

Chromosome Classification and Speech Recognition
using Inferred Markov Networks
with Empirical Landmarks

By

LAW HON MAN

A THESIS
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF PHILOSOPHY
DIVISION OF COMPUTER SCIENCE
THE CHINESE UNIVERSITY OF HONG KONG
JUNE 1993



11

thesis
QA
274.7
L38
1993



Abstract

A novel method for locating landmarks, which is called the *empirical landmarks*, in inferred Markov networks is presented in this thesis. The method is based on the property of Markov network inference which retains the landmark substrings of the input strings. An empirical landmark is another kind of forced landmark which may be found in all kinds of input strings. Advantages of the empirical landmarks are clear, no *a priori* knowledge on the segmentation of the input strings are needed and the forced landmark speedup can be applied to all kinds of data with empirical landmarks. Statistics are conducted on the banded human chromosome which have shown that the centromere positions can be located as an empirical landmark in most chromosome types. Experiments on chromosome classification have shown that the discrimination power of Markov network inference with empirical landmarks is similar to that of inherited landmarks.

Manipulations of empty states in the inferred Markov networks are the crucial part of the dynamic programming inference. But it is not easy to understand since it is embedded in the string-to-network alignment. Aspects concerning the empty states in modifying the inferred Markov networks are illustrated extensively in the thesis.

In order to find out the power of the inferred Markov networks on speech recognition in both Western and Eastern languages, extensive experiments on recognizing English phonemes and a Chinese dialect (Cantonese) have been carried out. It has been found that, the inferred Markov network is slightly better than the HMM for the English phonemes recognition. For the Cantonese recognition research in this project, despite that it is one of the very few first attempts, the results are encouraging.

Acknowledgement

I would like to express my gratitude to my supervisor, Dr. K. S. Leung for his supervision and advice. Thanks are also due to Dr. Felix Wong who helped me in my first year of study, E. Granum and C. Lundsteen for originating the Copenhagen chromosome database, J. Piper who prepared the density profiles and centromere data of the chromosome database and helped me from obtaining the database, and C. Lundsteen again for the information concerning the European Chromosome Workshops.

Contents

1 Introduction	1
2 Automated Chromosome Classification	4
2.1 Procedures in Chromosome Classification	6
2.2 Sample Preparation	7
2.3 Low Level Processing and Measurement	9
2.4 Feature Extraction	11
2.5 Classification	15
3 Inference of Markov Networks by Dynamic Programming	17
3.1 Markov Networks	18
3.2 String-to-String Correction	19
3.3 String-to-Network Alignment	21
3.4 Forced Landmarks in String-to-Network Alignment	31
4 Landmark Finding in Markov Networks	34
4.1 Landmark Finding without <i>a priori</i> Knowledge	34
4.2 Chromosome Profile Processing	37
4.3 Analysis of Chromosome Networks	39
4.4 Classification Results	45
5 Speech Recognition using Inferred Markov Networks	48
5.1 Linear Predictive Analysis	48
5.2 TIMIT Speech Database	50
5.3 Feature Extraction	51
5.4 Empirical Landmarks in Speech Networks	52
5.5 Classification Results	55

6 Conclusion	57
6.1 Suggested Improvements	57
6.2 Concluding remarks	61
Appendix A	63
Reference	67

Chapter 1

Introduction

Pattern recognition is very important in the field of machine intelligence. Input data can be categorized into different pattern classes based on the characteristic features. The characteristic features used to distinguish between classes are referred to as the interested features and are extracted from a large amount of background details of the given data.

The two main streams in the field of pattern recognition are the statistical and the structural approaches.

Feature extraction is emphasized in statistical approaches and is indeed the key to success. Statistical methods are employed to cluster the interested features as well as to assign the features of unknown classes to those of the known clusters. Features are measured numerically so that their numerical differences can be interpreted.

In structural pattern recognition, structural model is inferred from a finite set of input samples. Each sample is expressed as a sequence of discrete symbols where no numerical difference can be defined. Each symbol represents a set of configurations of the features which are measured in a restricted context. Inference methods are based on the formal language theory and the automata theory. Since samples are expressed in terms of strings, both symbolic information and the sequential properties of symbols are considered in the inference methods.

Chromosome classification is an important but time-consuming task in cytogenetic analysis. Automation of chromosome classification has been studied

since late 50's. With the inventions of different staining techniques, many features can be measured from a single sample. Chromosome recognition is a common example in many textbooks concerning the structural pattern recognition. The outline of a chromosome can be expressed in terms of a set of boundary elements. A structural model called the chromosome grammar can be constructed where the boundary description of the chromosomes can be derived from the grammar. However, different chromosome classes may have similar outline, using chromosome grammar for chromosome classification is impractical.

Significant advances in automation of chromosome classification were observed when G-banded patterns in chromosome were discovered with special stains. Most attempts in automation of chromosome classification have used global statistical properties only for band pattern description. Only few structural approaches rely on the locally dependent details of the band patterns.

Among the structural approaches for automated chromosome classification, inference of Markov networks by dynamic programming is the most successful one. Such an approach can be used for general structural pattern recognition problems with suitable preprocessing of input data. Since the complexity of the computation of a inferred Markov network is proportional to the size of the network and the input string, considerable computations are needed when the size of the inferred Markov network is getting larger. Inference of Markov networks with forced landmarks provides an alternative to the original approach which speed up the computations based on the inherited properties of the input samples.

In this thesis, a method called the empirical landmark finding is investigated which enables the forced landmark speed-up for general types of input samples with inherited properties concerning the forced landmarks. Further computational savings may be obtained by employing more than one forced landmarks. Chromosome classification is selected again for the testing and analysis of the empirical landmarks.

Aspects of the automated chromosome classification are reviewed in Chapter 2. Inference of Markov networks by dynamic programming and the forced landmark speed-up are given in Chapter 3. Chapter 4 discusses the empirical landmark finding with experiments on chromosome classification. Chapter 5 presents a series of experiments concerning the speech recognition using inferred Markov networks with empirical landmarks. Finally, a conclusion on the further development of the inferred Markov network will be presented in Chapter 6.

Chapter 2

Automated Chromosome

Classification

Automation in cytology has gone a long way in the past forty years. As early as 1950's, several systems had been developed for carrying out simple tasks in cytological sample analysis. In fact, these systems were implemented on circuits which ran in video rate. Later, in the 1960's, a revolution in the computer industry led to the decreasing of the hardware cost and the increasing in the degree of sophistication of computers. Relative complicated tasks can be achieved with the help of the general-purposed computers. This led to the development of several highly complex research projects whose performance, at least in some areas of cytology, came close to equaling that of the human technologist.

The major applications of the automation in cytology include the blood cell analysis in Hematology, chromosome analysis in genetic and the cervical smear analysis. Since the preparation of the blood sample is relatively simple, automation in Hematology has been developed very well. Most successfully automatic systems in the early era of this field were hematological analysis systems. Commercial systems are also available which include the automatic sample preparation and analysis. With the increasing of the speed of the general-purposed computers, it is possible to implement an automatic system for practical chromosome analysis and cervical smear analysis.

The characteristics of cytological analysis is the minute size of the objects (cell range from 1 to 100 micrometer in diameter). Hence, the technician should use the microscope to perform the visual examination. In order to simplify the visual degree of individual cell, samples will be stained with various organic chemicals. Interested cells attached with the organic chemicals form the significant chromatic labels for an individual cell such that it can be uniquely identified by the technician.

The size of the sample cells is small with respect to the microscope slides. Normally, a sample will consists of hundreds of cells. Therefore, visual examination of individual cells is a tedious and time-consuming task. In fact, these mechanical tasks are often poorly performed. Hence, the need for automation in cytology is very keen.

Chromosomes are resided in the nucleus of people's nucleated cell which is the carrier of genetic information for the development of an individual. The majority of people have 46 chromosomes in a single cell which can be grouped as 22 pairs of *autosomes* and two *sex chromosomes*¹. Each pair of autosomes contains one chromosome inherited from the father and the other from the mother.

The aim of chromosome analysis is to determine the major chromosome structures of the species. Many chromosome abnormalities caused by environmental agents and radiation can be detected by examination of chromosome structures. It has been proposed that at least one genetic analysis should be carried out for each new born. However, chromosome analysis, as other cytological tasks, is an expensive and time-consuming task. So automation in chromosome analysis is in great demand.

Chromosomes in a species should be paired before it can be analyzed. The process of pairing is called *karyotyping* and the resulted graph which list all the

¹Theoretically, two chromosomes in a pair have identical properties. This may not be true, since two chromosomes are inherited from different individuals.

chromosomes in pairs is called the *karyotype* of a species. Normally, 22 pairs of autosomes in a normal human cell is indexed from 1 to 22 and the index itself is the *class number* or the *type* of that pair². The properties of a specific chromosome class is similar between human cells. Therefore, a standard labeling method is constructed for human chromosome analysis such that the class number of a specific chromosome can be determined. In other words, karyotyping is a classification problem which determines the class index of each chromosome in a species.

Expert knowledge is required in analyzing the karyotype of a species. Such process is the routine jobs for a cytogeneticist and is difficult to be automated. Therefore, most researches in automation of chromosome analysis concentrated in the classification of chromosomes which is tedious for human to carry [26][22].

The main procedures concerning the classification of chromosomes will be introduced in next section. The preview and previous works of each part of the classification procedures will be discussed in the subsequent sections.

2.1 Procedures in Chromosome Classification

Normally, the task of chromosome classification can be divided into several procedures:

- a. Biochemical staining (homogeneous or banded).
- b. Image acquisition.
- c. Segmentation of chromosomes.
- d. Centromere finding.
- e. Features measurement or selection.
- f. Classification.
- g. Rearrangement.

²Two sex chromosome may not appears in pair. A species from a *female* contains 44 autosomes and two identical sex chromosome which is called the class X chromosomes. On the other hand, a species from a *male* have a class X chromosomes, and a sex chromosome which belongs to a new class called class Y instead of 2 class X chromosome.

The above procedures should be carried out in sequence. In fact, each procedure is independent from each other. However, it has been noticed that the intermediate classification result can be consider as a features for further segmentation of the chromosomes. Therefore, we can consider the whole analysis procedure as an iterative procedure [14] (iterate from step c to step f or step c to step g).

2.2 Sample Preparation

For every cytological image processing project, sample preparation is the key to success. This is true when the population of the sample cells is huge and only a small amount of them are of interest. The choice of the staining chemicals will help the technician to select the interested cells from a large population. Normally, the sample cells are extracted from the cultured tissue of human body. The culture process is out of the scope of this paper and will not be discussed here. This section will focus on the choice of staining techniques and its relative changes in measurement features of an image.

There are two general-purposed stains called the *Feulgen* and the *Papanicolaou* stains. With Feulgen stain, the nuclei are stained very dark and are easy to be thresholded. Cytoplasm under Feulgen stain is nearly invisible. On the contrary, cytoplasm is visible under Papanicolaou stain. It is suitable for calculating nucleus-to-cytoplasm ratio in cancerous cell recognition.

For chromosome analysis, chromosomes are extracted from the blood smear of the patient. The cultured blood cells are arrested at the *metaphase*³ stage of the cell development with biochemicals. Once the cultured blood specimen is ready, it is

³Chromosomes in metaphase stage was located altogether as a small black object and such object is called metaphase.

treated with a hypotonic solution (staining chemicals) which enlarges the metaphase cells so that the chromosomes are spread on the microscopic slide very well.

Chromosome sample using classical staining techniques appeared as a set of black connected objects. Hence, for a given sample, only the morphological features can be measured. These features include the shape and size of the individual chromosomes and the centromeric index (the ratio of the two short chromosome arms to the two long ones). With the Caspersson stain, the banding patterns appears in individual chromosomes as shown in Fig 2.1(c). The visual analysis of chromosomes is changed since the banding pattern of individual class of chromosome is unique and standardized.

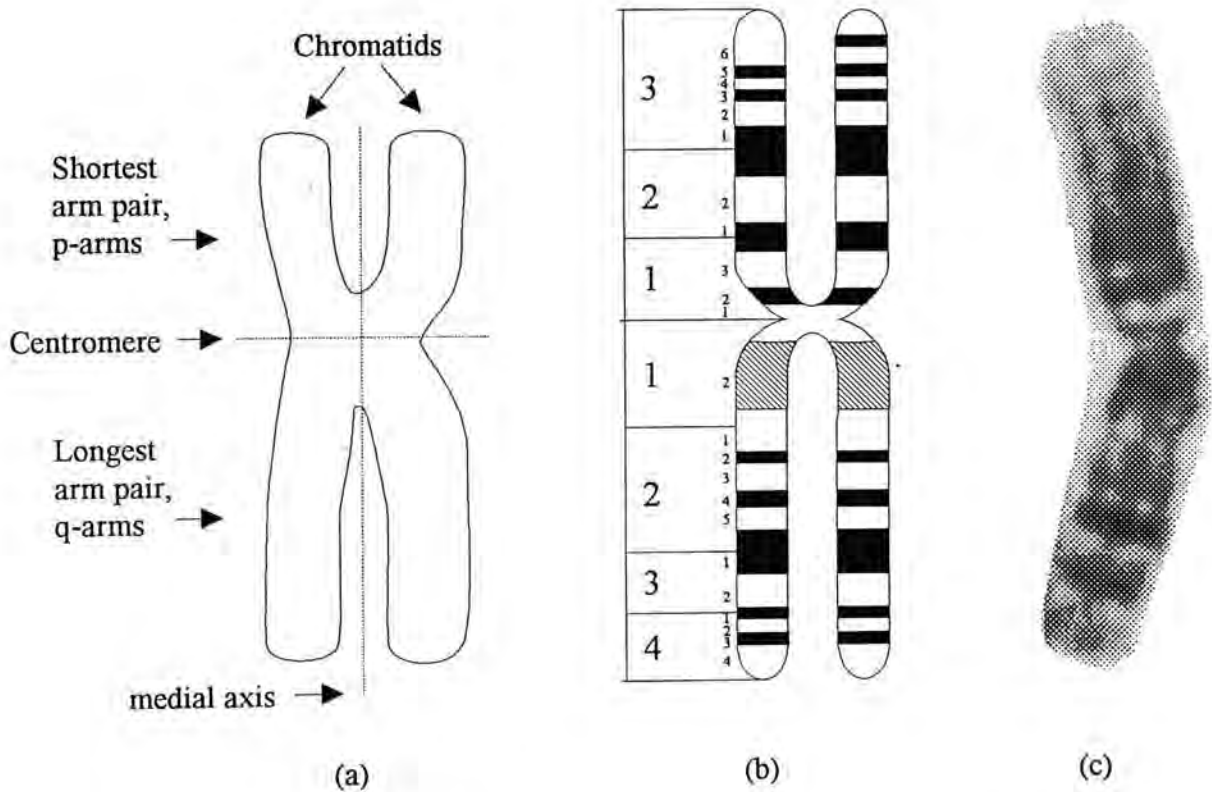


Fig 2.1 (a) The structure of a chromosome extracted in metaphase.
 (b) The reference band model of the chromosome class 2.
 (c) A real sample of chromosome class 1 with band patterns.

2.3 Low Level Processing and Measurement

Since the size of the chromosomes is so small that a high-magnification microscope is needed in order to finish the visual examination. In an automated environment, the images of the stained cell will be captured at magnification $\times 100$. Metaphases in the stained cell will be located by the *Metaphase finders*. The image of metaphases will be captured at magnification $\times 1000$ for further processing. Most preliminary features of a chromosome such as area, shape, axis as well as the centromeric index can be measure in the 2-D image without further projection into profile domain.

2.3.1 Segmentation of chromosomes

For poorly prepared samples, some chromosomes will touch or overlap with another chromosomes. In such cases, these chromosomes cannot be separated very well by simple thresholds or boundary tracing. If two chromosomes are heavily overlapped, some banding patterns will be covered. In order to maintain the speed of the whole system, the overlapping cases may be left for human handling.

Most segmentation techniques in image processing domain such as heuristic search and region growing can be used for separating touched chromosomes. Also, some specifically designed techniques have been proposed for splitting chromosomes. Vanderheydt et al [37] have proposed a method which applies the generalized fuzzy binary relation to assist the decision making in decomposition.

Apart from the structural method as mentioned above, Ji et al [13] has shown that by carefully selecting the shape of the structuring elements, slightly touched chromosomes can be split by erosion which is a local transformation in mathematical morphology. Since the transformation is independent of the size of objects, with interval coding of binary images [28], it is considered to be an efficient method that can be implemented on a serial computer.

As summarized by L. Ji [14], the chromosome segmentation problem can be solved in most cases by a procedure which based on the concavity analysis in relation to expected chromosome shape and a heuristic search for the minimum density path. In fact, a 95% success rate can be achieved by an algorithm based on the above assumption as reported in [14].

2.3.2 Centromere finding

The determination of the centromere position is an important step in chromosome analysis. Centromere is the reference point of a chromosome. By locating the centromere position, we can determine the proper orientation of the chromosome. Centromere position itself is an important feature in chromosome classification. For old staining techniques, such as homogeneous stain, centromere position can be found by locating the point with minimum width. Structural methods such as locating the maximum concavity in the chromosome contour have been introduced. While another method approximate the chromosome with a convex hull of the boundary and calculate the centromere position with different morphological features [26].

Groen et al [12] have evaluated two new methods for determining the centromere position. The first method aims at searching the closest pair of opposite contour points. Since the search will be done exhaustively along the opposite pair of contours of a clipped chromosome, the complexity and time requirement are critical. The second method based on the profile of the width of the chromosome, defined as the distance between the borders measured perpendicular to the main axis. The method search for the relative minimum between two maxima of a smoothed profile. The original profile will be fitted with a second order polynomial to find the precise position of the centromere if the relative minimum exists.

2.4 Feature Extraction

The goal of feature extraction is to find a transformation from an n -dimensional observation space X to a smaller m -dimensional feature space Y that retains most of the information needed for pattern classification [33]. The computational complexity for pattern classification is reduced by dealing with the data in a lower dimensional space. On the other hand, generalization can be obtained from a given number of training samples such that a more reliable decision rule can be formulated.

Features in chromosome analysis can be classified into four levels according to the *robustness* (how much *a priori* information is needed before it can be measured) of features [9][27]. *Level 1* features can be obtained immediately after thresholding, when the chromosome outline is known. Features like *area*, *average* and *integrated density*, other *density histogram* features and *contour measure* can be calculated. *Level 2* features include knowledge of object orientation and medial axis which allows the measure of *length* and *width*, and the derivation of the profile. *Level 3* features require both the axis and profile and the knowledge of chromosome polarity which is necessary for making use of the centromere position. *Level 4* features require the axis, the polarity and the centromere position. An example of level 4 features is the centromeric index of a chromosome which requires different lower levels of features to determine.

A good feature extraction method is also a key to the success of the whole classification algorithm. Most feature extraction part cannot be isolated or reused from the whole classification algorithm. For the chromosomes band patterns, the original 2-D image will be transformed into a 1-D profile along the medial axis of one of the chromatids. Such profile exhibits most characteristics of the original bands and becomes the standard feature of the most modern chromosome classifier.

The research of automation in chromosome analysis has been developed rapidly since the discovery of the banding pattern. Up to now, the successful

classification rate is around 70-80%. One reason for failing to fully automate the task is that chromosomes are not as 'stable' objects as human expected [12]. Although, the banding patterns of chromosome classes are known, the appearances of these patterns are not clear in real life cases. It has been reported that, using band staining techniques, not more than 58% of all bands supposedly present are found in reality [21]. Most recent approaches combine both banding patterns and morphological characteristics for classification.

Band pattern description

The chromosome profile along the medial axis is a projection of the chromosome which can be used for the description of band patterns. A set of simplified parameters can be generated from this profile. This set of parameters can be used to measure the similarity between a profile and a set of templates representing the ideal chromosomes. The method used to describe the band patterns can be subdivided into *global* and *local* methods according to Piper's classification [26].

2.4.1 Global band descriptor

With global band descriptors the number of features is fixed in advance. Therefore, the classification scheme based on the global band descriptor is relatively simple because of its lower degrees of freedom. The problem with the global band descriptor is that the small abnormality may be indistinguishable from random noise and the position of the abnormality is not known even it is detectable.

Although the classification schemes from the global band descriptor are relatively simple, the better results in chromosome classification are based on this type of descriptors. The human chromosomes, represented by their density profiles, are described by a set of distribution function called the *Weighted Density Distributions* (WDDs) [5][18] by application of a number of sawtooth-like weighting

functions. The chromosome profile is correlated with these functions to produce the global features.

2.4.2 Local band descriptor

The local descriptor aims at isolating the individual bands in a density profile, and describes them by size and position. Since this is a two stages process, segmentation and measurement, the success of the operation may be evaluated at different stages. Also, the analysis of the bands may be named individually. Thus, band description may be expressed in a format which is suitable for visual examination and comparable to the one used by a cytogeneticist. Such an approach is capable of identifying small anomalies such as extra or missing bands explicitly. Four different local descriptors are reviewed in the following paragraphs.

2.4.2.1 Gaussian decomposition of banding profile

Most early local descriptors aim at the decomposition of the density profile of the chromosome into a sum of Gaussian distribution, whose mean, standard deviation and peak value effectively describe a band in the profile [8]. The descriptors are generated by an iterative procedure which, in each turn, with reference to the first peak remains in the profile, applied a different Gaussian curve with a peak height, width and position. In each iteration, the selected Gaussian function will be subtracted from the profile such that the first peak remains will be eliminated. Such iterative procedure will be repeated until some prescribed condition is satisfied.

2.4.2.2 Encoding of profiles

Lundsteen [19] introduced a method which encoded each peak in a nonlinear filtered profile into a simple sequence of band transitions (BT-sequence). The code represents three attributes of the peak:

- a. Density of the peak.
- b. Density difference between the peak and the neighboring valley.
- c. Position of the peak.

The BT-profile retrieved from the BT-sequence representation can also be used for visual classification [20].

On the other hand, the BT-profile is based on a subset of the information of the idealized profile. The idealized profile can be used to approximate the original profile with fewer density levels. Granum et al [7] have developed an inferred Markov network model for chromosome classification which based on the "difference-string" (the encoding of the idealized profile based on the transition between incremental bands).

2.4.2.3 Tree structure "split and merge" description

As described by Rosenfeld [31], the gray-level-dependent properties of image subsets can all be naturally extended to fuzzy subsets m by simply weighting each pixel by its degree of membership in m . Region splitting and merging techniques are often used to improve a given segmentation of an 2-D image. The 1-D analogy of the "split and merge" procedure can be applied to the integrated density profile based on the line pattern obtained from different thresholds [38][39]. At each increment of the threshold level, connected segment of dark line within the same level form a node of the tree, linked to the connected segment at the previous threshold which contains them. Following the split procedure, non-branching node sequences are merged into single nodes.

2.4.2.4 Laplace local band descriptor

While other descriptors concentrated on the chromosome profile along the medial axis, Groen [12] developed a Laplace descriptor based on 2-D Laplace filter which operate on individual isolated chromosome images directly. Such second derivative

filter leads to the detection of peaks and valleys (convex and concave regions) in gray scaled images. Normally, bands in the image form the concave regions such that labeling is possible.

A set of band parameters for each band such as the area, darkness, and the minimum, maximum and middle position will be measured. The features generated by the descriptor is a set of locations of different bands such as the location of the band with the largest area, the darkest band and the first band after the centromere.

2.5 Classification

Classifiers in chromosome classification, as other classification problems, can be categorized into structural approaches and statistical approaches. Although, chromosome grammar is a common example in structural pattern recognition, there is a small number of chromosome classifiers using structural approaches [20][34]. Inference of Markov networks by dynamic programming is the best classifier among all structural approaches [7]. This approach constructs a Markov network from a given set of samples in string form using string-to-network alignment which is a dynamic programming computation. The string-to-network alignment can be accelerated when forced landmark is considered [10]. This accelerating method will be generalized in this thesis such that sample data with no *a priori* information on the landmark position can be considered.

With the statistical approaches, discrimination functions are employed after features have been measured. For *within-cell* chromosome classification, there are three main classification approaches as follows.

1. Simple *context-free classifiers* using such methods as linear or quadratic discriminant functions [25][30], distance functions to pre-learned classes, or a Bayesian approach.

2. *Fuzzy subset theory classifier* [39] evaluates an unknown chromosome according to the model of a particular chromosome. With the tree structure description obtained from the "split and merge" procedure and the global features such as the length, a model can be built to aggregate the goodness of fit of measured features to the learned features of the class and the complements of the overall goodness of fit of the unknown chromosome to neighboring classes. The aggregation function is a generalization of a weighted mean of the features.

3. *Context sensitive (rearrangement) classifiers* exploit the assumption that most cells are normal. For example, given an initial guess (which normally be obtained from other contex-free classifier) of a karyogram from the classification procedure, we can rearrange the chromosomes in each group by shifting chromosomes from a group with too many chromosomes to one with too few chromosomes such that the class size is two (the normal class size). Most contex sensitive classifiers use maximum likelihood methods for the preliminary classification. For each shifting of a chromosome from a class to another class, an additional likelihood will be accumulated to the total likelihood reflecting the similarity of the chromosome to the new class and the dissimilarity to the old class.

Piper [29] has reviewed four rearrangement classifier which resulted in small improvement to the accuracy of chromosome classification. Tso et al [35] has proposed a transportation algorithm which rearrange chromosomes in the 10 Denver groups based on the maximum likelihood approach. The algorithm is extended later [36] for rearranging chromosomes in the 24 groups.

Chapter 3

Inference of Markov Networks by Dynamic Programming

In this chapter, a data-driven inference method based on dynamic programming (DP) will be reviewed. As proposed by Thomason and Granum [34], such method constructs a structural model called the *Markov Networks* from a finite set of sample strings. The DP inference method can be applied to different classification problems with suitable feature extraction and encoding method which encode testing samples into symbol strings. With the profile processing method described in next chapter, the DP inference can be used as a chromosome classifier which produced the superior results as claimed in [7][6].

The definition of the Markov networks, as a constrained Markov chain, will be given in Section 3.1. Section 3.2 describes the DP solution to a string-to-string correction problem where the concept of the *minimal cost editing* is introduced. Section 3.3 describes the DP computations for string-to-network alignments. Section 3.4 introduces the forced landmark concept as a speedup in the DP for string-to-network alignments.

3.1 Markov Networks

The string-to-network alignment described in Section 3.3 creates a first-order, finite-state Markov chain from a finite set O of sample strings. A finite Markov-Chain is defined as [24]:

Let $\{X_n\}$ be a sequence of random variables taking the values $i \in I$. $\{X_n\}$ is said to be a first-order Markov chain or Markov-dependent, if for all $i_0, \dots, i_{n+1} \in I$, and $\forall n$

$$\text{Prob}[X_{n+1} = i_{n+1} | X_0 = i_0, \dots, X_n = i_n] = \text{P}[X_{n+1} = i_{n+1} | X_n = i_n]$$

If I is a finite set of integers, $\{X_n\}$ is said to be a finite Markov chain.

Normally, an arbitrary first-order Markov chain is characterized by five items [34]:

1. the set of states, $S = \{s_0, s_1, \dots, s_n, s_f\}$, also called nodes;
2. the set of outputs assignable to states, $V = \{v_0, v_1, \dots, v_T\}$;
3. the initial state distribution $(p_0^0, p_1^0, \dots, p_f^0)$;
4. the state transition matrix P with entries $p_{ij} = \text{prob}(s_j^{t+1} | s_i^t)$;
5. the output matrix B with entries $b_{ik} = \text{prob}(s_k^t | s_i^t)$;

The inference method described in Section 3.3 constructs a constrained Markov chain called *Markov networks*. A Markov network M is a first-order, finite-state Markov chain for which:

1. the initial state distribution is $(1, 0, \dots, 0)$, i.e. there is a unique starting state, s_0 ;
2. $p_{ii} = 0$ for $i <> f$, i.e. no cycle can be created in M ;
3. $p_{ff} = 1$, i.e. there is a single absorbing state s_f ;

4. the states are ordered such that $i < j$ implies $p_{ji} = 0$, i.e., a realization of the process moves "left-to-right" from s_0 without loops or cycles until absorption in s_f ;
5. each state deterministically outputs one specific symbol, but different states may output the same symbol.

For each node in the network, the transition probabilities to other states can be measured by the relative frequencies. For the first sample string in O , a single-path network will be created. Subsequent sample strings will be installed into the network through the string-to-network alignment which maximizes the sample's probability as a network realization. At the same time, common substrings in the sample strings can be retained. Network modifications will be applied for each installation of sample string according to the DP computation in string-to-network alignment. In order to achieve the alignment, empty states will be introduced which generates the empty string e (null string).

3.2 String-to-String Correction

Dynamic programming technique has been used for finding the minimal cost editing sequence between the landmark string and the input string [40][3]. For a landmark character sequence $A = a_1a_2\dots a_m$ and text $B = b_1b_2\dots b_n$, a *k*-approximate match is a match of A in B that has at most k differences. The differences may be any of the following three types:

1. The corresponding characters in A and B are different;
2. A is missing a character that appears in B ;
3. B is missing a character that appears in A .

The above three differences are corresponding to three edit operations [40]. An edit operation is a pair $(a, b) \neq (\Lambda, \Lambda)$ of strings of length less than or equal to 1

and is usually written as $a \rightarrow b$ where Λ is defined as null string. A is transformed to B , written as $A \Rightarrow B$, via $a \rightarrow b$ if $A = \sigma a \tau$ and $B = \sigma b \tau$ for some string σ and τ . String B results from the applications of at most k edit operations to string A . The above differences can be viewed as:

1. *Change operation* : $a \neq \Lambda$ and $b \neq \Lambda$;
2. *Insert operation* : $b \neq \Lambda$;
3. *Delete operation* : $a \neq \Lambda$.

The k -approximate match can be computed by using the dynamic programming technique. Each element $D_{i,j}$ in the cost matrix D represents the minimum number of differences between $a_1 \dots a_i$ and a segment of B ending at b_j . $D_{i,j}$ is the minimum of the following three values:

1. if $a_i = b_j$ then $D_{i-1,j-1}$ (*match*) else $D_{i-1,j-1} + 1$ (*substitute*);
2. $D_{i-1,j} + 1$ (*insert*)
3. $D_{i,j-1} + 1$ (*delete*)

The rows in D are the elements in A and the columns are the elements in B . Thus $D_{m,n}$ is the minimum number of edit operations required in order to change A into B . The actual edit operations can be traced in the backward direction for $D_{m,n}$ to $D_{0,0}$. The complexity of the whole computation is $O(mn)$ for two string with lengths m and n .

Fig 3.1 illustrates an example of string matching. The computation aims at finding a 1-approximate match between the landmark string $A = \text{"happy"}$ and the checking sentence $B = \text{"Have a hsppy"}$. The approximated string can be found by backtracking the column with a value of 1 in the fifth row, i.e., the matched string is "hsppy" in this example.

		H	a	v	e	a	h	s	p	p	y
0	0	0	0	0	0	0	0	0	0	0	0
h	1	1	1	1	1	1	1	0	1	1	1
a	2	2	1	2	2	2	1	2	1	2	2
p	3	3	2	2	3	3	2	2	2	1	2
p	4	4	3	3	3	4	3	3	3	2	1
y	5	5	4	4	4	4	4	4	4	3	2

Fig 3.1 Cost matrix for aligning the string "Have a hsppy"
with the landmark "happy"

Similar techniques has been employed into various fields concerning the recognition of landmark patterns [16].

3.3 String-to-Network Alignment

For a given network M and a sample string, the task of the string-to-network alignment is to modify the network to include the sample string explicitly. String-to-network alignment is a dynamic programming computation which finds the minimum total cost of the network modification to have the sample string installed. The total cost is the logarithm of the probability of the generation of the string.

The dynamic programming matrix for aligning a string $O = z_1 z_2 \cdots z_j \cdots z_m$ into a network M is shown in Fig 3.3. In actual computation, a null character $z_0 = \Lambda$ is attached to the beginning of O in order to maintain the homogeneity of the computation. As in the string-to-string correction problem, rows in the matrix are the elements in the landmark string. However, in this case, the landmark string is the spread of the network M . A *spread* of M is a string in which each element is a character generated from a state in M . Therefore, each character corresponds to a state in M . A spread of a network can be obtained by applying a topological sort over this network. The sequence generated from the topological sort represents the dependence of the states in the network.

Since relative frequency estimates of transition probabilities, $p_{i,k}$ are used, each network arc is also labeled with its frequency of use on sample string alignments, denoted by $f_{i,k}$ for an arc from s_i to s_k , so that $p_{i,k} = f_{i,k}/f_i$ where

$$f_i = \sum_k f_{ik} \quad (3.1)$$

For the dynamic programming matrix D , antilog D_{ij} is the maximum probability with which a modified network can generate substring $z_1 \cdots z_j$ by an alignment for which the neighborhood of s_i is the point reached in the network so far. For any node s_i in the network, modifications allowed in the neighborhood of s_i are shown in Fig 3.2. The optimal value for D_{ij} is the maximum values of $D_{i-1,j}$, $D_{i-1,j-1}$, and $D_{i,j-1}$ (Eq(3.2 - 3.4) respectively) such that substring $z_1 \cdots z_j$ is generated by one of the following possible modifications with maximum probability, i.e., D_{ij} is the maximum of :

$$1. D_{i-1,j} + \log \left[\frac{f_{i-1,j} + 1}{f_{i-1} + 1} \right] + \log \left[\frac{1}{f_i + 1} \right] \quad (3.2)$$

$$2. D_{i-1,j-1} + \log \left[\frac{f_{i-1,j} + 1}{f_{i-1} + 1} \right] + \begin{cases} 0 & \text{if } z_j = s_i \\ \log \left[\frac{1}{f_i + 1} \right] & \text{if } z_j \neq s_i \end{cases} \quad (3.3)$$

$$3. D_{i,j-1} + \log \left[\frac{1}{f_i + 1} \right] \quad (3.4)$$

which correspond to the deletion, match or substitution, and insertion cases in Fig 3.2, respectively.

$D_{f,m}$ is the last element of the cost matrix D which is the logarithm of the maximum probability with which a modified network can generate the sample string.

The initial values in the cost matrix D can be filled in the following manner [40]:

1. $D_{0,0} = 0$
2. $D_{i,0} = \sum_{r=1}^i (\text{cost of delete the state } s_i)$
3. $D_{0,j} = \sum_{r=1}^j (\text{cost of insert } z_j \text{ after the initial state } s_0)$.

In order to find out the optimal trace (the editing sequence) after the cost matrix has been filled, a path matrix has been created and updated in parallel with the cost matrix. Each entry in the path matrix contains the choice in calculating the corresponding entry in the cost matrix according to the minimum cost criteria. The optimal modification can be obtained by tracing the path matrix in backward direction from the lower-right entry to the upper-left entry.

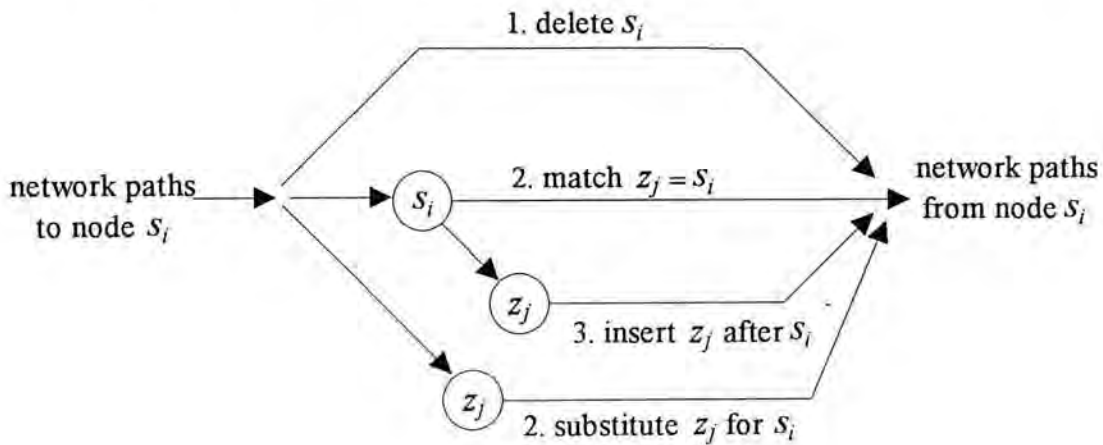


Fig 3.2 Neighborhood of node s_i

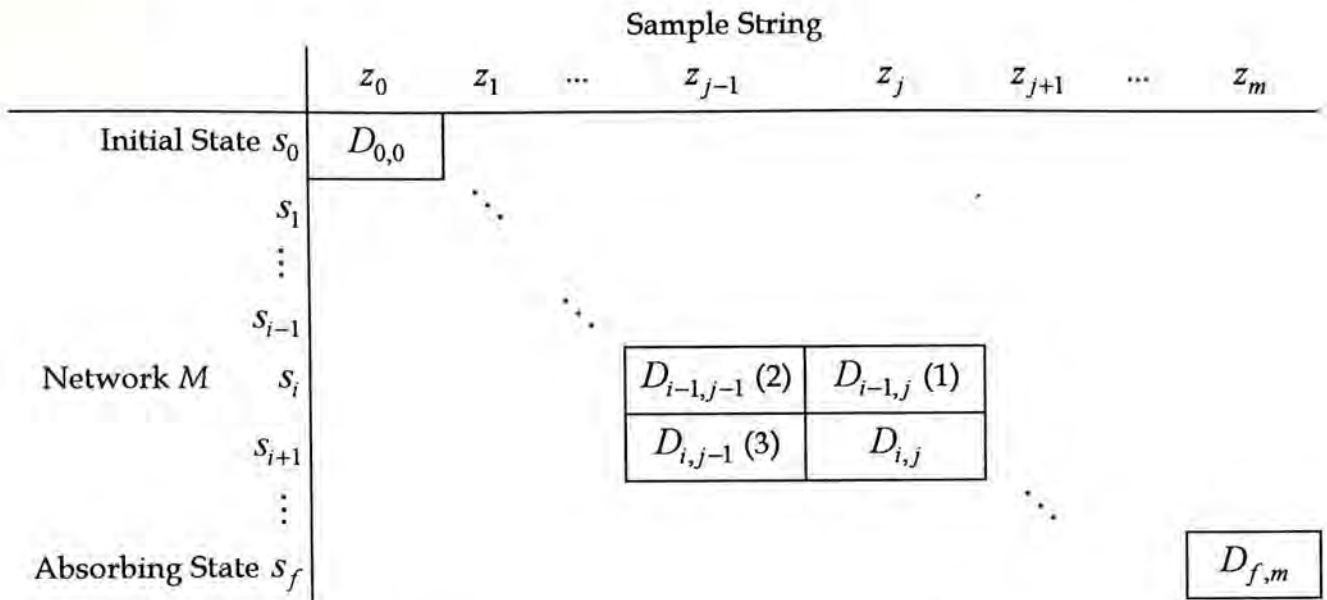


Fig 3.3 Dynamic programming matrix. (1) Deletion, (2) Match or substitution, (3) insertion.

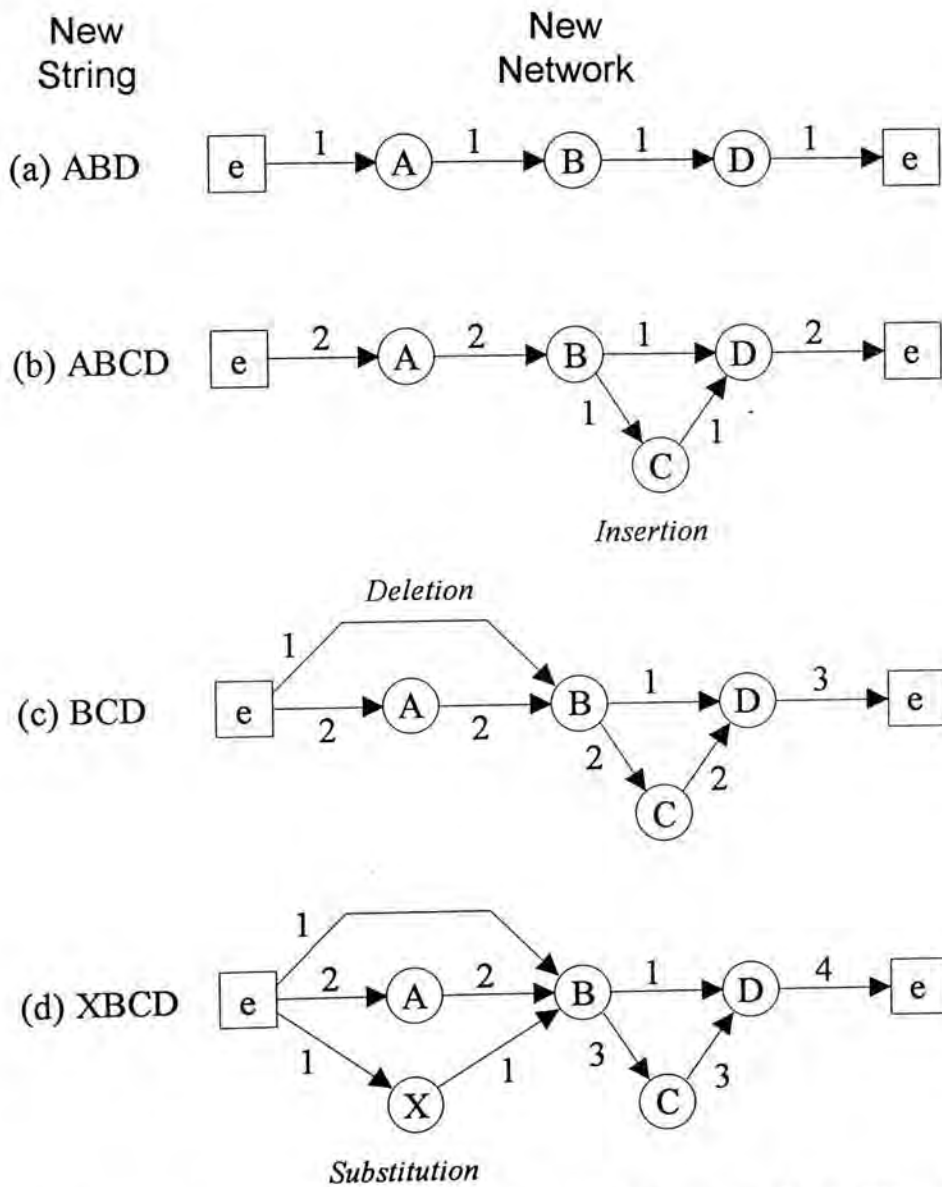


Fig 3.4 Simple example illustrated the possible network modifications of the inference process.

Network modifications are illustrated in Fig 3.4 [7] where a small inferred Markov network was constructed from the strings "ABD", "ABCD", "BCD", and "XBCD". From the inferred Markov network in Fig 3.4, the strings can be aligned as follows (*e* represents the null character)

A	B	<i>e</i>	D
⋮	⋮	⋮	⋮
A	B	C	D
⋮	⋮	⋮	⋮
<i>e</i>	B	C	D
⋮	⋮	⋮	⋮
X	B	C	D

Classifiers can be constructed based on the string-to-network alignment which consists of two phases, the training and the recognition phases. In the training phase, a Markov network will be constructed based on the training sample strings as described in Section 3.1. In the recognition phase, the testing string will be aligned into the network to find out the maximum probability that generating the string. However, no modification will be applied after the alignment is completed. The resulting probability of generation for a testing string can be used as a discrimination between others testing strings.

3.3.1 Aspects concerning the empty-states

As stated in Section 3.1, the introduction of the empty states in the inferred Markov networks is to ensure the consistency between the string-to-network alignment and the network modifications. That is, the criteria for the insertion of empty states are embedded in the cost functions in equation (3.2 - 3.4). For the alignment between the single-path network containing the string "abcd" and the string "abd", the result can be simply obtained by observation as follows :

a	b	c	d
⋮	⋮	⋮	⋮
a	b	<i>e</i>	d

Character "c" is missing in the incoming string. So, the network modifications needed in order to incorporate the incoming string are

1. Match "a",
2. Match "b",
3. Delete "c", and
4. Match "d".

and the resulting network is

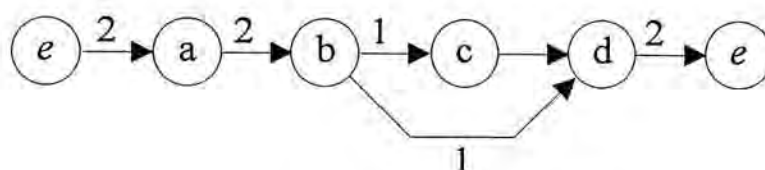


Fig 3.5 The resulting which incorporates the string "abd" into the single path network constructed from the string "abcd"

In general, the deletion and substitution applied on a state s_i , where $s_i \neq s_0$ and $s_i \neq s_f$, can be illustrated in Fig 3.6(b) and Fig 3.6(c) respectively with s_{i-1} as the predecessor in the last modification and s_{i+1} as the successor in the next modification as shown in Fig 3.6(a). As in Fig 3.6(b), the first subexpression, $\log((f_{i-1,i}+1)/(f_{i-1}+1))$, contributing to the overall probability of generation in Eq(3.2) is corresponding to the arc from s_{i-1} to e_1 . The second subexpression, $\log(1/(f_i+1))$, is the arc from e_1 to e_2 by passing s_i , where both e_1 and e_2 are inserted empty states which generate the empty string e . The situation is similar in the case of substitution as in Fig 3.6(c).

In most cases, the empty states inserted can be eliminated. For instance, in Eq(3.2), the only elimination can be done between the second and the third subexpressions when

$$\begin{aligned} f_{i-1,i} + 1 &= f_i + 1 \\ \Rightarrow f_{i-1,i} &= f_i \end{aligned}$$

That means s_{i-1} is the only predecessor of s_i . As in Fig 3.6(b), the edge a_1 and the empty state e_1 can be eliminated if a_1 is the only path from the s_{i-1} to e_1 . In addition, a_2 and e_2 can be eliminated if $f_i = f_{i,i+1}$. This required that the next modification should be a match, a deletion, or a substitution. Such elimination cannot be applied when s_i is being considered since future modification information is required. For a special case when

$$(f_{i-1,i} = f_i) \wedge (f_{i-1} = f_{i,i+1})$$

which means that s_{i-1} is the only predecessor and s_{i+1} is the sole successor of s_i , then both e_1 and e_2 can be eliminated. This is the case occurred in Fig 3.5.

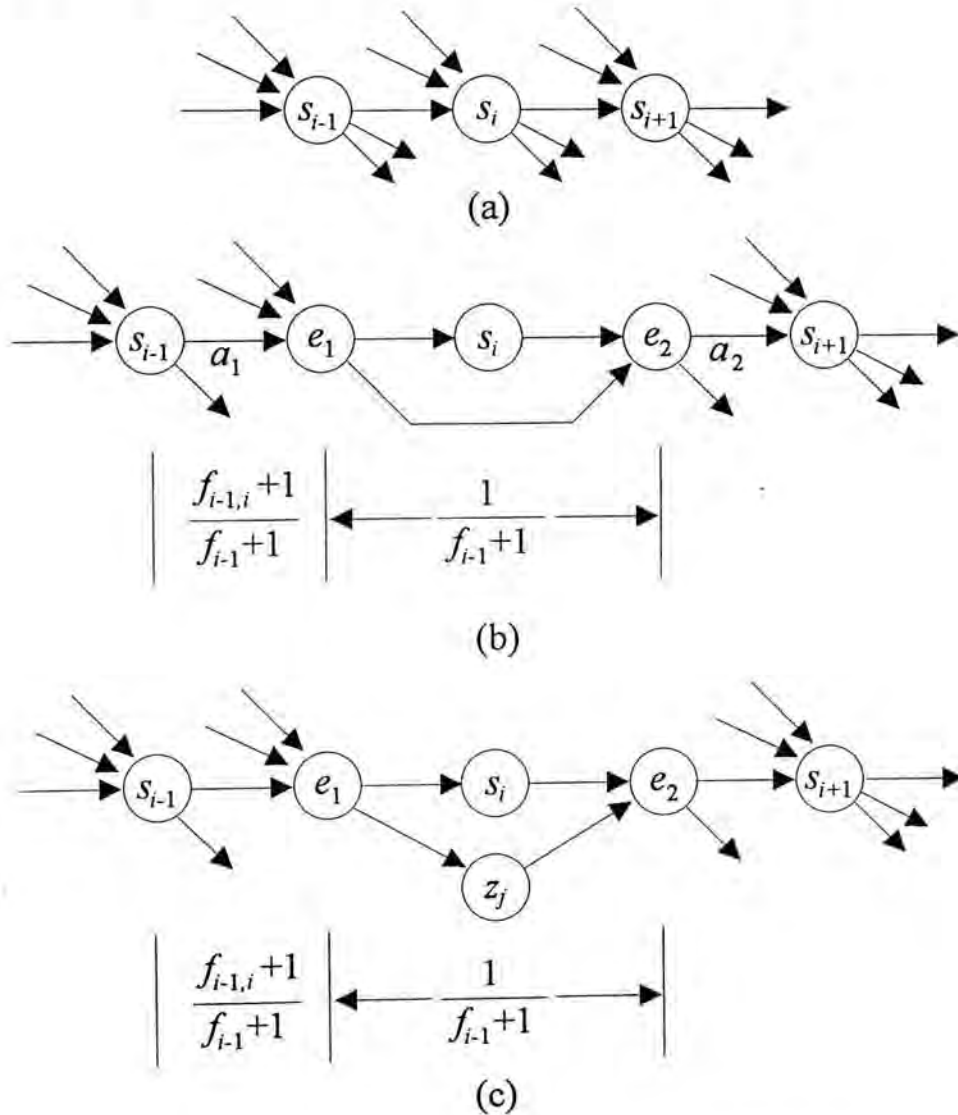


Fig 3.6 (a) A non-terminate state s_i with predecessor s_{i-1} and successor s_{i+1} . (b) Deletion. (c) Substitution.

In real implementation, eliminations need not be considered one by one. Cases can be summarized and pre-computed in order to speedup the time for the network modifications. The network modifications will be applied from the absorbing state s_f back to the starting state s_0 as the path matrix is traced in reverse order. The insertion of empty states can be summarized as follows:

If the current and the next (which modify the previous state as the modification is done in reverse order) modification is deletion, substitution, or "insert after", then a empty state should be inserted between the current and the previous state.

3.3.2 Entropy characteristics of Markov networks

This section aims at finding a good measure to discriminate between alternate maximum probability traces. Detailed analysis concerning the properties of the inferred Markov networks can be found in [34].

The entropy of a random variable is defined in terms of its probability distribution and can be shown to be a good measure of randomness or uncertainty.

Let x be a random variable with sample space $X = \{x_1, x_2, \dots, x_N\}$ and probability measure $P\{X_n\} = p_n$. The entropy of x is defined as [23]

$$H_X = -\sum_{n=1}^N p_n \log(p_n) \quad (3.5)$$

3.3.2.1 Network Entropy

An inferred Markov network M can be viewed as a Markov source of information with sample space $S = \{s_0, s_1, \dots, s_f\}$. From the long-run property of the absorbing Markov chain, a recurrent chain M_R can be constructed from M by changing p_{ff} from 1 to 0 and p_{f0} from 0 to 1 for starting state s_0 and final state s_f . The new process M_R has a steady-state distribution over the nodes,

$\Pi_M = (\Pi_0, \Pi_1, \dots, \Pi_n, \Pi_f)$, such that Π_k is the *asymptotic probability* of being in s_k . It has been shown that Π_k is also the expected fraction of time that M is in s_k [1][2][34].

The entropy for an inferred Markov network M with steady-state distribution Π_M can be formulated as

$$H_{\Pi_M} = -\sum_{\Pi} \Pi_i \log(\Pi_i) \quad (3.6)$$

3.3.2.2 Path Entropies

Let $X = \{x_1, x_2, \dots, x_N\}$ be a set of mode sequences through the network M , the path entropy of X and M is

$$H_{\rho_X} = -\sum_{x_i \in X} p_{x_i} \log(p_{x_i}) \quad (3.7)$$

$$H_{\rho_M} = -\sum_r p_r \log(p_r) \quad (3.8)$$

where p_r is the probability of path r in M .

In fact, H_{ρ_M} reflects the landmarking inferred from the training set in the sense that the lower the path entropy, the greater the concentration of probability mass in repetitive substrings.

The value of H_{ρ_M} is smart enough for evaluating the goodness of the installation of a new string into a network. However, information for calculating the value is not available until the string-to-network alignment process is completed. As stated in the previous section, if more than one alignment achieve the same maximum probability, only one of them is used to modify the network. In order to enable the decision within the alignment process, a new criterion which enables quicker computation will be discussed in the next section.

3.3.2.3 Network Modification and Entropies

In this section, a faster calculation will be discussed in order to approximate the changes in the network entropy after the substring $z_1 z_2 \dots z_j$ has been installed.

Suppose the network M has n non- e node (non-empty node which generate non- e symbol) with frequencies f_i where $1 \leq i \leq n$ and $F = f_1 + f_2 + \dots + f_n$.

From Eq(3.6), we have

$$\begin{aligned} H(\Pi_M) &= H\left[\frac{f_1}{F}, \frac{f_2}{F}, \dots, \frac{f_n}{F}\right] \\ &= -\sum_{\Pi} \frac{f_i}{F} \log\left[\frac{f_i}{F}\right] \\ &= \frac{1}{F} \sum_{\Pi} f_i (\log F - \log f_i) \end{aligned} \quad (3.9)$$

For any trace which install a new sample string O_k with length m in M , we can write

$$m = m_1 + m_2$$

where m_1 = number of new non- e nodes created

m_2 = number of non- e nodes matched

M will be modified into M' after the installation of O_k and $F' = F + M$. Therefore, the entropy of the new network M' is the sum of Eq(3.10), Eq(3.11), and Eq(3.12).

1. m_1 new nodes, each contributing $\log(F')/F'$. (3.10)

2. m_2 reenforced nodes, each contributing a term

$$\left[\frac{f_i+1}{F'}\right] \log\left[\frac{F'}{f_i+1}\right] \quad (3.11)$$

3. $n - m_2$ unreenforced nodes, each contributing a term

$$\left[\frac{f_i}{F'} \right] \log \left[\frac{F'}{f_i} \right] \quad (3.12)$$

The changes caused by installing O_k is

$$\begin{aligned} \Delta H &= m_1 \frac{\log F'}{F'} + \frac{1}{F'} \sum_n f_i \log F' + m_2 \frac{\log F'}{F'} \\ &\quad - \frac{1}{F'} \sum_{m_2} (f_i + 1) \log (f_i + 1) - \frac{1}{F} \sum_{n-m_2} f_i \log f_i \\ &\quad - \frac{1}{F} \sum_n f_i \log F - \frac{1}{F} \sum_n f_i \log f_i \\ &= \log \frac{F'}{F} + \frac{m}{FF'} \sum_n f_i \log f_i + \frac{1}{F'} \sum_{m_2} (f_i \log f_i - (f_i + 1) \log (f_i + 1)) \end{aligned} \quad (3.13)$$

From Eq(3.13), the only factor depending on the actual trace for O_k is

$$\sum_{m_2} (f_i \log f_i - (f_i + 1) \log (f_i + 1)) \quad (3.14)$$

This factor can be used to discriminate between alternate maximum probability traces, i.e., one can choose the alternative match with the smaller value calculated by Eq(3.14).

3.4 Forced Landmarks in String-to-Network Alignment

For a finite set O of sample strings, *landmarks* are the substrings appearing in large percentages of samples [34]. The network representation of the landmark substrings is called the *forced landmarks*. The computation which incorporates the forced landmarks in the DP for string-to-network alignment was introduced by Gregor & Granum in [10]. The following paragraphs summarizes the basic ideas of such computation.

Inference of Markov networks with forced landmarks corresponds to constraining the dynamic programming string-to-network alignment to take place on a per substring basis [10]. Each sample string in O will be partitioned into N substrings using *a priori* knowledge about unique landmarks. Such substrings are then be grouped into N subsets $O_k, k = 1, \dots, N$. The Markov subnetworks M_k inferred from the data subsets O_k will be concatenated into a single network, $M = M_1 M_2 \dots M_N$, which model the structure of the pattern class represented by O . A forced landmark is a single arc which links up the absorbing state of on subnetwork, M_k , and the starting state of the next, M_{k+1} . Both states output the empty string e since the landmark represents a transition.

The time complexity for the inference of forced landmark inferred Markov networks can be reduced since the substrings, O_k , in a sample string, O , are required to align with one corresponding subnetwork, M_k , only. Computational savings can be shown in Fig 3.7 [10]. Fig 3.7(a) illustrates a unconstrained string alignment with an ordinary Markov network where the whole cost matrix should be computed. Fig 3.7(b) and (c) illustrates the constrained alignments with Markov networks with one and two forced landmarks respectively where the shaded areas of the cost matrices are not computed.

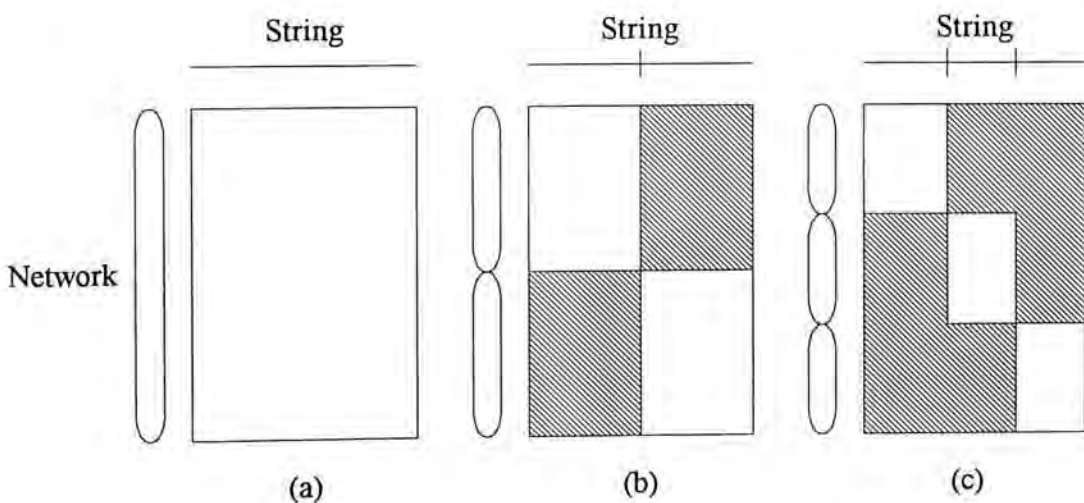


Fig 3.7 (a) Unconstrained string-to-network alignment. (b) and (c) Constrained alignments with networks that have one and two landmarks, respectively.

Perhaps the most important step for the inference of Markov networks with forced landmarks is the estimation of landmark positions in the sample strings. A stronger confidence estimation can be achieved if the position of a unique landmark is typical for the pattern class represented by a network. It has been shown that, with the centromere position as an estimated landmark, the classification (recognition) rate of 7 chromosome classes can compete with the original model which uses the unconstrained Markov networks.

In next chapter, a new landmark estimating method will be introduced which employs no inherit information on the landmark position. More computational savings can be achieved when more forced landmarks are employed.

Chapter 4

Landmark Finding in Markov

Networks

The basic forced landmark model which required *a priori* knowledge on the segmentation of sample strings is presented in last chapter. In this chapter, a new string segmentation method will be presented. This method can deal with sample strings without *a priori* knowledge on the segmentation of the sample strings. The segmentation method will be presented in Section 4.1. Section 4.2 provides a profile processing algorithm which transforms the chromosome profiles into strings for analysis in Section 4.3. In Section 4.3, chromosomes in string representation will be employed as an example for the analysis of the string segmentation method. Finally, an experiment on the chromosome classification with forced landmarks found by the segmentation method will be presented in Section 4.4.

4.1 Landmark Finding without *a priori* Knowledge

Landmarks of a set of sample strings are retained in the inferred Markov network through the string-to-network alignment. The string segmentation method introduced in this section uses the string-to-network alignment as an *empirical landmark finder* for a given set of strings. Landmarks found in an inferred Markov network guarantees that it can be found in *all* aligned strings. The algorithm uses

the positions of landmarks in the aligned strings to approximate those in the unaligned strings. In the area of classification problems, the landmark positions can be found in the Markov network inferred from the training set and such landmark information will be applied to subsequent testing strings.

Given a inferred Markov network, M , all the nodes which occurs with probability 1 will be identified in order to find the landmarks of the training set of strings, O . Such a set of nodes, which are called the *landmark states* as convenient, will be partitioned into two sets, one for the nodes that generate observed outputs and the other for the empty states. A forced landmark can be found by inserting an empty state between a landmark state and it's successors as shown in Fig 4.1. The relative landmark positions in the training strings which corresponds to the landmark states in the Markov network will be extracted. For a given training string, such positions can be measured by following the state sequence in the realization of the Markov network that generates the string.

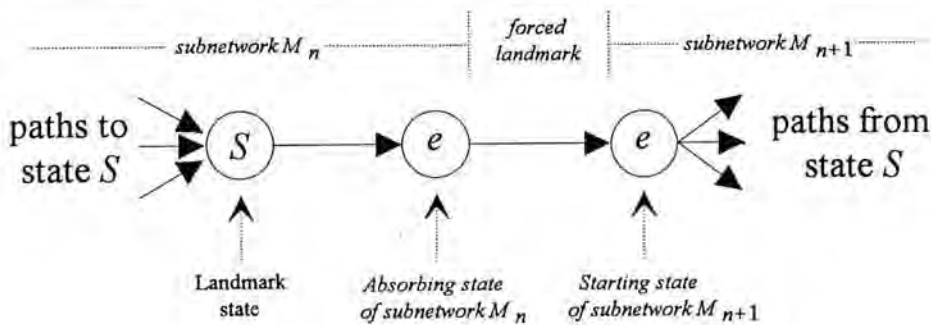


Fig 4.1 Construction of a forced landmark from a landmark state.

Since more than one landmark state may be found in a Markov network, one must decide which landmark(s) will be used for recognition. The original Markov network will then be split into a number of subnetworks according to the landmark states selected. As in Fig 4.1, two empty states which formed a forced landmark will be inserted for each selected landmark state. The first empty state is the absorbing state of the subnetwork M_n while the second empty state is the starting state of M_{n+1} . These subnetworks together with the relative landmark positions can be

applied in the recognition phase which provides a faster alternative to the original alignment.

In recognition phase, each string in the testing set will be partitioned into substrings according to the selected relative landmark positions. Normally, the realization of a landmark state in an input string can be found in the neighborhood of the estimated position¹ where the character in such position is equal to the character generate by the landmark state. The substrings extracted from an input string will be aligned with the Markov subnetworks in sequence. As stated in Section 3.4, the string-to-network with forced landmarks is a concept which uses a sequence of small alignments to approximate the original alignment. Therefore, in the sequence of small alignments, the result of the previous alignment will be propagated to the current computation. The probability of generating the whole string by the sequence of subnetworks is identical to the product of the probability of generating the substrings in the corresponding subnetworks.

With the aid of *a priori* knowledge, the string segmentation model described in Section 3.4 can speed up the string-to-network alignment in both training and testing phase of a classification problem. The method presented in this section requires the result from the training phase to find out the landmark states. Therefore, computational savings can be obtained in the testing phase only. This can be justified since more landmarks can be found by the empirical landmark finder and the computation saving is proportional to the number of landmarks used while the size of the testing set is much larger in real life applications.

¹The estimated position of a forced landmark in a string can be calculated as

$$\text{estimated position} = \text{relative landmark position} * \text{length of the string.}$$

However, the character in such position in the string may not equal to the character generated by the corresponding landmark state (which generate a observable output, a non-empty state). Therefore, a search scheme should be designed in order to find a position with the right character.

4.2 Chromosome Profile Processing

A chromosome profile is a series of data points sampled along the medial axis of a chromosome. Each data point represents the grey level of the corresponding location in the original chromosome image. The aim of the profile processing, on the one hand, is to minimize the effect of the interference produced in the image acquisition procedure. On the other hand, essential features can also be extracted in the profile processing phase.

The approach described in this section, which is called Idealized Profile [7], tries to include the sequential nature of the band patterns explicitly. With similar philosophy, the Band-Transition Sequences (BTS) has been employed for both automatic and visual classification with competitive levels of success. The following procedure consisting of 4 phases describes the construction of a discrete version of the Idealized Profile which transform the chromosome profiles into character strings such that both sequential and band-transition nature of band patterns can be preserved [7].

(1). 3-point smoothing (weights: 1,2,1) : The smoothing processing aims at reducing the noise in the profile. Such noise may appear in a form of local extremum with unreasonable increment or decrement with its neighbors. These extremums will affect the result very much since it will be selected as the sample level of its neighborhood.

(2). Differential analysis [15] : The differential analysis is a non-linear transformation applied on the smoothed profile. The process tries to establish the positions of inflection of the profile. The hypothesis is that positions of inflection can be used as estimations of transitions between bands of different densities. Normally, one local extremum will exists between two inflections along the profile. The subprofile lies between two successive inflections can be estimated by the

corresponding local extremum by extending its density over that region. As a result, the smoothed profile will be transformed into simpler distinguishable bands.

For each finite set X , if $N(x)$ is the neighborhood of x and F is a function of X , we associate F with two other functions, the local maximum function \bar{F} and the local minimum function \underline{F} as follows:

$$\bar{F}(x) = \max\{F(y) | y \in N(x)\},$$

$$\underline{F}(x) = \min\{F(y) | y \in N(x)\}.$$

We define the sharpening transformation S as

$$(SF)(x) = \begin{cases} \bar{F}(x) & \text{if } (\bar{F}(x) - F(x)) \leq (F(x) - \underline{F}(x)) \\ \underline{F}(x) & \text{otherwise} \end{cases}$$

It has been proved that the sequence $S^n F$ is pointwise converged where n is the number of applications of S .

The non-linear transformation is the applications of the sharpening transformation iteratively over a finite set X until S converge. For a 2-D image, the neighborhood $N(x)$ can be defined as 4- or 8-neighbor. In 1-D cases, $N(x)$ is defined as a 2-neighbor case (ie. only the immediately left and right neighbor of a point is considered).

(3). Non-linear mapping : In this phase, a non-linear mapping will be applied over the band sequence obtained from the previous phase. The transformation simplifies the profile by remapping the density of bands into six levels [7] while retaining all the transitions between bands. This can be accomplished by scaling the bands linearly and then examining the transitions one by one from left to right. If two adjacent bands have been mapped onto the same level, then the one in the right hand side must be modified in order to maintain the transition before the mapping. The adjustment is done according to the difference between the densities of two bands before mapping.

(4). **String construction** : The band patterns in the discrete idealized profile obtained from phase 3 will be expressed in the form of a difference string. Band transitions are emphasized in strings based on the differences between successive incremental bands. In the experiments, the symbols $\{...,c,b,a,=,A,B,C,..\}$ are used to represent the differences $\{...,-3,-2,-1,0,1,2,3,...\}$.

4.3 Analysis of Chromosome Networks

This section aims at providing a brief picture on the distribution of the empirical landmarks in chromosome networks. As stated in [10], the mean and the variance of the distribution of the estimated landmarks may be obtained from the string-to-network alignment if it is a Gaussian distribution and based on one-dimensional feature vector. In fact, it is very difficult in giving a model for the distribution of the estimated landmarks especially a model which depends on one-dimensional feature vector. Therefore, the remaining of this section is to show that the distribution of the empirical landmarks in chromosome networks is near normal so that the landmark position can be estimated.

The Copenhagen chromosome database has been selected for analysis. The database contains 180 blood cells. For each chromosome type, 100 samples were extracted which after the profile processing stage described in the previous section, were aligned sequentially to form a Markov network. The empirical landmark states were extracted and the relative positions of such landmarks in the aligned samples were measured. Table 4.1 summarizes the empirical landmarks found in the chromosome networks with size 100 together with the relative positions of the centromeres which are employed in the basic forced landmark model [10]. As described in Section 4.1, each empirical landmark corresponds to a landmark state in the inferred Markov network. The landmark states considered in Table 4.1 are the states which output observable symbol (non-empty states). The relative positions of

the centromeres are calculated according to the information provided in the Copenhagen chromosome database.

Type	Max. length	Min. length	Avg. length	Centromere position	Number of Empirical LMs
1	130	64	90.66	0.4571	22
2	121	63	86.19	0.3702	16
3	103	45	73.74	0.4368	16
4	97	47	69.80	0.2774	16
5	99	48	67.65	0.2768	14
6	90	49	66.14	0.3679	11
7	89	36	60.29	0.3620	13
8	100	40	56.19	0.3304	11
9	70	39	53.42	0.3286	11
10	74	19	52.85	0.3133	6
11	72	39	53.21	0.3676	7
12	73	39	52.98	0.2888	10
13	62	30	44.82	0.1879	7
14	68	31	43.85	0.1961	6
15	58	31	42.90	0.2081	7
16	54	30	39.22	0.4042	5
17	54	29	38.99	0.3148	5
18	48	28	36.31	0.2854	7
19	44	21	32.23	0.3866	2
20	44	24	32.48	0.3971	2
21	36	17	26.21	0.2437	5
22	43	20	28.98	0.2593	3

Table 4.1 Empirical landmarks found in chromosome networks with size 100.

A series of histograms which exhibit the distribution of the empirical landmarks of types 5, 13, and 22 are shown in Fig 4.2. These histograms were sampled with an interval of 0.05. As shown in the histogram for types 5 and 13, the distributions of the relative positions of centromeres in these types can be approximated by an empirical landmark which appears to be normal. Such empirical landmarks can also be found in other 8 types. In fact, the centromere position of a chromosome can be deduced from an inferred Markov network with the centromere position in the training set as a forced landmark [11]. Therefore, if the centromere position can be used as a forced landmark, other empirical landmarks can also be employed. On the other hand, the distribution of the centromere position may not appeared to be normal nor can be approximated by an empirical landmark as shown in the histogram for the type 22.

The rest of this section will show, in numerical calculations, the normality of the distribution of the empirical landmarks found in types 5, 13, and 22. If the distribution of the relative position of a empirical landmark is normal among all the training samples, then the mean of the relative positions of the landmark is expected to be a proper estimation of the empirical landmark

For a normal distribution with mean μ and standard deviation σ , the intervals $(\mu-\sigma, \mu+\sigma)$, $(\mu-2\sigma, \mu+2\sigma)$, and $(\mu-3\sigma, \mu+3\sigma)$ contain the probabilities 0.6826, 0.9544, and 0.9974, respectively. Alternatively, the probabilities outside these intervals are roughly 1/3, 1/20, and 1/300, respectively. With a reasonably large sample size, one can expect the sample mean \bar{X} to be close to μ and the sample standard deviation s to be close to σ .

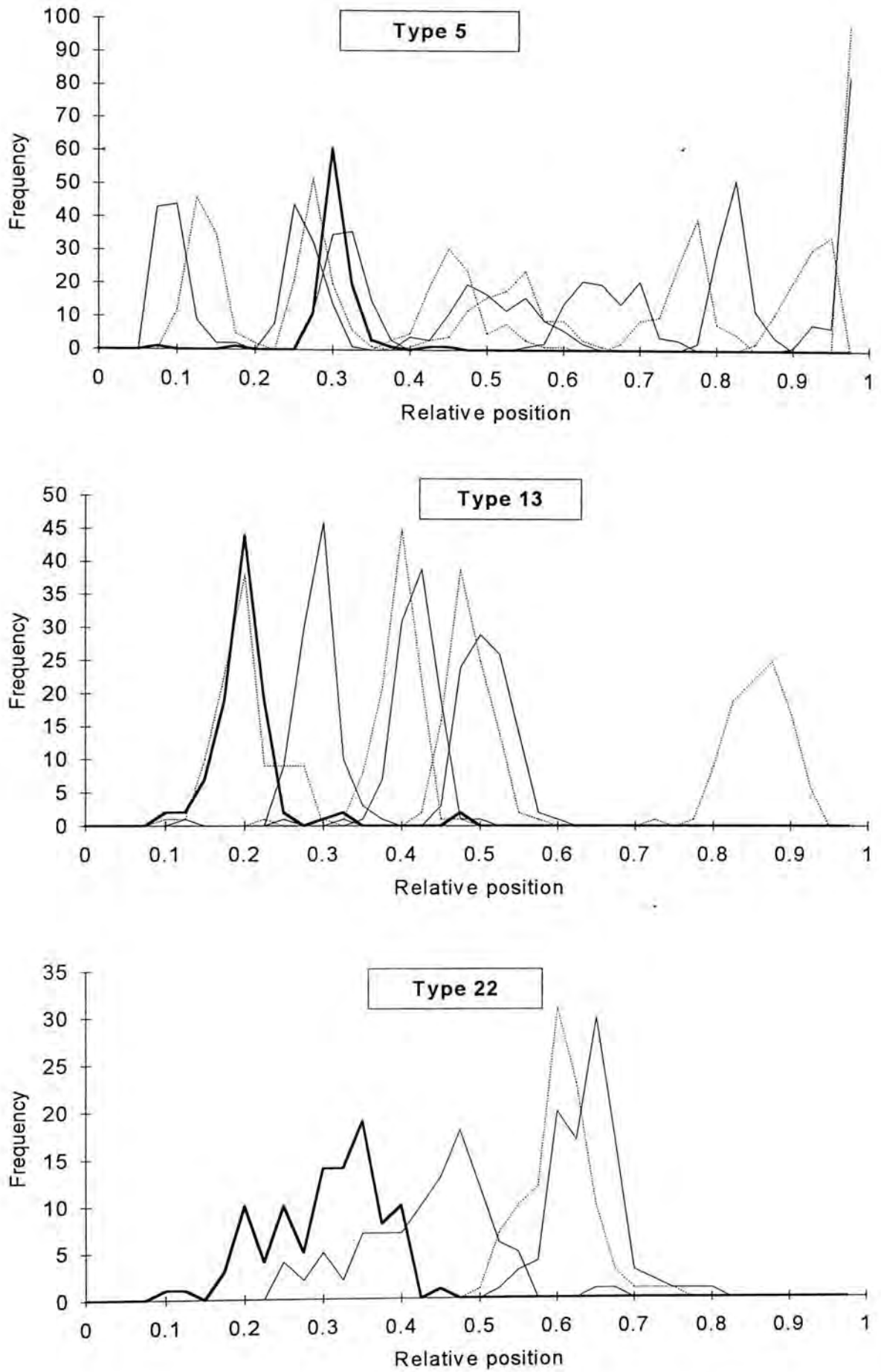


Fig 4.2 Histograms of the distribution of empirical landmarks in types 5, 13, and 22.

(Bold line represents the centromere)

In order to estimate the quantitative measure on the normality of the distribution of the empirical landmarks, the sample mean \bar{X} and the sample standard deviation s have been measure based on the relative landmark position extracted from the chromosome networks with size 100. Since the sample size is relatively small, intervals which closed to the sample mean are tested. Such intervals include $(\mu-\sigma/2, \mu+\sigma/2)$, $(\mu-\sigma, \mu+\sigma)$, and $(\mu-3\sigma/2, \mu+3\sigma/2)$ which contain the probabilities 0.3830, 0.6826, and 0.8664, respectively. The discrepancy measure

$$|\hat{p} - p| / \sqrt{\frac{p(1-p)}{n}}$$

has been employed which measure the discrepancy between the observed relative frequency \hat{p} and the expected fraction p . A large discrepancy, say 3, would indicate lack of normality.

Table 4.2 shows the discrepancy measure of the empirical landmarks in type 5, 13, and 22. It has been shown in the table that the discrepancy measure for the distribution of most empirical landmarks are relatively small. Therefore, the distribution of most empirical landmarks are nearly normal and \bar{X} can be used as an approximation of μ in most cases. There are a few exceptional cases such as last 2 landmarks in type 5. The reason for such a large discrepancy is that these landmarks are closed to the end of the sample string (the mean relative position of the second to last landmark for type 5 is 0.979 and exactly 1 for the last one). For example, symbol 'a' occurs in the last position in every sample strings of type 5, so the discrepancy measure is relatively large which is greater than 12.

Type 5	Relative position		Discrepancy measure		
Landmark	Mean	Std. dev.	$\sigma/2$	σ	$3\sigma/2$
1	0.1054	0.0195	0.5554	1.0183	0.9876
2	0.1463	0.0207	2.2011	1.6629	0.7760
3	0.2761	0.0196	1.3783	0.1590	0.4820
4	0.2915	0.0201	1.1726	0.0559	0.4820
5	0.3278	0.0214	1.5840	0.9152	0.1058
6	0.4728	0.0424	2.8182	0.8035	1.3638
7	0.5223	0.0510	1.2960	0.7004	0.1058
8	0.5426	0.0491	0.4731	0.0559	0.7760
9	0.6667	0.0442	1.5017	1.1301	0.6937
10	0.7727	0.0323	2.2011	1.2332	0.4820
11	0.8321	0.0184	2.8182	1.4480	0.1881
12	0.9332	0.0258	1.7074	1.1301	1.5754
13	0.9790	0.0147	5.4925	4.8854	1.2815
14	1	0	12.6924	6.8190	3.9268

Type 13	Relative position		Discrepancy measure		
Landmark	Mean	Std. dev.	$\sigma/2$	σ	$3\sigma/2$
1	0.2124	0.0351	1.9954	0.1590	0.3997
2	0.3028	0.0284	1.7897	2.3074	1.8694
3	0.4063	0.0294	1.9954	2.0925	2.1633
4	0.4293	0.0299	1.9954	2.5222	2.1633
5	0.4981	0.0320	1.9954	2.3074	2.1633
6	0.5211	0.0329	0.9668	2.0925	1.5754
7	0.8713	0.0378	0.4731	1.1301	1.2815

Type 22	Relative position		Discrepancy measure		
Landmark	Mean	Std. dev.	$\sigma/2$	σ	$3\sigma/2$
1	0.4489	0.0850	0.8846	0.5887	0.1058
2	0.6149	0.0435	1.9954	1.2332	0.1058
3	0.6513	0.0438	1.7897	1.6629	0.3997

Table 4.2 Discrepancy measure of the distribution of landmarks in type 5, 13, and 22.

4.4 Classification Results

With the inferred Markov networks created in Section 4.3, two classification experiments have been conducted. The first experiment tests the inferred Markov network with one empirical landmark (LM). While the second experiment examines the effect of two empirical landmarks networks.

6 chromosome types have been selected. In each type, a total of 100 chromosomes were extracted to create the inferred Markov network (chromosome network) and 30 extra chromosomes for the testing set. In the first experiment, a empirical landmark was employed in the testing phase. Such empirical landmarks are summarized in Table 4.3.

Type	5	13	18	20	21	22
Relative landmark position	0.3280	0.4981	0.6532	0.6789	0.5040	0.6149

Table 4.3 Empirical landmarks used in experiment one.

The aligned probabilities in each alignment were normalized with the Maximum Representative Probability (MRP) of the tested chromosome networks which minimized the effect of the length of the samples in the alignment process. The result of this experiment is shown in Table 4.4.

Type	5	13	18	20	21	22	Average
No LM	90%	90%	83.3%	83.3%	100%	83.3%	88.3%
1 LM	96.7%	90%	83.3%	86.7%	96.7%	86.7%	90%

Table 4.4 Classification result of experiment one.

The average correct rate is 90% which can be compared with the original forced landmark model (92-93%) as claimed in [10].

In the second experiment, two empirical landmarks was employed. Since only 2 empirical landmarks were found in type 20 and these landmarks are closed together (0.6789 and 0.7528), therefore only one of them was selected. Table 4.5 summarizes the empirical landmarks used in this experiment.

Type	5	13	18	20	21	22
LM 1	0.2924	0.2124	0.2389	0.6789	0.3363	0.4489
LM 2	0.7728	0.4981	0.6532	Not applied	0.5040	0.6149

Table 4.5 Landmarks used in experiment two.

The classification is shown in Table 4.6. There is a penalty on the correct rate (5%) when two empirical landmarks are employed. The tradeoff with such penalty is the time saved in the string-to-network alignments.

Type	5	13	18	20	21	22	Average
No LM	90%	90%	83.3%	83.3%	100%	83.3%	88.3%
2 LM	93.3%	83.3%	83.3%	96.7%	83.3%	70%	85%

Table 4.6 Classification result of experiment two.

Type	5	13	18	20	21	22	Average
1 LM	0.5593	0.5	0.5469	0.5640	0.5	0.5264	0.5328
2 LM	0.3679	0.3786	0.3490	0.5640	0.3863	0.3774	0.4039

Table 4.7 Fraction of time needed.

Table 4.7 shows the fraction of time, with respect to computation with no landmarks, needed in the string-to-network alignments when forced landmark is

considered. This table was calculated with the assumption that given a relative landmark position a , the fraction of edges that precedes the corresponding landmark state is near or equal to a . For the cases with one landmark, if a is the relative position of the landmark, the fraction of time needed to complete the string-to-network alignment is

$$a^2 + (1-a)^2.$$

The computation is similar in two landmarks case. If a and b represent the relative position of the first and second landmark, respectively, then the fraction of time needed is

$$a^2 + (b-a)^2 + (1-b)^2.$$

Chapter 5

Speech Recognition using Inferred Markov Networks

The inferred Markov networks with empirical landmarks were employed in this chapter for the classification of speaker independent speech data. The TIMIT speech database has been used in this chapter. The database consists of the phonemes from continuous English sentences. The features used in the experiments were similar to the traditional approaches which concentrates on the spectrum of the speech windows. Section 5.1 describes the preprocessing tools for the feature extraction delineated in Section 5.3. The data set used in the experiments is discussed in Section 5.2. Section 5.4 discusses the empirical landmarks used in the experiment. Lastly, the classification results are presented in Section 5.5.

5.1 Linear Predictive Analysis

There is a high correlation between adjacent samples of speech waveforms. The basic idea of *linear prediction* of speech is to express the current signal y_n as a linear combination of the past signals [32] as

$$y_n \approx \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \dots + \alpha_p y_{n-p} \quad (5.1)$$

where $\{\alpha_i\}$, $i = 1, 2, \dots, p$ are linear predictive coefficients which satisfy the least mean square prediction error criterion.

By minimizing the mean square prediction, which is the difference between the predicted value and the real value, a p -dimensional first-order simultaneous equations can be obtained as follows:

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & r_0 & r_1 & \cdots & r_{p-2} \\ \vdots & & \ddots & & \vdots \\ r_{p-1} & r_{p-2} & & \cdots & r_0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{bmatrix} \quad (5.2)$$

where $r_j = \overline{y_n y_{n+j}}$ is a correlation coefficient of waveform $\{y_n\}$, and $\{\alpha_i\}$ can be derived by solving Eq(5.2). In practice, r_j is defined within a finite number of samples of N of $\{y_n\}$. A time-window of N samples is applied where y_n exists inside the window ($w_n = 1$) and equals to zeros ($w_n = 0$) outside the window. Thus

$$r_j = \frac{1}{N} \sum_{n=j}^{N-1} y_n w_n y_{n+j} w_{n+j}. \quad (5.3)$$

Calculation of r_j with the time-window assumption is called the *correlation method*.

Since the linear predictive analysis of speech is based solely on the output samples, an *all-pole system* (an autoregression (AR) process in statistical sense) is identified. The poles correspond to the formants of the speech spectrum.

An all-pole system can be identified by the following system response function

$$H(z) = \frac{1}{A(z)} \quad (5.4)$$

where $A(z) = 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots + \alpha_p z^{-p}$, and $\{\alpha_i\}$ are the linear predictive coefficients derived by analysis. The poles of $H(z)$ can be found from the roots of

$$\begin{aligned} A(z) &= 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots + \alpha_p z^{-p} = 0 \\ \Rightarrow z^p + \alpha_1 z^{p-1} + \alpha_2 z^{p-2} + \cdots + \alpha_{p-1} z + \alpha_p &= 0. \end{aligned} \quad (5.5)$$

By solving Eq(5.5), where it is a p -order equation, p roots can be found in the form of $p/2$ complex conjugate pairs. For each conjugate pair

$$z_i = r_i e^{j\theta_i}, \bar{z}_i = r_i e^{-j\theta_i}, i = 1, 2, \dots, p,$$

a formant in the speech spectrum can be identified and the formant frequency, f_i , and the bandwidth, b_i , are given by

$$f_i = \frac{1}{2\pi T} \theta_i = \frac{1}{2\pi T} \arg(z_i) \text{ Hz}$$

$$b_i = \frac{1}{\pi T} |\log r_i| \text{ Hz}$$

where T is the sampling period.

In practice, the order of the linear predictive coefficients, p , should be two times of the number of formants in the frequency range of 0 to $1/2T$. This criterion guarantees that the roots in Eq(5.5) correspond almost exactly to the formants.

5.2 TIMIT Speech Database

The TIMIT speech database consist 6300 English sentences recorded from 630 speakers where each speaker spoken 10 sentences. The speakers came from 8 major regions of the United States. The speech signals were digitized into 16-bit PCM format with a frequency of 16 KHz. 20 male speakers were selected in the experiments with each spoke around five sentences. Six phoneme groups were extracted from the sentences by the labeling index provided by the TIMIT database. The phoneme groups are "aa", "ae", "ah", "er", "ih", and "iy". From each of the phoneme group, 100 samples were extracted which consists of 50 training samples and 50 testing samples.

5.3 Feature Extraction

A preprocessed speech sample will be divided into a number of overlapped speech windows where each speech window contains a fixed number of data points (the *window size*). A feature vector will be extracted for each speech window. Therefore, a speech sample can be expressed in terms of a set of ordered feature vectors. All feature vectors from the speech samples will be collected and grouped, according to a certain distance measure, into a number of clusters. The collection of the characteristic feature vectors (which are called the *centroids* of the clusters) is called a *codebook*. The size of the codebook is identical to the number of clusters which is selected before the feature vectors are grouped. An index will be assigned to each feature vector according to the cluster to which the feature vector belongs. Since the string-to-network alignment can accept symbol strings only, a symbol is assigned to each codebook index. For a size-8 codebook, symbols 'A', ..., 'H' are selected. A speech sample will be transformed into a *string* where the n -th character in the string represents the codebook index of the feature vector of the n -th speech window.

The features used in the experiments were based on the distribution of the first and the second formants, f_1 and f_2 , of the speech signals. These formants were extracted by solving the all-pole system response function with 14-order LPC coefficients. The setting of the window size is 25.6ms (256 data points) and the window overlap is 10ms. The feature vectors, (f_1, f_2) , from the windows of speech samples were vector quantized to create a codebook with the given size. A symbol was assigned to each feature vector (which was extracted from a speech window) of a speech sample according to the codebook index where the vector belongs after quantization.

Half of the speech samples in string representation in each speech group will be used to create the inferred Markov networks using the string-to-network alignment. The classification results of the speech groups will be presented in later sections.

The codebook sizes 16, 32 and 64 were tested for the TIMIT speech database. The best performance was obtained when codebook size 32 is employed. There are no rules on the determination of the codebook size with respect to the recording conditions (continuous or isolated). The choice for TIMIT database is different from that of in other experiments where a speech corpus with isolated Cantonese syllables (a Chinese dialect) is considered. For the Cantonese speech corpus, codebook size 8 is selected which is relatively small among other models using Hidden Markov Models (HMM). The main different is that phonemes in TIMIT database were extracted in continuous sentences while the Cantonese syllables were recorded in isolated manner. Experiments on Cantonese speech recognition is presented in Appendix A.

5.4 Empirical Landmarks in Speech Networks

In Section 4.3, chromosome networks were selected in order to investigate the distributions of the empirical landmarks. The discrepancy measures were employed to examine the normality of the distributions of the empirical landmarks. Chromosomes are the samples with strong sequential nature in selected features. For the analysis in Section 4.3, the states in chromosome networks which output observable symbol (non-empty states) with probability equal to 1 were selected as empirical landmarks. The robustness of such empirical landmarks have been proved in the analysis in Section 4.3 and the classification result given in Section 4.4.

As discussed in the next section, phoneme samples in TIMIT database does not exhibit strong sequential nature, no empirical landmarks which were formed by non-empty states can be found in the speech networks. In these cases, empty states with probability equal to 1 should be considered. Table 5.1 summarizes the characteristics of the phoneme samples and the speech networks with size 50. As shown in the table, the numbers of empirical landmarks found in the speech networks are large. On the other hand, the normality of the distributions of the

empirical landmarks are varying as shown in Table 5.2 which given the discrepancy measure of the empirical landmarks in group "aa". Therefore, the actual empirical landmarks used in the string-to-network alignment should be selected carefully, i.e., the relative position of the landmark should be roughly in the middle of the string and the corresponding discrepancy measures should be small.

Phoneme Group	Sample length				Number of Empirical LMs
	Max.	Min.	Mean	S.D.	
aa	64	10	31.86	10.97	25
ae	65	17	37.54	11.76	62
ah	35	9	19.66	6.30	24
er	57	6	30.06	9.72	36
ih	40	10	18.56	6.55	21
iy	45	5	17.54	7.28	14

Table 5.1 Characteristics of the phoneme samples and empirical landmarks found in the speech networks.

Group "aa"	Relative Position		Discrepancy Measure		
	Mean	S. D.	$\sigma/2$	σ	$3\sigma/2$
1	0.1183	0.1465	1.4982	2.6950	0.6983
2	0.1839	0.1637	1.2073	2.3911	1.1140
3	0.2354	0.1662	1.2073	0.0395	1.1140
4	0.2373	0.1649	0.9164	0.6472	1.1140
5	0.3165	0.1748	1.2073	0.6472	0.6983
6	0.3274	0.1723	0.3346	0.9510	0.6983
7	0.3311	0.1692	0.3346	0.6472	0.6983
8	0.4020	0.1764	0.6255	1.5586	0.1330
9	0.4037	0.1760	0.6255	1.2548	0.6983
10	0.4054	0.1760	0.6255	1.2548	0.6983
11	0.4085	0.1757	0.6255	1.2548	0.6983
12	0.4128	0.1737	1.4982	1.2548	0.6983
13	0.4211	0.1772	1.2073	1.8625	1.1140
14	0.5067	0.2026	1.7892	0.6472	0.1330
15	0.5481	0.2036	1.4982	0.6472	0.1330
16	0.5987	0.1903	1.2073	0.6472	0.1330
17	0.6768	0.1920	1.4982	2.1663	0.6983
18	0.6800	0.1918	0.9164	1.8625	0.6983
19	0.7536	0.1826	1.2073	0.6472	0.1330
20	0.7620	0.1760	1.7892	0.0395	0.2827
21	0.7690	0.1787	1.2073	0.0395	0.1330
22	0.8250	0.1780	1.4982	1.7835	0.5487
23	0.8272	0.1778	1.4982	1.7835	0.5487
24	0.9734	0.0894	7.2294	3.9103	1.5297
25	0.9992	0.0059	8.6840	4.5179	2.3610

Table 5.2 Discrepancy measure of the distribution of landmarks in group "aa"

5.5 Classification Results

The experiments conducted with the TIMIT database were carried in a context-independent manner. Each phoneme was treated as an independent speech signal. With the Hidden Markov Model using one feature, the recognition rate 49.78% was reported for the context-independent phone model with a single codebook [17].

Three experiments have been conducted. Experiment 1 tests the phoneme speech networks with respect to the training set. Experiment 2 deals with the testing sets. The empirical landmarks discussed in last section were employed in experiment 3 concerning the classification of the testing sets with one landmark. Table 5.3 shows the empirical landmarks used in experiment 3 which are located near the middle of the sample strings such that half of the computations needed in string-to-network can be saved. The aligned probabilities were normalized with the Maximum Representative Probability (MRP) of the tested networks.

Group	aa	ae	ah	er	ih	iy
Relative landmark position	0.5067	0.4813	0.4936	0.4870	0.4888	0.4897

Table 5.3 Empirical landmarks used in experiment 3

The recognition rate for the training set is 79% as shown in Table 5.4. The recognition rate for the testing set is 59%. The performance is better than the one using HMM (49.78%) [17]. However, the number of phoneme groups used here is relatively small. The recognition rate dropped to 53% in experiment 3 where one empirical landmark was employed. Dropping of recognition rate indicates that empirical landmarks formed by non-empty states are more robust than the one formed by empty states. However, the recognition rate of individual groups between experiments 2 and 3 are consistent. Further normalization on the classification result with landmarks may be applied.

Group	aa	ae	ah	er	ih	iy	average
Train set	86%	72%	78%	76%	78%	84%	79%
Test set	52%	50%	62%	58%	58%	72%	59%
Test set & 1 LM	46%	44%	64%	46%	50%	68%	53%

Table 5.4 Classification result of the TIMIT database.

Training \ Testing	aa	ae	ah	er	ih	iy
aa	86	0	12	0	2	0
ae	4	72	4	0	12	8
ah	10	2	78	6	0	4
er	2	0	14	76	2	6
ih	0	2	2	2	78	16
iy	0	0	0	0	16	84

Table 5.5 Confusion matrix of experiment 1.

Training \ Testing	aa	ae	ah	er	ih	iy
aa	52	2	34	0	10	2
ae	2	50	8	0	34	6
ah	8	2	62	2	12	14
er	6	0	22	58	10	4
ih	0	0	10	8	58	24
iy	0	0	0	0	28	72

Table 5.6 Confusion matrix of experiment 2.

Training \ Testing	aa	ae	ah	er	ih	iy
aa	46	2	42	0	8	4
ae	2	44	14	0	26	14
ah	12	4	64	2	8	10
er	8	0	30	46	12	4
ih	0	0	16	4	50	30
iy	0	0	0	0	32	68

Table 5.7 Confusion matrix of experiment 3.

Chapter 6

Conclusion

Automation of chromosome classification is a challenging and rewarding task since it is tedious and time consumed. Most effort have been devoted to the statistical approaches as most classification problems do. Structural approaches can be applied because of the sequential nature of the band patterns on chromosomes. The band patterns are also the basis of visual classification.

Inference of Markov network by dynamic programming is the most successful method among all structural approaches in automated chromosome classification. Samples are incorporated into the Markov network through the dynamic programming search on the optimal modification of the original inferred Markov network. The dynamic programming search guarantees that the probability of which the modified network generates a given sample is maximum among all other modifications.

6.1 Suggested Improvements

Improvements on the performance of the inferred Markov networks may concentrates on two external aspects. The first aspect concerns the speed of the string-to-network alignment (the dynamic programming search). The complexity of the string-to-network alignment is proportional to the size of the inferred Markov network being modified. The network is getting larger when more samples are incorporated. The forced landmark concept can speedup the string-to-network alignment which employes the inherited knowledge to segment the input strings.

This concept has been generalized into empirical landmarks where the speedup can be applied with no inherited knowledge on the segmentation of input strings.

The second aspects aims at improving the recognition rate of the classification by inferred Markov networks. The alignment probability of a given input string with the inferred Markov network has been employed as the discrimination criteria. Such alignment probability was calculated by string-to-network alignment and was affected by the size of the network. Therefore, normalization is needed in order to obtain better recognition results. Normalization used in the experiments of this thesis is based on the Maximum Representative Probability (MRP) of the tested inferred Markov network. The input strings will be aligned with networks of different types. The alignment probabilities will be divided (normalized) by the MRP of the corresponding networks. For a given input string, the network which produces the maximum among all normalized probabilities is the type of this sample. A method for explicit normalization of the alignment probabilities may be developed by investigating the probability distribution functions of the inferred Markov networks.

Better recognition result may be obtained by adjusting the normalization parameter where MRP is the initial guess. An optimization problem can be constructed which maximizes the recognition rate of the training set (the input samples which constructed the networks). The idea can be demonstrated by a problem with two types. For each type, N samples are selected to construct the network for that type. Let $P_{i,j,k}$ be the logarithm alignment probability of the k^{th} sample of type j and the network i , and m_i , the normalization parameter of network i . The following inequalities are expected to be satisfied if the recognition rate is 100%:

$$P_{1,1,n} - m_1 > P_{2,1,n} - m_2 \quad , \quad n = 1, \dots, N \quad (6.1)$$

$$P_{2,1,n} - m_2 > P_{1,2,n} - m_1 \quad , \quad n = 1, \dots, N \quad (6.2)$$

Let $D_{1,n} = P_{2,1,n} - P_{1,1,n}$ and $D_{2,n} = P_{1,2,n} - P_{2,2,n}$, Eq(6.1) and Eq(6.2) become

$$m_2 - m_1 > D_{1,n} \quad , \quad n = 1, \dots, N \quad (6.3)$$

$$m_1 - m_2 > D_{2,n} \quad , \quad n = 1, \dots, N \quad (6.4)$$

Eq(6.3) and Eq(6.4) can be simplified by finding

$$D_1 = \max\{D_{1,n}\} \quad , \quad n = 1, \dots, N \quad (6.5)$$

$$D_2 = \max\{D_{2,n}\} \quad , \quad n = 1, \dots, N \quad (6.6)$$

Therefore, Eq(6.3) and Eq(6.4) become

$$m_2 - m_1 > D_1 \quad \Rightarrow \quad m_2 - m_1 - D_1 > 0 \quad (6.7)$$

$$m_1 - m_2 > D_2 \quad \Rightarrow \quad m_1 - m_2 - D_2 > 0 \quad (6.8)$$

The inequalities Eq(6.7) and Eq(6.8) can be solved as in Fig 6.1.

From Fig 6.1, the solution space can be found when two shadowed regions are intersected. This required that one of D_1 or D_2 is negative and the absolute value is greater than another one, or both D_1 and D_2 are negative. However, D_1 and D_2 are positive in most cases. Therefore, the criteria in Eq(6.5) and Eq(6.6) should be relaxed such that the smaller values can be considered. By relaxing the conditions, some inequalities in Eq(6.1) and Eq(6.2) will never be satisfied. Therefore, the recognition rate will less that 100% even a solution is found after relaxation. The optimization problem is to find the optimal combination of D_1 and D_2 such that the recognition rate is maximum.

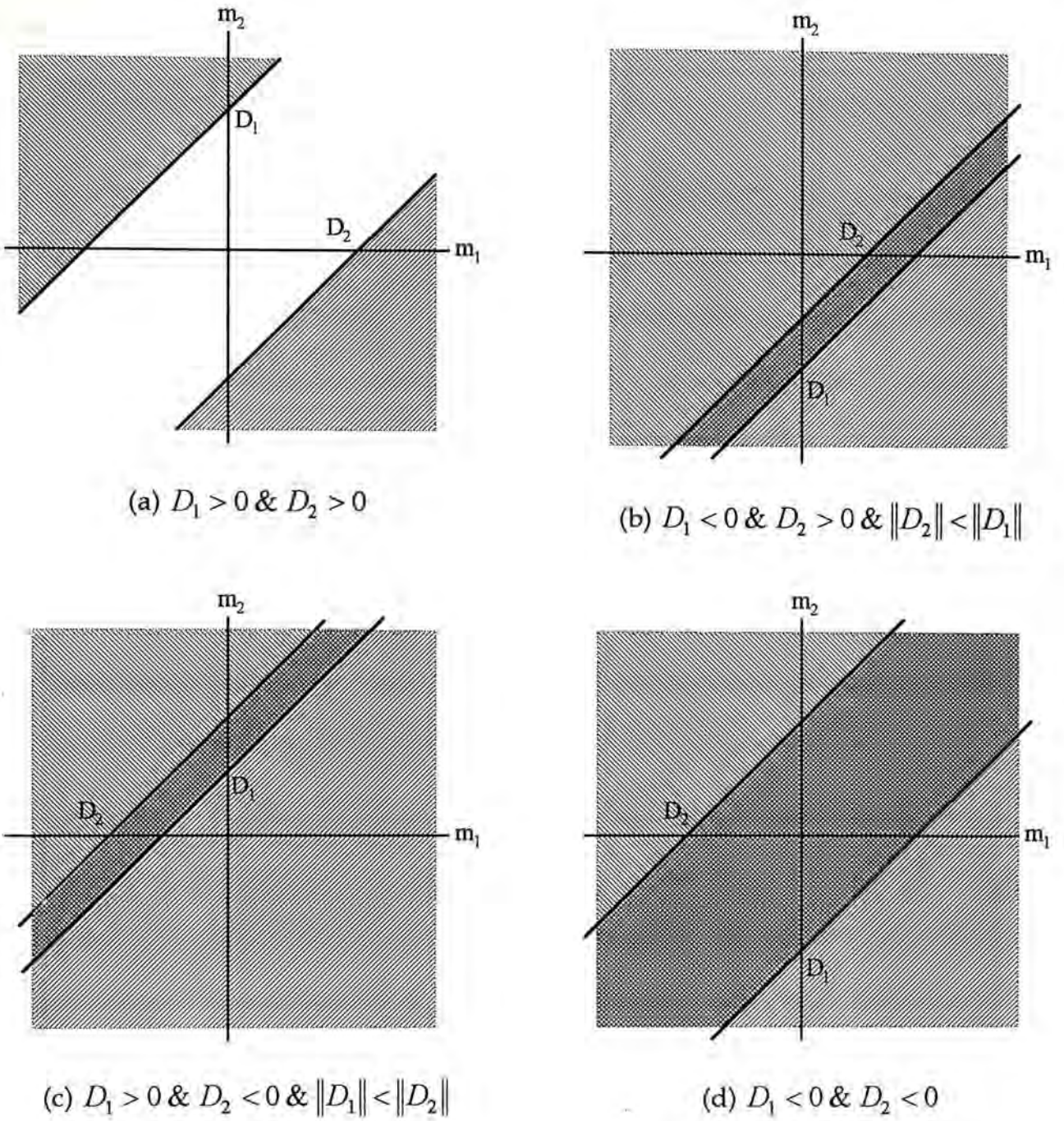



Fig 6.1 Possible solutions of the inequalities Eq(6.7) and Eq(6.8)

where  represents the solution space of Eq(6.7),

 for Eq(6.8), and  for both Eq(6.7) and Eq(6.8).

6.2 Concluding Remarks

A method for locating empirical landmarks in inferred Markov networks, which requires no *a priori* information on the segmentation of the input strings, has been introduced. The method is based on the property of the Markov network inference which retains the landmark substrings of the input strings. This property is important for the inference since the probability of an input string generated by the modified network (the output of the string-to-network alignment) is proportional to the number of landmark substrings (in any length) found in the input string. In other words, the Markov network inference can be interpreted as a string segmentation process. In fact, one of the applications of the forced landmark model is to estimate the centromere position of human chromosomes [10].

Analysis on the distributions of the empirical landmarks of human chromosomes have been carried out. It has been shown that the distributions of most relative landmark positions are normal. Therefore, the mean of the relative landmark positions is a proper estimation. In compare with the distributions of centromere position of 22 human chromosome types, half of it can be estimated by an empirical landmark. Experiments on chromosome classification have shown that the discrimination power of Markov network inference with empirical landmarks is similar to that of inherited landmarks.

Manipulations of empty states in the inferred Markov networks are the crucial part of the dynamic programming inference. Network modifications are directed by the result and the *cost function* of the string-to-network alignment explicitly. The structure of the cost function for the string-to-network alignment have been illustrated and explained extensively.

In last chapter, the inferred Markov networks have been employed for speech recognition. Extensive experiments on recognizing phonemes in the TIMIT

database and isolated speech in a Cantonese speech corpus have been conducted. The result on phoneme recognition is better than the one using HMM with single codebook. With the application of empirical landmarks, half of the computations can be saved without sacrificed much of classification power. Since the size of the Cantonese speech corpus is relatively small, only tone independent classification can be tested. The result are encouraging despite no strong sequential nature in the speech data (phonemes, especially) can be found by the primitive feature extractor.

Appendix A

Classification experiments on a Cantonese speech corpus using inferred Markov networks are described in this appendix. The aims of these experiments is to provide a reference case for the feature selection and classification of isolated speech using inferred Markov networks where the sequential nature of the feature patterns are supposed to be strong.

A.1 Cantonese Speech Corpus

The Cantonese speech corpus used in the experiments was originally created for the recognition of tone in Cantonese syllables [4]. A Cantonese syllables may consist of three phonemes, which are *Initial Consonant (IC)*, *Middle Vowel* and *Final Consonant (FC)* as shown in Fig A.1.

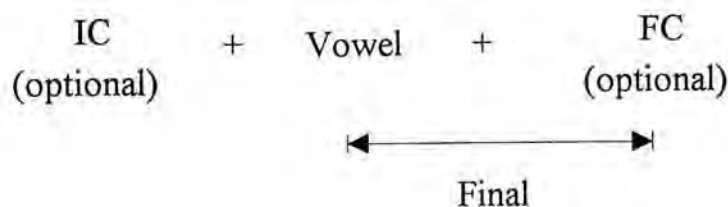


Fig A.1 Components of a Cantonese Syllable.

The *final* of a Cantonese syllable is defined to be the concatenation of the middle vowel and the final consonant. According to the reference [41], there are 36 non-entering tone finals and 17 entering tone finals where each non-entering tone finals can be pronounced in 6 different tones while each entering tone finals in 3 different tones. So there are 267 different syllables $(36 \times 3 + 17 \times 3)$ with which 234

syllables are valid¹. For each syllable, a valid Cantonese syllable was selected which is the concatenation of a initial consonant and the syllable itself.

The Cantonese speech corpus is a collection of such selected Cantonese syllables which were pronounced in isolated form by a group of 20 native Cantonese speakers aged around 20 (10 male and 10 female). Each speaker was asked to pronounce each syllable three times. So there are total 14040 speech samples. The speech signals were bandpass filtered with a passband from 100 Hz to 4.3 KHz and were digitized by a 12-bits linear A/D converter at 10 KHz sampling rate.

For each Cantonese syllable, there are 60 samples (3 times from 20 speakers). This is a relatively small sample set for the classification experiments with inferred Markov networks. Therefore, the speaker independent, tone independent syllable groups were selected. Each group is a collection of identical syllables regardless of tones. Six Cantonese syllable groups were selected as shown in Table A.1. In each group, 100 samples were extracted which consists of 50 training samples and 50 testing samples.

Group	IPA symbol	Mnemonic	Sample in tone 1	Available tones
1	/a/	ah	啊	1, 3, 5
2	/fu/	fu	夫	3, 4, 5
3	/hau/	hau	敲	1, 3, 5
4	/ji/	ji	衣	1, 2, 3, 4, 5, 6
5	/jyn/	jyn	淵	1, 2, 3, 4, 5, 6
6	/əm/	rm	庵	1, 3, 5

Table A.1 Six Cantonese syllable groups.

¹A valid syllable here is a syllable that can produced valid Cantonese syllables with the concatenations of some initial consonants.

A.2 Feature Extraction

The feature extraction process is the same process as described in Section 5.3. For the Cantonese speech corpus, the codebook sizes 8 and 16 were tested and the size 8 codebook turned out to be a better choice. It is reasonable for more sequential nature can be exhibited with less symbols in quantization. For the case of chromosome classification, 5 to 7 symbols ('c', 'b', 'a', '=', 'A', 'B', 'C') were active. Therefore, size 8 codebook seems to be a reasonable choice.

A.3 Classification Results

Two classification experiments have been conducted. The first experiment tests the capability of the syllable networks with respect to the training set itself while the second experiment deals with the testing set. Again, the aligned probabilities were normalized with the MRP of the tested networks.

As shown in Table A.2, the average correct rate for the training sets is 79.67%, while the correct rate for the testing set is 64.67%. The syllable group "jyn" turns out to be the worst among the six groups in both experiments. Part of the reasons of the bad performance is that the inferred Markov network for this group is the biggest one (such network have 1316 states). Such phenomenon indicates that the MRP normalization method should be reconsidered. From the confusion matrix shown in Tables A.3 and A.4, "jyn" is often confused with "ji" and getting worse when the testing set is examined in experiment 2.

Group	ah	fu	hau	ji	jyn	rm	average
Train set	82%	88%	84%	78%	68%	78%	79.67%
Test set	70%	78%	62%	62%	52%	64%	64.67%

Table A.2 Classification result of the Cantonese syllables.

Training \ Testing	ah	fu	hau	ji	jyn	rm
ah	82	0	2	0	14	2
fu	4	88	0	2	0	6
hau	2	2	84	0	10	2
ji	4	8	0	78	8	2
jyn	0	14	0	18	66	2
rm	8	2	8	0	4	78

Table A.3 Confusion matrix of classification of training set.

Training \ Testing	ah	fu	hau	ji	jyn	rm
ah	70	0	2	4	14	10
fu	2	78	0	10	2	8
hau	20	0	62	0	12	6
ji	0	24	0	62	14	0
jyn	0	12	0	32	52	4
rm	16	4	10	0	6	64

Table A.4 Confusion matrix of classification of testing set.

Reference

- [1] Abramson, N. *Information Theory and Coding*. McGraw-Hill, New York, 1963.
- [2] Ash, R. *Information Theory*. Interscience, New York, 1965.
- [3] Baase, S. *Computer Algorithms: Introduction to Design and Analysis*. Addison-Wesley, Reading, Mass., 1988.
- [4] Cheng, Y.H. An Efficient Tone Classifier for Speech Recognition of Cantonese. MPhil. Thesis, The Chinese University of Hong Kong, 1991.
- [5] Granum, E., Gerdes, T., and Lundsteen, C. "Simple weighted density distributes, WDDs, for discrimination between G-banded chromosome." In *Proceedings of IV European Chromosome Annual Workshop*, Edinburgh, 1981.
- [6] Granum, E., Thomason, M.G., and Gregor, J. "On the use of automatically inferred Markov networks for chromosome analysis." In *Automation of Cytogenetics*. Springer-Verlag, Berlin, 1989.
- [7] Granum, E. and Thomason, M.G. Automatically inferred Markov network models for classification of chromosomal band pattern structures. *Cytometry* 11, 1 (1990), 26-39.
- [8] Granlund, G.H. Identification of human chromosomes by using integrated density profiles. *IEEE Trans. on Biomedical Engineering* 23, 3 (May 1976), 182-192.
- [9] Granum, E. "Applications of statistical and syntactical methods of analysis and classification to chromosome data." In *Pattern Recognition Theory and Application*. D. Reidel, Dordrecht, 1981.

- [10] Gregor, J. and Granum, E. String segmentation and classification by forced landmark Markov networks. *International Journal of Pattern Recognition and Artificial Intelligence* 5, 3(1991), 413-423.
- [11] Gregor, J. and Granum, E. Finding chromosome centromeres using band pattern information. *Comp. Biol. Med.* 21, (1991), 55-67.
- [12] Groen, F.C.A., Kate, T.K.T., Smeulders, A.W.M., and Young, I.T. *Human chromosome classification based on local band descriptors*. *Pattern Recognition Letters* 9, (April 1989), 221-222.
- [13] Ji, L., Piper, J., and Tang, J.Y. Erosion and dilation of binary images by arbitrary structuring elements using interval coding. *Pattern Recognition Letters* 9, (Apr 1989), 201-209.
- [14] Ji, L. Intelligent splitting in the chromosome domain. *Pattern Recognition* 22, 5 (1989), 519-532.
- [15] Kramer, H.P. and Bruckner, J.B. Iteration of a non-linear transformation for enhancement of digital images. *Pattern Recognition* 7, (1975), 53-58.
- [16] Landau, G.M., Vishkin, U., and Nussinov, R. "Fast alignment of DNA and protein sequences." In *Methods in Enzymology vol 183*. Academic Press, San Diego, 1990.
- [17] Lee, K.F. and Hon, H.W. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. on Acoustics, Speech, and Signal Processing* 37, 11 (1989), 1641-1648.
- [18] Lundsteen, C., Gerdes, T., and Maahr, J. Automatic classification of chromosomea as part of a routine system for clinical analysis. *Cytometry* 7, (1986), 1-7.
- [19] Lundsteen, C. and Granum, E. Description of chromosome banding patterns by band transition sequences. *Clinical Genetics* 15, (1979), 418-429.

- [20] Lundsteen, C. and Granum, E. Visual classification of banded human chromosomes. III. Classification and karyotyping of density profiles described by band transition dequences. *Clinical Genetics* 15, (1979), 430-439.
- [21] Lundsteen, C., Philip, J., and Granum, E. Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromsomes. *Clinical Genetics* 18, (1980), 355-370.
- [22] Lundsteen, C. Aspects of automated chromosome analysis. *Danish Medical Bulletin* 27, 1 (Feb 1980), 1-21.
- [23] Mansuripur, M. *Introduction to Information Theory*. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [24] Medhi, J. *Stochastic Processes*. Wiley Eastern Limited, New Delhi, 1982.
- [25] Paton, K. Automatic chromosome identification by the maximum-likelihood method. *Annual Human Genetic* 33, (1969), 177-184.
- [26] Piper, J., Granum, E., Rutovitz, D., and Ruttledge, H. Automation of chromosome analysis. *Signal Processing* 2, (1980), 203-221.
- [27] Piper, J. and Granum, E. On fully automatic frature measurement for banded chromosome classification. *Cytometry* 10, (1989), 242-255.
- [28] Piper, J. Efficient implementation of skeletonisatrion using interval coding. *Pattern Recognition Letters* 3, (1985), 389-397.
- [29] Piper, J. Classification of chromosomes constrained by expected class size. *Pattern Recognition Letters* 4, (Oct 1986), 391-395.
- [30] Piper, J. The effect of zero feature correlation assumption on maximum likelihood based classification of chromosomes. *Signal Processing* 12, (1987), 49-57.
- [31] Rosenfeld, A. The fuzzy geometry of image subsets. *Pattern Recognition Letters* 2, (1984), 311-317.

- [32] Saito, S. and Nakata, K. *Fundamentals of Speech Signal Processing*. Academic Press, Tokyo, 1985.
- [33] Therrien, C.W. *Decision estimation and classification : an introduction to pattern recognition and related topics*. John Wiley & Sons, New York, 1989.
- [34] Thomason, M.G. and Granum, E. Dynamic programming inference of Markov networks from finite sets of sample strings. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 8, 4(Jul 1986), 491-501.
- [35] Tso, M.K.S. and Graham, J. The transportation algorithm as an aid to chromosome classification. *Pattern Recognition Letters* 1, (Jul 1983), 489-496.
- [36] Tso, M., Kleinschmidt, P., Mitterreiter, I., and Graham, J. An efficient transportation algorithm for automatic chromosome karyotyping. *Pattern Recognition Letters* 12, (Feb 1991), 117-126.
- [37] Vanderheydt, L., Dom, F., Oosterlinck, A., and Van Den Berghe, H. Two-dimensional shape decomposition using fuzzy subset theory applied to automated chromosome analysis. *Pattern Recognition* 13, 2 (1981), 147-157.
- [38] Vanderheydt, L., Oosterlinck, A., and Van Den Berghe, H. Design of a special interpreter for the classification of human chromosomes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 1, 2 (Apr 1979), 214-219.
- [39] Vanderheydt, L., Oosterlinck, A., Van Daele, J., and Van Den Berghe, Design of a graph-representation and a fuzzy-classifier for human chromosome. *Pattern Recognition* 12, (1980), 201-210.
- [40] Wagner, R.A. and Fischer, M.J. The string-to-string correction problem. *JACM*. 21, 1(Jan 1974), 168-173.
- [41] Wong, S.L. *A Chinese Syllabary Pronounced According to the Dialect of Canton*. Chung Wah, Hong Kong, 1941.



CUHK Libraries



000388963