Chapter VIII
Hierarchical Profiling, Scoring, and Applications in Bioinformatics

Li Liao
University of Delaware, USA

ABSTRACT

Recently, clustering and classification methods have seen many applications in bioinformatics. Some are simply straightforward applications of existing techniques, but most have been adapted to cope with peculiar features of the biological data. Many biological data take a form of vectors, whose components correspond to attributes characterizing the biological entities being studied. Comparing these vectors, aka profiles, are a crucial step for most clustering and classification methods. We review the recent developments related to hierarchical profiling where the attributes are not independent, but rather are correlated in a hierarchy. Hierarchical profiling arises in a wide range of bioinformatics problems, including protein homology detection, protein family classification, and metabolic pathway clustering. We discuss in detail several clustering and classification methods where hierarchical correlations are tackled in effective and efficient ways, by incorporation of domain-specific knowledge. Relations to other statistical learning methods and more potential applications are also discussed.

INTRODUCTION

Profiling entities based on a set of attributes and then comparing these entities by their profiles is a common, and often effective, paradigm in machine learning. Given profiles, frequently represented as vectors of binary or real numbers, the comparison amounts to measuring "distance" between a pair of profiles. Effective learning hinges on proper and accurate measure of distances.

In general, given a set A of N attributes, $A = \{a_i | i = 1, \ldots, N\}$, profiling an entity x on A gives

a mapping $p(x) \rightarrow \Re^N$, namely, p(x) is an N vector of real values. Conveniently, we also use x to denote its profile p(x), and x_i the i-*th* component of p(x). If all attributes in *A* can only have two discrete values 0 and 1, then $p(x) \rightarrow \{0,1\}^N$ yields a binary profile. The distance between a pair of profiles x and y is a function: $D(x, y) \rightarrow \Re$. Hamming distance is a straightforward, and also one of the most commonly used, distance measures for binary profiles; it is a simple summation of difference at each individual component:

$$D(x, y) = \Sigma_i^n d(i) \qquad (1)$$

where $d(i) = | x_i - y_i |$. For example, given x = (0, 1, 1, 1, 1) and y = (1, 1, 1, 1, 1), then $D(x, y) = \Sigma_{i=1}^5 d(i) = 1+0+0+0+0 = 1$. A variant definition of d(i), which is also very commonly used, is that d(i) = 1 if $x_i = y_i$ and d(i) = -1 if otherwise. In this variant definition, $D(x, y) = \Sigma_{i=1}^5 d(i) = -1+1+1+1+1 = 3$.

The Euclidean distance, defined as $D = \sqrt{\Sigma_i^n (x_i - y_i)^2}$, has a geometric representation: a profile is mapped to a point in a vector space where each coordinate corresponds to an attribute. Besides using Euclidean metric, in vector space the distance between two profiles is also often measured as dot product of the two corresponding vectors: $x \cdot y = \Sigma_i^n x_i y_i$. Dot product is a key quantity used in Support Vector Machines (Vapnik, 1997, Cristianini & Shawe-Taylor, 2000, Scholkopf & Smola, 2002). Many clustering methods applicable to vectors in Euclidean space can be applied here, such as K-means.

While Hamming distance and Euclidean distance are the commonly adopted measures of profile similarity, both of them imply an underlying assumption that the attributes are independent and contribute equally in describing the profile. Therefore, the distance between two profiles is simply a sum of distance (i.e., difference) between them at each attribute. These measures become inappropriate when the attributes are not equally contributing, or not independent, but rather correlated to one another. As we will see, this is often the case in the real-world biological problems.

Intuitively, nontrivial relations among attributes complicate the comparisons of profiles. An easy and pragmatic remedy is to introduce scores or weighting factors for individual attributes to adjust their apparently different contribution to the Hamming or Euclidean "distance" between profiles. That is, the value of d(i) in equation (1) now depends not only on the values of x_i and y_i, but also on the index i. Often, scoring schemes of this type are also used for situations where attributes are correlated, sometimes in a highly nonlinear way. Different scoring schemes thereby are invented in order to capture the relationships among attributes. Weighting factors in these scoring schemes are either preset *a priori* based on domain knowledge about the attributes, or fixed from the training examples, or determined by a combination of both. To put into a mathematical framework, those scoring based approaches can be viewed as approximating the correlations among attributes, which, without loss of generality, can be represented as a polynomial function. In general, a formula that can capture correlations among attributes as pairs, triples, quadruples, and so forth, may look like the following:

$$D' = \Sigma_i^n d(i) + \Sigma_{i \neq j}^n d(i)c(i,j)d(j) + \Sigma_{i \neq j \neq k}^n d(i)d(j)d(k)c(i,j,k) + \ldots \qquad (2)$$

where the coefficients c(i,j), c(i,j,k), …, are used to represent the correlations. This is much like introducing more neurons and more hidden layers in an artificial neural network approach, or introducing a nonlinear kernel functions in kernel-based methods. Because the exact relations among attributes are not known *a priori*, an open formula like equation (2) is practically useless: as the number of these coefficients grows exponentially with the profile size, solving it would be computationally intractable, and there would not be enough training examples to fit these coefficients.