N}{2} \ln 2\pi + \frac{N}{2} \ln \hat{\sigma}^2 + \frac{1}{2\hat{\sigma}^2} \sum_i x_i^2 - \frac{N}{2} \hat{\mu}^2 \right]
\]

\[
= -\frac{N}{2} \left[ \ln 2\pi + 1 + \ln \hat{\sigma}^2 \right]
\]

**WARNING:** This is not the same result we got in class. This result seems to fit with the rest of the equations, and the next subsection imitates what was done in class with this result.

3.2 Evaluating Cluster Vectors

What we actually want to do, is find a way to compare different cluster vectors. We can apply the above analysis to this problem. Given data \( \vec{X} \) and clustering \( \vec{C} \), assuming normal distributions, we can easily find the most likely normal
variable parameters as above:

\[ \hat{\mu}_{ca} = \frac{1}{N_c} \sum_{g \mid C_g = c} x_{ga}, \quad \text{and} \quad \hat{\sigma}^2_{ca} = \left( \frac{1}{N_c} \sum_{g \mid C_g = c} x^2_{ga} \right) - \hat{\mu}^2_{ca}. \]

We now recall that

\[ P(\hat{X} \mid \hat{C}) = \prod_g P(\hat{X}_g \mid C_g) = \prod_g \prod_a P(X_{ga} \mid C_g). \]

We now rewrite this equation, slightly changing the product ordering.

\[ P(\hat{X} \mid \hat{C}) = \prod_g P(\hat{X}_g \mid C_g) = \prod_g \prod_a \prod_{c \mid C_g = c} P(X_{ga} \mid C_g). \]

Taking log, and noting the inner product is a general normal distribution, we plug in the above results:

\[
\ln \left( \prod_a \prod_c \prod_{g \mid C_g = c} P(X_{ga} \mid C_g) \right) = \sum_a \sum_c \sum_{g \mid C_g = c} \ln (P(X_{ga} \mid C_g)) \\
\propto \sum_a \sum_c \left( -\frac{N_c}{2} \left[ \ln 2\pi + 1 + \ln \hat{\sigma}^2_{ca} \right] \right) \\
= -|A| \cdot |G| \cdot \frac{\ln 2\pi + 1}{2} + \sum_a \sum_c \left( -\frac{N_c}{2} \left[ \ln \hat{\sigma}^2_{ca} \right] \right) \\
= -|A| \cdot |G| \cdot \frac{\ln 2\pi + 1}{2} + \sum_a \sum_{g \mid C_g = c} \left( -\frac{1}{2} \left[ \ln \hat{\sigma}^2_{ca} \right] \right)
\]

of which the left half of the final term is effectively a constant. I'm not sure which of the last two lines is more clear. Either way what we get looks reasonable. In the next section I will refer to the right term from these lines as \( f(\sigma^2) \).

Another problem is that size 1 clusters have infinite scores. This can be solved by setting a minimum variance of some \( \varepsilon > 0 \).

### 3.3 Probabilistic Agglomerative Clustering

We now find the best division into \( k \) clusters. Look at all possible cluster vectors with (up to) \( k \) unique clusters. For each one we calculate its score, and find the best one. This is not tractable, of course. There are approximately \( k^{\mid G \mid} / k! \) such vectors, which are too many to check.

So we use an agglomerative (bottom-up) method. We start with each gene in a cluster by itself. At each step we combine two clusters from the previous step. This gives us a tree which we can cut at any point we choose.

More formally: Let \( \mathcal{C} = \{C_1, C_2, \ldots, C_n\} \) be a set of clusters, that is a set of sets of genes, such that \( \forall i \neq j : C_i \cap C_j = \emptyset \), and \( \bigcup_i C_i = \mathcal{G} \).

A legal step in combining clusters of \( \mathcal{C} \) is of the form \( C^{i,j} = \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_i \cup C_j\} \).

The difference in score between these two sets will be noted as \( \Delta_{\mathcal{C}, C^{i,j}} = \sum_a \left( f(\sigma_{C^{i,j},a}) - f(\sigma_{C_i,a}) - f(\sigma_{C_j,a}) \right) \).

It is pretty clear why we didn’t really need \( N \) as a parameter of \( f \).

The main part of the actual algorithm: loop: find \( \max_{i,j} \Delta_{\mathcal{C}, C^{i,j}} \) merge \( C_i, C_j \) repeat.

This algorithm gives us a tree, from which we choose which clustering to take.

And of course we need initialization: \( \forall g : C_g = \{g\} \).
4 EM

This time we assume a constant number of clusters, with a new parameter $\theta$ defining the probability of a gene to be in a specific cluster. The clusters will behave as hidden states in the EM.

Assuming independence, $P(\vec{C}) = \prod_g P(C_g | \vec{\theta})$

Using the same notation as above:

$$l(\vec{\mu}, \sigma^2, \vec{\theta}) = \ln P(\vec{\bar{X}} | \vec{C}, \vec{\mu}, \sigma^2, \vec{\theta})$$

These can, of course, be calculated directly.

4.1 Sufficient Statistics

The sufficient statistics for this problem are:

$$M_0^c = \sum_g 1(C_g = c)$$
$$M_{ca}^1 = \sum_g X_{ga} \cdot 1(C_g = c)$$
$$M_{ca}^2 = \sum_g X_{ga}^2 \cdot 1(C_g = c)$$

$$E(M_0^c | \vec{\bar{X}}, \vec{C}, \vec{\mu}, \sigma^2, \vec{\theta}) = \sum_g P(C_g = c | \vec{\bar{X}}, \vec{C}, \vec{\mu}, \sigma^2, \vec{\theta})$$

The other statistics are defined similarly. Like before, this quickly simplifies to a direct calculations. Showing that these statistics are sufficient is straight-forward, and is left as an exercise for the reader, (see review lecture).

For the “m” step we just need to remember from the MLE section that:

$$\hat{\mu}_{ca} = \frac{M_{ca}^1}{M_0^c}$$
$$\hat{\sigma}_{ca}^2 = \frac{M_{ca}^2}{M_0^c} - \hat{\mu}_{ca}$$
$$\hat{\theta}_c = \frac{M_0^c}{|G|}$$

Where in the EM case we use the expected values of these statistics.

4.2 Two Dimensional Clustering

In theory, we can repeat the above to clustering both genes and arrays. We change the independence assumptions on arrays to independence of array clusters. In all notation, $D$ will be to arrays what $c$ is to genes. The basic equation will be:

$$P(\vec{\bar{X}}, \vec{\bar{D}}) = \prod_g \prod_{a} P(X_{ga} | C_g, D_a) \sim N(\mu_{C_g, D_a}, \sigma^2_{C_g, D_a})$$

The sufficient statistics are also similar:
\[ M_c^0 = \sum_g 1(C_g = c) \]
\[ M_d^0 = \sum_g 1(D_a = d) \]
\[ M_{cd}^1 = \sum_g \sum_a X_{ga} \cdot 1(C_g = c) \cdot 1(D_a = d) \]
\[ M_{cd}^2 = \sum_g \sum_a X_{ga}^2 \cdot 1(C_g = c) \cdot 1(D_a = d) \]

However, this is not practical. In this case, the probabilities of gene clustering is not independent of the array clustering, making the “E” step inefficient and impractical. The two dimensional clustering problem has to be dealt with differently.