Chapter 13

Datamining: Discovering Information From Bio-Data

Limsoon Wong

Kent Ridge Digital Labs

21 Heng Mui Keng Terrace

Singapore 119613

Email: limsoon@krdl.org.sg

13.1 Introduction

This chapter is an introduction to what has come to be known as datamining and knowledge discovery in the biomedical context. The major reason that datamining has attracted increasing attention in the biomedical industry in recent years is due to the increased availability of huge amount of biomedical data and the imminent need to turn such data into useful information and knowledge. The knowledge gained can lead to improved drug target, improved diagnostics, and improved treatment plan.

Datamining is the task of discovering patterns from large amounts of potentially noisy data where the data can be kept in regular relational databases or other forms of information repositories such as flat text files commonly used by biologists. It is a very interdisciplinary subject, relying on ideas and developments in database systems, statistics, machine learning, data visualization, neural networks, pattern recognition, signal processing, and so on. More background on datamining is presented in Section 13.2, where we describe the key steps of the knowledge discovery process, the diverse functionalities of datamining, and some popular datamining techniques.

Datamining has many functionalities such as association analysis, classification, prediction, clustering, and trend analysis. The material in this chapter is presented from the classification perspective, where emphasis is placed on basic techniques for uncovering interesting factors that differentiate one class of samples from a second class of samples. Specifically, datamining techniques for the classification of MHC-binding peptides and diabetes clinical study data are described. The two types of data are chosen because they are very different in nature and thus require very different datamining techniques.

The classification of MHC-binding peptides is described in Section 13.3. It is a target discovery problem in computational immunology. It is an illustration of the application of artificial neural network to the classification of noisy homogeneous biomedical data. Our description is based on a collaboration (Brusic and Zeleznikow, 1999) of Kent Ridge Digital Labs and the University of Pittsburgh.

The classification of diabetes clinical study data is described in Section 13.4. It is a problem of forecasting the onset of diabetes. It is an illustration of the application of an idea known as emerging patterns to the classification of heterogenous biomedical data. Our description is based on a collaboration (Dong *et al.*, 1999) of Kent Ridge Digital Labs and the University of Melbourne.

13.2 Datamining Background

Datamining is a natural evolution of information technology along the path of data collection, database creation, database management, and data analysis and interpretation (Han and Kamber, 2000). Here, we briefly explain various aspects of datamining and knowledge discovery in general terms.

13.2.1 Process

The knowledge discovery process can be broken down into six stages (Adriaans and Zantinge, 1996): data selection, cleansing, enrichment, coding, datamining, and reporting.

The first stage of the knowledge discovery process is that of data collection and selection. In the case of the MHC-binding peptide example, it is the collection of information on what peptides are known to bind or not bind which MHC molecules. In the case of the diabetes example, it is the collection of certain clinical information from a select group of diabetic and non-diabetic patients.

The second stage is a data cleansing process to remove noise or irrelevant data. An important element of this process is the de-duplication of data records to produce a non-redundant dataset. For example, the same MHC-binding peptide information may be reported in two separate papers. Another important element of this process is the normalization of data records to deal with the kind of pollution caused by the lack of domain consistency. This type of pollution is particularly damaging because it is hard to trace. For example, MHC-binding peptide information reported in a paper might be wrong due to a variety of experimental factors. In fact, (Schoenbach *et al.*, 2000b) made a detailed study of swine MHC sequences and found that out of the 163 records they examined, there were 36 critical mistakes. Similarly, clinical records from different hospitals may use different terminologies, different measures, capture information in different forms, or use different default values to fill in blanks. As another example, gene expression experiments under similar conditions may produce different data because the overall intensity of a DNA chip can vary substantially from chip to chip.

The third stage, enrichment, is the acquisition of extra information that can be integrated into the existing data. For example, disease demographic data or linkage data in the case of clinical records, or known biological pathway information relating to genes on a DNA chip.

The fourth stage is coding, where data are transformed or consolidated into forms appropriate for datamining. In the case of clinical records, this might be the transformation of the absolute age of a patient into groupings such as "young," "middle age," and "old." In the case of MHC-binding peptides, this might be the transformation of the MHC-binding affinity into groups such as "non-binder," "weak binder," "moderate binder," and "strong binder." Sometimes, this coding step is performed using an automatic discretization algorithm such as the entropy method (Kohavi and Sahami, 1996).

The fifth stage, data mining, is the phase of real discovery. It is an essential process where intelligent methods are applied in order to extract data patterns. We discuss it in greater detail shortly. The last stage is reporting, where visualization and knowledge representation techniques are used to present the mined knowledge to the user, or where a prediction system is produced.

13.2.2 Functionalities

In general, datamining tasks can be split into two categories: descriptive and predictive (Han and Kamber, 2000). Descriptive datamining tasks characterize the general properties of the data. Predictive datamining tasks perform inference on the current data in order to make predictions.

We briefly touch on the main varieties of datamining tasks and functionalities below.

Classification is the process of finding a set of models that describe and distinguish between two or more data classes or concepts. The model is derived by analysing a set of training data that have been explicitly labelled with the classes that they belong to. The model is then used to predict the class of objects whose class label is unknown. In the rest of this chapter, we describe in detail two examples of classification and prediction and also briefly survey a number of other examples of classification and prediction in molecular biology.

Cluster analysis, on the other hand, is used in the situation where the training data do not have any known class labels. The purpose of clustering is to generate class labels for the data. The data objects are typically clustered so that objects within a cluster have high similarity to each other but are very dissimilar to objects in other clusters. Much work (Ben-Dor, 1999; Eisen *et al.*, 1998) on analysing gene expression data belong to this category of datamining tasks.

Outlier analysis deals with objects that do not comply with the general behavior of a model of the data. Most datamining applications discard outliers. However, in some applications the rare events can be more interesting than the more regularly occurring ones. For example, the detection of new particles in nuclear accelerator experiments.

Trend analysis describes and models regularities or trends for objects whose behavior changes over time. The distinctive features of such an analysis include time-series data analysis, periodicity pattern matching, and clustering of time-related data. The inference of gene relationships from large-scale temporal gene expression patterns (D'haeseleer *et al.*, 1998) is an example of this topic.

Association analysis is the discovery of rules showing attribute-value conditions that occur frequently together in a dataset. For example, the co-expression of genes. It is not difficult to develop algorithms for detecting association rules in a large database. The problem, however, is that such an algorithm often returns so many associations that it is difficult to distinguish interesting associations from uninteresting ones. The idea of emerging patterns, to be seen later in the classification of diabetes clinical data, is one method for separating interesting information from uninteresting information.

13.2.3 Techniques

Many techniques have been used in datamining. We now briefly survey some datamining techniques that have been successfully used for classification in the biomedical context.

The most popular classification technique is the idea of decision tree induction, where a dataset is recursively partitioned into discrete subgroups based on the value of an attribute in the dataset. The remaining attributes in the dataset are selected as to whether or not they provide a predictive segregation of the remaining data for different values of the classification variable. The final result is a set of series of splits on values of the attributes, each series of which leads to a classification value. Algorithms for decision tree induction include CART (Breiman *et al.*, 1984), ID3 (Quinlan, 1986), C4.5 (Quinlan, 1992), SLIQ (Mehta *et al.*, 1996), FACT (Loh and Vanichsetakul, 1988), QUEST (Loh and Shih, 1997), PUBLIC (Rastogi and Shim, 1998), CHAID (Kaas, 1980), ID5

(Utgoff, 1988), SPRINT (Shafer *et al.*, 1996), and BOAT (Gehrke *et al.*, 1999). This group of algorithms are most successful for analysis of clinical data and for diagnosis from clinical data. Some examples are diagnosis of central nervous system involvement in hematooncologic patients (Lossos *et al.*, 2000), prediction of post-traumatic acute lung injury (Rainer *et al.*, 1999), identification of acute cardiac ischemia (Selker *et al.*, 1995), prediction of neurobehavioral outcome in head-injury survivors (Temkin *et al.*, 1995), diagnosis of myoinvasion (Longacre *et al.*, 1995), etc.

Another popular classification technique is Bayesian classification. It is based on Bayes theorem $P(H|X) = P(X|H) \times P(H)/P(X)$, which provides a way to derive the posterior probability $P(H|X)$ that a hypothesis H holds given a sample X from the prior probabilities $P(H)$, $P(X)$, and $P(X|H)$. Note that many Bayesian classifiers make the simplifying assumption that the effect of an attribute value of a given class is independent of the values of the other attributes. Nevertheless, these Bayesian classifiers are comparable in performance to decision tree and neural network classifiers in many applications. More information on algorithms for Bayesian classifiers can be obtained in (Baldi and Brunak, 1999; Duda and Hart, 1973; Mitchell, 1997; John, 1997; Heckerman, 1996; Jensen, 1996; Russell *et al.*, 1995; Lauritzen, 1995). Some example applications of Bayesian classifiers in the biomedical context are mapping of locus controlling a genetic trait (Ghosh and Majumder, 2000), screening for macromolecular crystallization (Hennessy *et al.*, 2000), classification of cNMP-binding proteins (McCue *et al.*, 2000), prediction of carboplatin exposure (Huitema *et al.*, 2000), prediction of prostate cancer recurrence (Demsar *et al.*, 1999), prognosis of femoral neck fracture recovery (Kukar *et al.*, 1996), prediction of protein secondary structure (Kasif and Delcher, 1998; Stultz *et al.*, 1997; Arnold *et al.*, 1992), etc.

Related to the Bayesian classifiers are the hidden Markov models or HMMs. A HMM is a stochastic generative model for sequences defined by a finite set S of states, a finite alphabet A of symbols, a transition probability matrix T, and an emission probability matrix E. The system moves from state to state according to T while emitting symbols according to E. In an n-th order HMM, the matrices T and E depend on all n previous states. More detailed introduction to HMMs can be found in (Baldi and Brunak, 1999; Krogh, 1998; Durbin *et al.*, 1998; Eddy, 1996). HMMs have been applied to a variety of problems in sequence analysis, including protein family classification and prediction (Bateman, 1999; Baldi and Chauvin, 1994; Krogh *et al.*, 1994), tRNA detection in genomic sequences (Lowe and Eddy, 1997), Methylation guide snoRNA screening (Lowe and Eddy, 1999), gene finding and gene structure prediction in DNA sequences (Borodovsky *et al.*, 1995; Borodovsky and McIninch, 1993; Baldi *et al.*, 1997; Krogh, 1998; Salzberg *et al.*, 1998), protein secondary structure modeling (Di Francesco *et al.*, 1997), promoter recognition (Yada *et al.*, 1996; Pedersen *et al.*, 1996), etc.

Artificial neural networks are another important approach to classification that have high tolerance to noisy data. A more detailed introduction to them is presented in a later section on the prediction of peptide binding to MHC molecules. Other useful presentation of artificial neural network algorithms can be found in (Rumelhart *et al.*, 1986; Baldi and Brunak, 1999; Chauvin and Rumelhart, 1995). Successful applications of artificial neural networks in the biomedical context include protein secondary structure prediction (Riis and Krogh, 1996; Rost and Sander, 1994; Qian and Sejnowski, 1988), signal peptide prediction (Claros *et al,*, 1997; Nielsen *et al.*, 1997; Emanuelsson *et al.*, 1999), gene finding and gene structure prediction (Uberbacher and Mural, 1991; Snyder and Stormo, 1995), protein translation initiation site recognition (Pedersen and Nielsen, 1997), T-cell epitope prediction (Honeyman *et al.*, 1998), RNA secondary structure prediction (Steeg, 1993), toxicity prediction (Burden and Winkler, 2000), disease diagnosis and outcome prediction (Vriesema *et al.*, 2000; Scott *et al.*, 2000; Turton *et al.*, 2000), etc.

Support vector machines or SVMs are a new approach to the classification problem that has clear connections to statistical learning theory. They differ radically from approaches such as artificial neural networks. In particular, SVM training always finds a global minimum. A SVM is largely characterized by the choice of its kernel function. Thus SVMs connect the problem they are desgined for to a large body of existing research on kernel-based methods. For a detailed discussion of SVMs, see (Raudys, 2000; Vapnik, 1995; Burges, 1998). Some recent applications of SVM in the biomedical context include protein translation initiation site recognition (Zien *et al.*, 1999), protein homology detection (Jaakkola *et al.*, 2000), microarray gene expression data classification (Brown *et al.*, 2000), breast cancer diagnosis (Mangasarian *et al.*, 1995; Friess *et al.*, 1998), etc.

Let us now embark on our datamining examples on the classification of MHC-binding peptides and diabetes clinical data.

13.3 Short Peptides

Short peptides are a special case of protein sequence data. They are often the key to the functional role of protein sequences such as active sites, binding core, etc. Their short length makes it possible to perform certain type of analysis on them that are not computationally feasible on long protein sequences. In this section, we describe a general method called "artificial neural networks" that has worked well on a broad class of classification and prediction problems involving short peptides.

13.3.1 Problem

The immune system employs two means of recognition: soluble antibodies and T-cell receptors. T-cell mediated immunity is the more subtle of the two and is further divided into those associated with class I vs class II MHC molecules. Killer T-cells continually scan the surface of all cells and destroy those with foreign markings which came from short peptides—derived from cytosolic proteins—bound to class I MHC molecules. Helper T-cells scan cell surfaces for short peptides—derived from proteins internalized by endocytosis—bound to class II MHC molecules; association of a foreign peptide to a class II MHC molecule signals that a cell has encountered a pathogen and serves as a call for help (Stryer, 1995). Peptides that induce immune response are called immunogenic peptides or T-cell epitopes. T-cell epitopes are targets for the discovery of vaccine and immunotherapeutic drug components.

We use the work of (Brusic and Zeleznikow, 1999; Brusic *et al.*, 1998b) as an example classification problem in analysing short peptides. Brusic employs an artificial neural network (Patterson, 1996), implemented using the PlaNet package (Miyata, 1991), for the identification of T-cell epitopes from melanoma antigens. That is, given a short peptide derived from a melanoma antigen, his artificial neural network classifies it according to its binding affinity to particular MHC molecules. Brusic's work scans four melanoma-related antigens (MART-1, gp100/pmel17, MAGE-3, and MAGE-6) to identify short peptides that bind HLA-DR4, a class II MHC molecule. Brusic's artificial neural network is trained from a set of 602 HLA-DR4 binding peptides and 713 non-binders drawn from MHCPEP(Brusic *et al.*, 1998a), experimental binding data (Hammer *et al.*, 1994), and FIMM(Schoenbach *et al.*, 2000a). Each training sample has an associated binding affinity that we refer to as the "targeted output" for that training sample. In the rest of this section, we supply the technical details.

13.3.2 Solution

Artificial neural networks are networks of highly interconnected neural computing elements that have the ability to respond to input stimuli and to learn to adapt to the environment. Although the architecture of neural networks differ in several characteristic ways, a typical artificial neural network computing element is a comparator that produces an output when the cumulative effect of the input stimuli exceeds a threshold value.

Figure 13.1 depicts a single computing element. Each input x_i has an associated weight w_i, which acts to either increase or decrease the input signal to the computing element. The computing element behaves as a monotonic function f producing an output $y = f(net)$, where net is the cummulative input stimuli to the neuron. The number net is usually defined as the weighted sum of the inputs:

$$net = \sum_i x_i w_i$$

and the function f is usually defined as a sigmoid:

$$f(net) = \frac{1}{1 + e^{-net}}$$

Such computing elements can be connected in multiple layers into an artificial neural network. Figure 13.2 depicts a fully-connected feed-forward artificial neural network with two layers of computing elements. The output from the two hidden computing elements in the first layer are fed as inputs into the computing element at the second layer. The network is used for classifiction decision as follows. The inputs x_i are fed into the network. Each computing element at the first layer produces its corresponding output, which is fed as input to the computing elements at the next layer. This process continues until an output is produced at the computing element at the final layer.

The artificial neural network used by (Brusic and Zeleznikow, 1999) has just two layers, with two computing elements at the first layer and one computing element at the second layer. It is also fully connected in the sense that each input x_i is connected to both computing elements at the first layer, and both of which are in turn connected to the computing element at the second layer. From previous experience (Brusic *et al.*, 1998b), other architectures for the artificial neural network are possible but are not expected to yield significant differences. A short peptide derived from an antigen is converted into input values x_i according to a sparse coding scheme, where each amino acid in the short peptide is represented as a string of 20 bits. For example, Alanine is represented by "10000000000000000000" and Cysteine is represented by "01000000000000000000". From previous experience (Brusic *et al.*, 1994), other representations are possible but are not expected to produce significantly different results. Since HLA-DR4 is a class II MHC molecule, a short peptide will bind it through a 9-mer core with the rest of the peptide flanking outside the MHC cleft (Rammensee *et al.*, 1995). In other words, we need to consider only 9-mers and thus there are $9 \times 20 = 180$ bits to be used as inputs per peptide. So Brusic's artificial neural network has 180 inputs x_1, ..., x_{180}. An input sample is classified as HLA-DR4 binding if the final output exceeds a threshold, and as non-HLA-DR4 binding otherwise.

The details are in how the weights on the links between the inputs and computing elements are chosen. These weights are learned from training data using the error back-propagation method (Rumelhart *et al.*, 1986). To describe the error back-propagation method, we need to introduce some notations. Let v_{ij} denote the weight on the link between x_i and the jth computing element of the first layer in the artificial neural network. Let w_j denote the weight on the link between the jth computing element of the first layer and the computing element of the last layer. Let z_j denote the output produced by the jth computing element of the first layer. Then the output y produced by the artificial neural network for a given training sample is given by

$$y = f\left(\sum_j w_j f\left(\sum_i x_i v_{ij}\right)\right)$$

This y may differ from the targeted output t for that particular training sample by an error amount Δ. We need a method to reduce this error through an adjustment of the weights v_{ij}

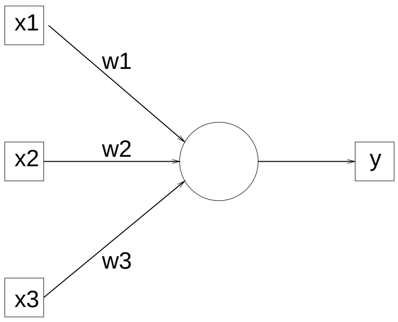

Figure 13.1: A artificial neural network computing element.

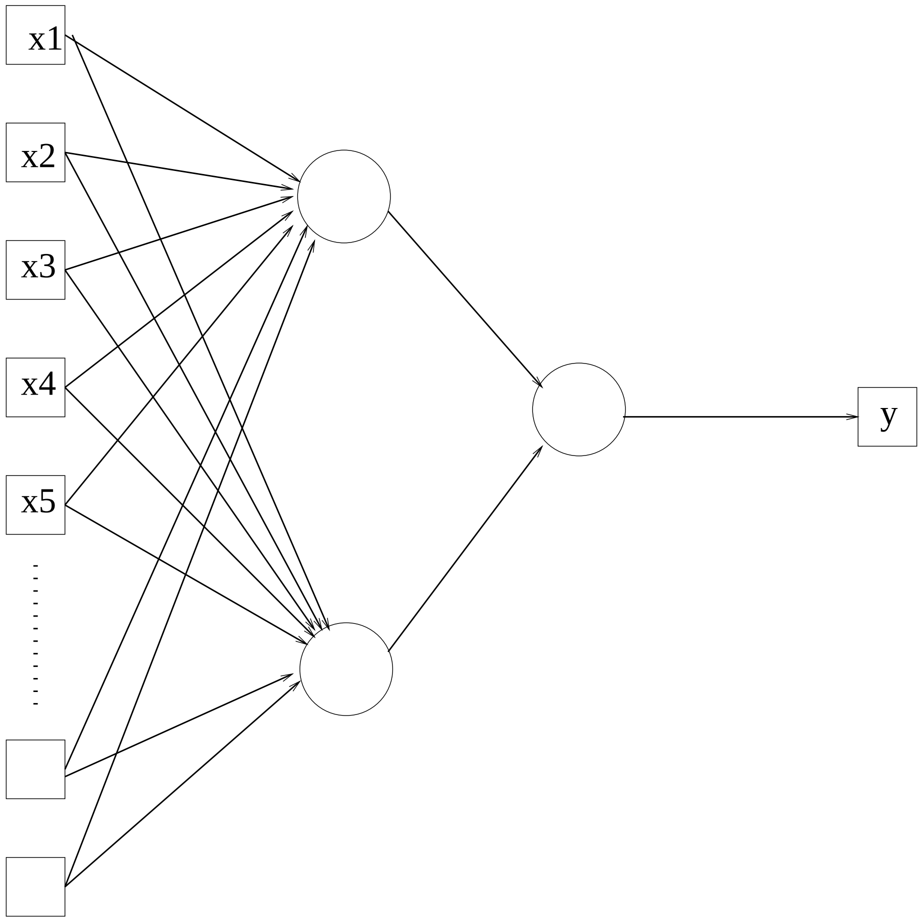

Figure 13.2: A fully-connected feed-forward artificial neural network with two layers of computing elements.

and w_j. This is accomplished by adjusting the weights in proportion to the negative of the error gradient. For mathematical convenient, the squared error E can be defined as

$$E = \frac{(t-y)^2}{2}$$

In finding an expression for the weight adjustment, we must differentiate E with respect to weights v_{ij} and w_j to obtain the error gradients for these weights. Applying the chain rule a couple of times and recalling the definitions of y, z_j, E, and f; we derive

$$\begin{aligned}
\frac{\delta E}{\delta w_j} &= \frac{\delta E}{\delta \sum_j w_j f(\sum_i x_i v_{ij})} \frac{\delta \sum_j w_j f(\sum_i x_i v_{ij})}{\delta w_j} \\
&= \frac{\delta E}{\delta \sum_j w_j f(\sum_i x_i v_{ij})} f\left(\sum_i x_i v_{ij}\right) \\
&= \frac{\delta E}{\delta y} \frac{\delta y}{\delta \sum_j w_j f(\sum_i x_i v_{ij})} f\left(\sum_i x_i v_{ij}\right) \\
&= (t-y) f'(\textstyle\sum_j w_j f\left(\sum_i x_i v_{ij}\right)) f\left(\sum_i x_i v_{ij}\right) \\
&= (t-y) f'(\textstyle\sum_j w_j z_j)(z_j) \\
&= (\Delta)(y)(1-y)(z_j)
\end{aligned}$$

The last step follows because f is a sigmoid and thus $f'(x) = f(x)(1-f(x))$. Then the adjustment Δ_{w_j} to w_j is defined as below, where η is a fixed learning rate.

$$\Delta_{w_j} = -\eta \frac{\delta E}{\delta w_j} = -\eta(\Delta)(y)(1-y)(z_j)$$

However, for the weights v_{ij}, we do not have a targeted output to compute errors. So we have to use the errors Δ_{w_j} as surrogates and apply a similar derivation to obtain

$$\Delta_{v_{ij}} = -\eta(\Delta_{w_j})(z_j)(1-z_j)(x_i)$$

The above derivation provides the adjustments on the weights for one training sample. An "epoch" in the training process is a complete iteration through all training samples. At the end of an epoch, we compute the total error of the epoch as the sum of the squares of the Δ of each sample. If this total error is sufficiently small, the training process terminates. If the number of epoch exceeds some predefined limit, the training process also terminates.

An artificial neural network model for HLA-DR4 binding was built as described above using 602 HLA-DR4 binding peptides and 713 non-binders. The model was then used to process a pool of 1374 peptides from MART-1, gp100/pmel17, MAGE-3, and MAGE-6. Thirty candidate binders identified by the aritificial neural network were synthesized and tested for T-cell responses and DR4 binding. 18 novel T-cell epitopes were confirmed, giving a $18/30 = 60\%$ success rate in predicting T-cell epitopes.

13.3.3 Remarks

A systematic experimental study of a protein antigen for T-cell epitopes would include the following steps. First, a large number of overlapping peptides spaning the length of the protein must be synthesized. Second, each of these peptides must be tested in binding assays. Typically, much less than 5% of peptides emerge from these assays as binders. Third, each of those peptides that bind must be further tested in T-cell assays to confirm immunogenicity. Typically much less than 1% of the original peptides emerge through these assays as immunogenic. Such an exhaustive study would be prohibitively expensive and time consuming.

A highly accurate method of predicting the binding of peptides to MHC molecules enables binding assays to be skipped or significantly reduced (Gulukota, 1998; Brusic and Zeleznikow, 1999).

Given the protein sequence of the antigen, all overlapping peptides spanning the sequence are derived by computer. Each peptide is then classified using the MHC-peptide binding prediction method described earlier. Only those peptides classified by this method as a MHC-binder proceed to be tested in T-cell assays for their immunogenicity. Only for those peptides tested positive in T-cell assays, we perform binding assays. This new procedure would reduce the number of binding assays dramatically.

There are several alternatives to the artificial neural network approach described here. The simplest is the peptide-binding motif approach and many motifs have been proposed for various MHC molecules (Rammensee *et al.*, 1995). These motifs usually indicate primary anchor positions and amino acids acceptable as primary anchors. The more sophisticated motifs usually take the form of a quantitative matrix that specifies the observed frequencies of different amino acids in each position. Their sensitivity and specificity levels are weaker than the artificial neural network approach (Brusic and Zeleznikow, 1999) when there is a large amount of training data available. However, when only a small amount of training data is available, a quantitative matrix may be more useful. Molecular modelling has also been used for the prediction of peptide binding to MHC molecules (Rognan *et al.*, 1994). Nevertheless, the accuracy of molecular models needs to be improved before they can be used for new predictions. Additional methods for predicting MHC-peptide binding are described in several newly published articles (Borras-Cuesta *et al.*, 2000; Raddrizzani and Hammer, 2000; Savoie *et al.*, 1999). Based on their reported sensitivity and precision figures, an artificial neural network-based methodology, as reported by (Brusic *et al.*, 1998b) and described above, appears to be more accurate.

13.4 Clinical Records

The clinical record of a patient is a more general type of data. The characteristics of clinical records are very different from that of protein sequences or gene expression results. The most important difference is perhaps the heterogeneity in the attributes of a clinical record. The meaning of one attribute in a clinical record can be very different from another attribute. For example, a clinical record may have an attribute recording the age of the patient and an attribute recording the blood pressure of the patient. In contrast, in a gene expression record, the attributes correspond to genes and the value of each attribute is the expression level of the corresponding gene, and thus all attributes have the same kind of values. Furthermore, clinical records kept for different studies or by different organizations can differ in the information they capture significantly. Therefore, the general analysis of patient clinical records calls for datamining methods that make less assumption and interpretation on the data. In this section, we describe a general method called "classification by aggregating emerging patterns", or CAEP for short, that has worked well on clinical records and other similar types of data (Dong *et al.*, 1999).

13.4.1 Problem

Classification is an interesting problem in analysing patient clinical records. Suppose we have a population of patients (more generally, samples or instances) that are divided into two groups. For example, we can divide our population of patients with a particular type of cancer into those who responded to a treatment vs those who did not. As another example, we can divide a population of samples into those who showed signs of a disease vs those who did not. The task of a classifier is to discover the factors that differentiate the two groups and to find a way to use these factors to predict which group a new patient should belong.

As our example, we use the PIMA Indians dataset (Smith *et al.*, 1988). The data were collected by the US National Institute of Diabetes and Digestive and Kidney Diseases from a population of 768 women who were at least 21 years old, of Pima Indian heritage, and living near Phoenix in Arizona. They were tested for diabetes according to World Health Organization criteria; that is, the 2-hour post-load plasma glucose was at least 200 mg/dl at any survey examination or during routine medical care. For each patient eight attributes were obtained: (i) number of times pregnant, (ii) plasma glucose concentration, (iii) diastolic blood pressure, (iv) triceps skin fold thickness, (v) 2-Hour serum insulin, (vi) body mass index, (vii) diabetes pedigree function, and (viii) age. Obviously these attributes are heterogeneous; for example, age is in years and diastolic blood pressure in mm Hg. Also, there is a size biase in this dataset in the sense that it contains 29% diabetic instances and 71% non-diabetic instances.

The CAEP method, in this particular example, builds a classifier using the PIMA Indian dataset. Given a new patient record with these eight attributes, the classifier predicts if the patient has diabetes according to World Health Organization criteria.

13.4.2 Solution

The CAEP relies on the recently proposed idea of *emerging patterns* (Dong and Li, 1999). An emerging pattern is a pattern whose frequency increases significantly from one class of data to another class of data. For example, the pattern {odor=none, stalk-surface-below-ring=smooth, ring-number=1} in the description of mushrooms by the Audubon Society (Lincoff, 1981) is a typical emerging patterns. Its frequency increases from 0.2% in the poisonous case to 57.6% in the edible case, at a growth rate of $57.6/0.2 = 288$. Each emerging pattern can have very strong power for differentiating the class membership of some instances. For example, with odds of 99.6%—we show how this is derived later—a mushroom that contains the above pattern is edible. The differentiating power of an emerging pattern is a function of its growth rate and frequency. However, an individual emerging pattern may only be able to classify a small number of instances. Thus it may have poor overall accuracy if it is used by itself on all instances. So to build an accurate classifier, it is necessary to make use of multiple emerging patterns. The CAEP discovers, for each class of data, all the emerging patterns of that class satisfying some threshold on frequency and growth rate. The differentiating power of these emerging patterns are then summed and normalized. The CAEP then chooses the class with the largest normalized score as the winner. We now proceed to describe the CAEP in detail.

We need to discretize of the dataset into a binary one. The value range of each attribute is discretized into a number of intervals using the entropy method (Kohavi and Sahami, 1996). Each $(attribute, interval)$ pair is called an *item* in the binary database. An instance t in the raw dataset is then thought of as a set of items such that an item (A, v) is in t if and only if the value of the attribute A of t is within the interval v. We use the term *itemset* to refer to t under this representation. In an item (A, v) in an itemset, the value v is not allowed to be a null value. As a consequence, missing data that often plague clinical records are conveniently taken care of.

The (binary) PIMA Indian dataset is divided into two sets that we denote D_1 and D_2, corresponding to diabetic and non-diabetic class of instances. We also use the notation D to denote either one of D_1 or D_2 and D' to denote the other set. Let I be the set of all possible items and $X \subseteq I$ be an itemset. The *support* of X in D is defined as

$$supp_D(X) = \frac{|\{t \in D \mid X \subseteq t\}|}{|D|}$$

The *growth rate* of X in D is defined as

$$grow_D(X) = \frac{supp_D(X)}{supp_{D'}(X)}$$

An *emerging pattern* of class D is an itemset which has "large" growth rate in D. Here "large" is an application-dependent threshold chosen by CAEP automatically based on the training dataset.

Each emerging pattern can differentiate the class membership of a fraction of instances that contain that emerging pattern. This differentiating power is derived from the difference between its support in the two sets. Suppose an instance t contains a particular emerging pattern X of class D. What is the likelihood that t belongs to class D? If D and D' are roughly equal in size, this likelihood is

$$likelihood_D(X) = \frac{supp_D(X)}{supp_D(X) + supp_{D'}(X)}$$

Now, using $grow_D(X) = supp_D(X)/supp_{D'}(X)$, we have

$$likelihood_D(X) = \frac{grow_D(X) * supp_{D'}(X)}{grow_D(X) * supp_{D'}(X) + supp_{D'}(X)} = \frac{grow_D(X)}{grow_D(X) + 1}$$

If D and D' differ significantly in size, the supports should be replaced by counts of X in D and D'; and thus

$$likelihood_D(X) = \frac{supp_D(X) * |D|}{supp_D(X) * |D| + supp_{D'}(X) * |D'|} = \frac{grow_D(X) * |D|}{grow_D(X) * |D| + |D'|}$$

Now consider a specific emerging pattern X of class D, say the diabetic class, having a moderate growth rate in D, say 3, and likelihood, say 75%, that appears in a relatively large number of instances in D, say 30%. Then X appears in $30\%/3 = 10\%$ of instances in D'. If we were to use X as our sole clue for class prediction, the achieved sensitivity for class D would be 30% and specificity would be 55%. (Recall that the number of non-diabetic samples is $545 = 71\% * 768$ and diabetic samples is $223 = 29\% * 768$. Furthermore, sensitivity is defined as the ratio of the number of correctly predicted diabetic instances to the number of diabetic instances and specificity is defined as the ratio of the number of correctly predicted diabetic instances to the number of predicted diabetic instances.) Hence the overall accuracy, defined as the percentage of instances correctly classified, would be 72%.

In spite of the relatively high accuracy in using this particular emerging pattern as the sole clue for class prediction, there is a crucial problem. It could only identify 30% of the diabetic patients. This level of sensitivity cannot be considered acceptable since it is much more important to recognize a diabetic person than a non-diabetic one. Furthermore, in real life clinical data, there may be no emerging pattern that has more than $3-5\%$ support. This situation calls for a more innovative use of emerging patterns, where we combine the strength of several emerging patterns to produce a good classifier.

Given a test instance t, we let all emerging patterns of the class D that t contains contribute to the decision of whether t belongs to D. The advantage is that this way more cases can be covered because different emerging patterns can complement each other in their applicable populations. In order to combine emerging patterns, we need a scoring method. Given an instance t and a set E of emerging patterns of a class D, the score of t for D is defined as

$$score(t, D) = \sum_{X \subseteq t, X \in E} likelihood_D(X) * supp_D(X)$$

It is tempting to classify a test instance t as D if $score(t, D) > score(t, D')$, and as D' otherwise. However, such a simple-minded approach is not robust when the numbers of emerging patterns in D and D' differ significantly. This situation is quite common in clinical records where one of the classes have more random distribution of values and consequently fewer emerging patterns. In order be more robust, the score should be normalized by dividing with a score at a fixed percentile of the instances in each class. More specifically, for each class D, a base score $base\_score(D)$ should be found for D and then the normalized score is defined as

$$norm\_score(t, D) = \frac{score(t, D)}{base\_score(D)}$$

Now given a new instance t, we classify it as a member of class D if $norm\_score(t, D) > norm\_score(t, D')$ and as D' otherwise.

It remains to choose $base\_score(D)$. It can be chosen as the median of scores $score(t, D)$ over all $t \in D$. In this case, exactly half of the instances in D have score greater than $base\_score(D)$ and half exactly less than it. In practice, other percentiles between $50 - 85\%$ produce roughly similar results. However, one should avoid percentiles at the extreme ends, say 3%, because clinical records are likely to contain some outliers and choosing such percentiles would give these outliers too much influence.

We compare the performance of the CAEP method described above with two state-of-art methods C4.5 (Quinlan, 1992) and CBA (Liu *et al.*, 1998) under 10-fold cross validation. The accuracy of C4.5 is 71.1%. The accuracy of CBA is 73.0%. The accuracy of CAEP is higher at 75.0%. In fact, its sensitivity and specificity for the diabetic class are 70.5% and 63.3%, and for the non-diabetic class are 77.4% and 83.1%.

It is also reasonable to expect that the emerging patterns used for such highly accurate classification can give insight into important factors of diabetes. Unfortunately, the CAEP method described above produces a very large number of emerging patterns in general. It produces about 2300 emerging patterns in the case of the PIMA Indian dataset. It is too time consuming for a medical expert to analyse such a large number of emerging patterns. This brings us to add a reduction step.

The CAEP method reduces the number of emerging patterns based on the following factors: the strength of emerging patterns, the relationships between emerging patterns, and the difference between their supports and growth rates. The main idea is to prefer strong emerging patterns over their weaker relatives. Let ρ be a new growth rate threshold chosen that is larger than the initial threshold. Let $X_1 >_D X_2$ means X_1, X_2 are both emerging patterns of class D and X_1 is preferrable to X_2; and it is defined as the relation such that

$$X_1 >_D X_2 \quad \text{iff} \quad \text{(a) } X_1 \subseteq X_2 \text{ and } grow_D(X_1) > grow_D(X_2)$$
$$\text{or} \quad \text{(b) } X_1 \subseteq X_2 \text{ and } supp_D(X_1) >> supp_D(X_2) \text{ and } grow_D(X_1) > \rho$$

The motivation for this definition is as follows. If $X_1 >_D X_2$ holds because of (a), then X_1 covers more cases than X_2—as $X_1 \subseteq X_2$—and also has stronger differentiating power—as $grow_D(X_1) > grow_D(X_2)$. The case where $X_1 >_D X_2$ holds because of (b) is more subtle. A typical situation captured by it is when X_2 is an emerging pattern with infinite growth rate but with very small support, whereas X_1 is an emerging pattern with lesser growth rate but has a much larger support, say 30-times more. In such a situation, X_1 is preferred because it covers many more cases than X_2, provided X_1 has a sufficiently high differentiating power—as guaranteed by the condition $grow_D(X_1) > \rho$. To illustrate this point, suppose $supp_D(X_1) = 100\%$, $grow_D(X_1) = 22.25\%$, $supp_D(X_2) = 3\%$, and $grow_D(X_2) = \infty$. Clearly, X_1 is more useful than X_2, since it covers 33 times more instances and its associated likelihood, $22.25/(22.25 + 1) = 95.7\%$, is also very close to that of X_2.

The reduced set of emerging patterns of D is simply any maximal antichain of this relation. That is, for any X_1 and X_2 in the reduced set of emerging patterns, it is the case that neither $X_1 >_D X_2$ nor $X_2 >_D X_1$; and for any emerging pattern X_1 not in this set, it is the case that there is some emerging pattern X_2 in the reduced set such that $X_2 >_D X_1$.

Then *norm_score*, *score*, and *base_score* are all redefined in terms of the reduced set of emerginng patterns. Now, given a new instance, we classify it as before: assign it to the class with the larger *norm_score*. We investigate the performance of the CAEP method with reduction of emerging patterns under 10-fold cross validation. As expected, the number of emerging patterns it produces is about 1600, which is considerably less than the 2300 emerging patterns without reduction. Its accuracy (75.1%), sensitivity for the diabetic class (69.0%), specificity for the diabetic class (64.1%), sensitivity for the non-diabetic class (78.4%), and specificity for the non-diabetic class (82.6%) are comparable to those without reduction of emerging patterns. Thus there is no loss in prediction performance using reduction of emerging patterns.

Let us summarize the steps of the CAEP method. It has two major types of inputs. The first input is the "training" dataset, which is the PIMA Indian dataset set in our example. The second input is the "test" dataset, which contains test instances that we want to classify as diabetic or non-diabetic in this case. The CAEP method goes through two phases. The first phase is called the training phase and consists of discretizing the training dataset; datamining the discretized dataset for emerging patterns; and computing the support, growth rate, and likelihood of each of these emerging patterns with respect to each of the classes; and computing the base score for each of the classes. The second phase is called the prediction phase and consists of computing the score and normalized score of each of the test instances with respect to each class, and making a prediction for each test instance. The training phase is the time-consuming part of the CAEP method and may take an hour or more. However, it needs to be done just once. Its results can be reused for many predictions. Each prediction typically takes much less than a second.

13.4.3 Remarks

Besides the PIMA Indian dataset, we have also tested the CAEP method on a large number of benchmark datasets and the CAEP has very good predictive accuracy on all data sets we have tested (Dong *et al.*, 1999). It gives better accuracy than previous classification algorithms such as C4.5(Quinlan, 1992) and CBA (Liu *et al.*, 1998) in general. The CAEP is highly accurate and is usually equally accurate on all classes even if their proportions are unbalanced. Being equally accurate on all classes is very useful for many applications, where there is a dominant class and a minority class and the main purpose of classification is to accurately predict instances of the minority class. For example, in a preliminary diagnostic situation, one should err on the cautious side and calls for further diagnostic tests for the gray cases.

The CAEP is rather different from previous classifiers because of the novelty of the emerging pattern idea. To arrive at a classification decision, the CAEP uses a set of emerging patterns and each emerging pattern corresponds to a multi-attribute test. Most previous classifiers such as C4.5 (Quinlan, 1992) consider only one test on one attribute at a time. A few exceptions such as CBA (Liu *et al.*, 1998) consider only one multi-attribute test to make a decision. It is also different from boosting (Schapire, 1990) that manipulates training data to generate different classifiers and then aggregate the votes of these classifiers. In the CAEP case, while the emerging patterns are easy to determine, they are too weak to serve as individual classifier.

There is however some similarity between the CAEP and Bayesian prediction. Let an instance t be viewed as a pattern. Bayesian prediction would classify t as belonging to a class C_k, where the probability $P(C_k) \times P(t|C_k)$ is largest among the classes. The optimal Bayesian classifier needs to "know" the probability $P(t|C_k)$ for every possible t, which is rather impractical for high dimensional data sets. We view the score used in the CAEP as a (not very rigorous) "surrogate" for $P(C_k) \times P(t|C_k)$.

The CAEP method is the first application of emerging patterns to classification. There are several variations around the main ideas of the CAEP method. We close this section by mentioning two of these variations: the JEP method (Li *et al.*, 2000b) and the DeEP method (Li *et al.*, 2000a). The JEP method uses exclusively emerging patterns whose supports increases from zero in one class to non-zero in the other class; such emerging patterns are called "jumping" emerging patterns. In the situation where there are many jumping emerging patterns, the JEP method would perform well and the classifier would be more efficient to train. The DeEP method is an instance-based method in the sense that it does not have a single training phase. Instead, for each instance to be predicted, a separate training phase is involved that is automatically "customized" for that instance. So while the DeEP method has to train as many times as there are instances to be predicted, each training phase takes much less time than the CAEP method. However, as each prediction must be preceded by its corresponding training phase, the overall prediction time for the DeEP method is typically an order of magnitude longer than the CAEP method. In terms of accuracy, the DeEP method is usually slightly better than the CAEP method. However, its biggest advantage is that it can incorporate new training data easily. This feature makes it very useful for applications where the training data must be frequently updated.

13.5 Conclusion

In this chapter, we began with a description of datamining in general and then focused on the classification and prediction aspect of datamining. Classification is the process of finding a set of models that describe and distinguish data classes or concepts. The model can then be used for the purpose of predicting the class of newly encountered data. The derived model could be in various forms such as if-then rules, decision trees, mathematical formulae, artificial neural networks, or emerging patterns.

In the MHC-binding peptide example, we encountered artificial neural networks. We gave a brief introduction to feed-forward artificial neural networks and showed how to do back propagation on these neural networks. Using the work of (Brusic and Zeleznikow, 1999), we demonstrated that such neural networks can be trained to differentiate peptides that bind the class II MHC molecule, HLA-DR4, from those that do not bind this molecule. We also briefly mentioned other MHC-binding peptide classification methods (Hammer *et al.*, 1994; Rammensee *et al.*, 1995; Rognan *et al.*, 1994).

In the diabetes example, we encountered emerging patterns. We gave a brief introduction to the concept of emerging patterns and showed how to perform classification by aggregating emerging patterns. Using the work of (Dong *et al.*, 1999), we demonstrated that this method can be used to differentiate diabetic PIMA Indians from non-diabetic ones. We also briefly mentioned other emerging pattern-based datamining methods (Dong and Li, 1999; Li *et al.*, 2000a; Li *et al.*, 2000b).

Besides artificial neural networks and emerging patterns, we also briefly surveyed other techniques such as Bayesian classifiers, hidden Markov models, and support vector machines. All of these techniques have proven useful in applications such as disease diagnosis, gene finding, protein family classification, gene expression data classification, protein secondary structure prediction, and so on. However, we did not go into their details due to space constraint.

All these techniques are general knowledge discovery methods. With appropriate data preparation, they are applicable in a large variety of classification applications. However, in some situations more specialized algorithms can produce better results. For example, the detection of translation initiation sites in RNA sequences (Hannenhalli *et al.*, 1999) and the detection of significant differentially expressed genes in gene expression data (Slonim *et al.*, 2000).

Finally, we should mention that classification may need to be preceded by relevance analysis, which attempts to identify attributes that do not contribute to the classification process. These attributes can then be excluded. In our two examples, we did not use a separate relevance analysis. However, in order to deal with datasets of much higher dimension, a separate relevance analysis step is often crucial. The reason is that datamining algorithms are in general exponential in complexity with respect to the number of attributes present in the datasets. For example, a gene expression record typically contains several thousand attributes, each attribute corresponds to the expression level of a distinct gene. A fast and cheap relevance analysis should be used to reduce these thousands of genes into several tens most relevant genes for subsequent classification analysis (Slonim *et al.*, 2000). There are of course applications involving a large number of dimensions where a separate relevance analysis step is not necessary. For example, (Temkin *et* *al.*, 1995) reported that a methodology like CART (Breiman *et al.*, 1984), which does not require a separate relevance analysis step, is able to produce a classifier of sufficiently high accuracy for predicting neurobehavioral outcome of head-injury survivors. Note also that while relevance analysis is useful in ranking the importance of each dimension of a dataset, it is not always desirable to aggressively eliminate lower ranking dimensions. For example, (Spellman *et al.*, 1998) would not have identified 800 cell cycle genes if they blindly reduced all the yeast genes to just the several tens of the most relevant ones!

Acknowledgements. Section 13.3 on MHC-binding peptides is based on the work by Vladimir Brusic of my lab. Section 13.4 on diabetes clinical data is based on my collaboration with Guozhu Dong, Jinyan Li, and Jenny Zhang. My lab is funded in part by the Singapore National Science and Technology Board. I am also grateful to the useful comments by the referees.

References

P. Adriaans and D. Zantinge. *Data Mining*. Longman, 1996.

G.E. Arnold *et al.* Use of conditional probabilities for determining relationships between amino acid sequence and protein secondary structure. *Proteins*, 12(4):382–399, 1992.

P. Baldi *et al.* Hidden Markov models for human genes: Periodic patterns in exon sequences. In *Theoretical and Computational Methods in Genome Research*, pages 15–32, 1997.

P. Baldi and Y. Chauvin. Hidden Markov models of G-protein-coupled receptor family. *J. Comput. Biol.*, 1:311–335, 1994.

P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach.* MIT Press, 1999.

A. Bateman *et al.* Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucleic Acid. Res.*, 27(1):260–262, 1999.

A. Ben-Dor *et al.* Clustering gene expression patterns. *J. Comput. Biol.*, 6:281–297, 1999.

M. Borodovsky and J. D. McIninch. GENEMARK: Parallel gene recognition for both DNA strands. *Comput. Chem.*, 17(2):123–133, 1993.

M. Borodovsky *et al.* Detection of new genes in a bacterial genome using Markov models for three gene classes. *Nucleic Acid. Res.*, 23:3554–3562, 1995.

F. Borras-Cuesta *et al.* Specific and general HLA-DR binding motifs: Comparison of algorithms. *Human Immunol.*, 61(3):266–278, 2000.

L. Breiman *et al.* *Classification and Regression Trees.* Wadsworth, 1984.

M.P. Brown *et al.* Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97(1):262–267, 2000.

V. Brusic *et al.* Prediction of MHC binding peptides using artificial neural network. In *Complex Systems: Mechanism of Adaptation*, pages 253–260, 1994.

V. Brusic *et al.* MHCPEP, a database of MHC-binding peptides: update 1997. *Nucleic Acid. Res.*, 26:368–371, 1998a.

V. Brusic *et al.* Prediction of MHC class II binding peptides using an evolutionary algorithm and artificial neural network. *Bioinformatics*, 14(2):121–130, 1998b.

V. Brusic and J. Zeleznikow. Computational binding assays of antigenic peptides. *Lett. Pept. Sci.*, 6:313–324, 1999.

F. R. Burden and D. A. Winkler. A quantitative structure-activity relationships model for the acute toxicity of substituted benzenes to *tetrahymena pyriformis* using Bayesian-regularised neural networks. *Chem. Res. Toxicol.*, 13:436–440, 2000.

C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Y. Chauvin and D. Rumelhart. *Backpropagation: Theory, Architectures, and Applications.* Lawrence Erlbaum, 1995.

M.G. Claros *et al.* Prediction of n-terminal protein sorting signals. *Curr. Opin. Struct. Biol.*, 7:394–398, 1997.

J. Demsar *et al.* Naive Bayesian-based nomogram for prediction of prostate cancer recurrence. *Stud. Health Technol. Inform.*, 68:436–441, 1999.

P. D'haeseleer *et al.* Mining the gene expression matrix: Inferring gene relationships from large scale gene expression data. In *Information Processing in Cells and Tissues*, pages 203–212, 1998.

G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*, pages 15–18, 1999.

G. Dong *et al.* CAEP: Classification by aggregating emerging patterns. In *Proc. 2nd Intl. Conf. on Discovery Science*, pages 30–42, 1999.

R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

R. Durbin *et al.* *Biological Sequence Analysis: Probabilistics Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

S.R. Eddy. Hidden Markov models. *Curr. Opin. Struct. Biol.*, 6:361–365, 1996.

M. B. Eisen *et al.* Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868, 1998.

O. Emanuelsson *et al.* ChloroP, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Sci.*, 8(5):978–984, 1999.

V. Di Francesco *et al.* Protein topology recognition from secondary structure sequences—applications of the hidden Markov models to the alpha class proteins. *J. Mol. Biol.*, 267:446–463, 1997.

T. T. Friess *et al.* The kernel adatron algorithm: A fast and simple learning procedure for support vector machines. In *Proc. 15th Intl. Conf. on Machine Learning*, 1998.

J. Gehrke *et al.* BOAT—optimistic decision tree construction. In *Proc. ACM-SIGMOD Intl. Confl. on Management of Data*, pages 169–180, 1999.

S. Ghosh and P.P. Majumder. Mapping a quantitative trait locus via the EM algorithm and Bayesian classification. *Genet. Epidemiol.*, 19(2):97–126, 2000.

K. Gulukota. Skipping a step with neural nets. *Nature Biotechnology*, 16(8):722–723, 1998.

J. Hammer *et al.* Rules governing peptide-class II MHC molecule interactions. *Behring Inst. Mitt.*, 94:124–132, 1994.

J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

S. S. Hannenhalli *et al.* Bacterial start site prediction. *Nucleic Acid. Res.*, 27(17):3577–3582, 1999.

D. Heckerman. Bayesian networks for knowledge discovery. In *Advances in Knowledge Discovery and Data Mining*, pages 273–305, 1996.

D. Hennessy *et al.* Statistical methods for the objective design of screening procedures for macromolecular crystallization. *Acta Crystallogr. D Biol. Crystallogr.*, 56(7):817–827, 2000.

M. C. Honeyman *et al.* Neural network-based prediction of candidate T-cell epitopes. *Nature Biotech.*, 16(10):966–969, 1998.

A.D. Huitema *et al.* Validation of techniques for the prediction of carboplatin exposure: application of Bayesian methods. *Clin. Pharmacol. Ther.*, 67(6):621–630, 2000.

T. Jaakkola *et al.* A discriminative framework for detecting remote protein homologies. *J. Comput. Biol.*, 7((1-2)):95–114, 2000.

F. V. Jensen. *An Introduction to Bayesian Networks*. Springer-Verlag, 1996.

G. H. John. *Enhancements to the Data Mining Process*. PhD thesis, Stanford University, 1997.

G. V. Kaas. An exploratory technique for investigating large quantities of categorical data. *App. Stat.*, 29:119–127, 1980.

S. Kasif and A. L. Delcher. Modeling biological data and structure with probabilistic networks. In *Computational Methods in Molecular Biology*, pages 335–352, 1998.

R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In *Proc. 2nd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 114–119, 1996.

A. Krogh. An introduction to hidden Markov models for biological sequences. In *Computational Methods in Molecular Biology*, pages 45–62, 1998.

A. Krogh *et al.* Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, 235:1501–1531, 1994.

M. Kukar *et al.* Machine learning in prognosis of the femoral neck fracture recovery. *Artif. Intell. Med.*, 8(5):431–451, 1996.

S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Comput. Stat. Data Anal*, 19:191–201, 1995.

J. Li *et al.* DeEPs: Instance-based classification using emerging patterns. In *Proc, 4th Eur. Conf. on Principles and Practice of Knowledge Discovery in Databases*, 2000a. To appear.

J. Li *et al.* Making use of the most expressive jumping emerging patterns for classification. In *Proc. 4th Pacific-Asia Conf. on Knowledge Discovery in Databases*, pages 220–232, 2000b.

G. H. Lincoff. *The Audubon Society Field Guide to North American Mushrooms.* Knopf, 1981.

B. Liu *et al.* Integrating classification and association rule mining. In *Proc. 4th Intl. Confl. on Knowledge Discovery in Databases and Data Mining*, pages 80–86, 1998.

W. Y. Loh and Y. S. Shih. Split selection methods for classification trees. *Statistica Sinica*, 7:815–840, 1997.

W. Y. Loh and N. Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *J. Am. Stat. Assoc.*, 83:715–728, 1988.

T.A. Longacre *et al.* Proposed criteria for the diagnosis of well-differentiated endometrial carcinoma. A diagnostic test for myoinvasion. *Am. J. Surg. Pathol.*, 19(4):371–406, 1995.

I.S. Lossos *et al.* Cerebrospinal fluid lactate dehydrogenase isoenzyme analysis for the diagnosis of central nervous system involvement in hematooncologic patients. *Cancer*, 88(7):1599–1604, 2000.

T. M. Lowe and S. R. Eddy. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acid. Res.*, 25(5):955–964, 1997.

T. M. Lowe and S. R. Eddy. A computational screen for methylation guide snoRNAs in yeast. *Science*, 283:1168–1171, 1999.

O. L. Mangasarian *et al.* Breast cancer diagnosis and prognosis via linear programming. *Oper. Res.*, 43(4):570–577, 1995.

L.A. McCue *et al.* Functional classification of cnmp-binding proteins and nucleotide cyclases with implications for novel regulatory pathways in mycobacterium tuberculosis. *Genome Res.*, 10(2):204–219, 2000.

M. Mehta *et al.* SLIQ: A fast scalable classifier for data mining. In *Proc. Intl. Confl. on Extending Database Technology*, pages 18–32, 1996.

T.M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

Y. Miyata. *A User's Guide to PlaNet Version 5.6.* University of Colorado, 1991.

H. Nielsen *et al.* Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Sci.*, 3:3–14, 1994.

D. Patterson. *Artificial Neural Networks: Theory and Applications.* Simon & Schuster, 1996.

A.G. Pedersen *et al.* Characterization of prokaryotic and eukaryotic promoters using hidden Markov models. *Intell. Syst. Mol. Biol.*, 4:182–191, 1996.

A. G. Pedersen and H. Nielsen. Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. *Intell. Syst. Mol. Biol.*, 5:226–233, 1997.

N. Qian and T.J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, 202:865–884, 1988.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

J. R. Quinlan. *C4.5: Program for Machine Learning.* Morgan Kaufmann, 1992.

L. Raddrizzani and J. Hammer. Epitope scanning using virtual matrix-based algorithms. *Brief. Bioinformat.*, 1(2):179–189, 2000.

T.H. Rainer *et al.* Derivation of a prediction rule for post-traumatic acute lung injury. *Resuscitation*, 42(3):187–196, 1999.

H. G. Rammensee *et al.* MHC ligands and peptide motifs: First listing. *Immunogenet.*, 41(4):178–228, 1995.

R. Rastogi and K. Shim. Public: A decision tree classifier that integrates building and pruning. In *Proc. 24th Intl. Conf. Very Large Data Bases*, pages 404–415, 1998.

S. Raudys. How good are support vector machines? *Neural Network*, 13(1):17–19, 2000.

S.K. Riis and A. Krogh. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *J. Comput. Biol.*, 3:163–183, 1996.

D. Rognan *et al.* Molecular dynamics simulation of MHC-peptide complexes as a tool for predicting potential T cell epitopes. *Biochem.*, 33(38):11476–11485, 1994.

B. Rost and C. Sander. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins*, 19:55–72, 1994.

D. E. Rumelhart *et al.* Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

S. Russell *et al.* Local learning in probabilistic networks with hidden variables. In *Proc. 14th Joint Intl. Conf. Artificial Intelligence, volume 2*, pages 1146–1152, 1995.

S. L. Salzberg *et al.* Microbial gene identification using interpolated Markov models. *Nucleic Acid. Res.*, 26(2):544–548, 1998.

C. J. Savoie *et al.* Use of BONSAI decision trees for the identification of potential MHC Class I peptide epitope motifs. In *Proc. Pacific Symp. Biocomput.*, pages 182–189, 1999.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

C. Schoenbach *et al.* FIMM, a database of functional molecular immunology. *Nucleic Acid. Res.*, 28(1):222–224, 2000a.

C. Schoenbach *et al.* Data warehousing in molecular biology. *Brief. Bioinformat.*, 1:190–198, 2000b.

J.A. Scott *et al.* How well can radiologists using neural network software diagnose pulmonary embolism? *AJR Am. J. Roentgenol.*, 175(2):399–405, 2000.

H.P. Selker *et al.* A comparison of performance of mathematical predictive methods for medical diagnosis: identifying acute cardiac ischemia among emergency department patients. *J. Investig. Med.*, 43(5):468–476, 1995.

J. Shafer *et al.* SPRINT: A scalable parallel classifier for data mining. In *Proc. 22nd Intl. Conf. Very Large Data Bases*, pages 544–555, 1996.

D. K. Slonim *et al.* Class prediction and discovery using gene expression data. In *Proc. 4th Intl. Conf. Computational Molecular Biology*, pages 262–271, 2000.

J. W. Smith *et al.* Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proc. Symp. Computer Applications and Medical Care*, pages 261–265, 1988.

E. E. Snyder and G. D. Stormo. Identification of protein coding regions in genomic DNA. *J. Mol. Biol.*, 248:1–18, 1995.

P.T. Spellman *et al.* Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell.*, 9(12):3273–3297, 1998.

E. W. Steeg. Neural networks, adaptive optimization, and RNA secondary structure prediction. In *Artificial Intelligence and Molecular Biology*, pages 121–160, 1993.

L. Stryer. *Biochemistry.* Freeman, 1995.

C.M. Stultz *et al.* Predicting protein structure with probabilistic models. In *Protein Structural Biology in Biomedical Research*, pages 447–506, 1997.

N.R. Temkin *et al.* Classification and regression trees (CART) for prediction of function at 1 year following head trauma. *J. Neurosurg.*, 85(5):764–71, 1995.

E.P. Turton *et al.* Ruptured abdominal aortic aneurysm: A novel method of outcome prediction using neural network technology. *Eur. J. Vasc. Endovasc. Surg.*, 19(2):184–189, 2000.

E. C. Uberbacher and R. J. Mural. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc. Natl. Acad. Sci. USA*, 88:11261–11265, 1991.

P. E. Utgoff. An incremental ID3. In *Proc. 5th Intl. Conf. Machine Learning*, pages 107–120, 1988.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

J.L. Vriesema *et al.* Neural network-based digitized cell image diagnosis of bladder wash cytology. *Diagn. Cytopathol.*, 23(3):171–179, 2000.

T. Yada *et al.* Extraction of hidden Markov model representations of signal patterns in DNA sequences. In *Proc. Pacific Symp. Biocomput.*, pages 686–696, 1996.

A. Zien *et al.* Engineering support vector machine kernels that recognize translation initiation sites. In *German Conf. Bioinformatics*, 1999. http://www.bioinfo.de/isb/gcb99/talks/zien.