CHAPTER 1

Introduction and Background

1.1 WHAT IS DISCRIMINATIVE LEARNING?

Discriminative learning is one of two major paradigms in constructing probabilistic pattern classifiers and recognizers, where classifiers usually deal with nonsequential data (i.e., with fixed-dimension input features) and the classification target is one of a limited set of categories, whereas recognizers handle sequential data (i.e., with variable-dimension input features) and the recognition target is an open output that can be of variable length. The other major paradigm is generative modeling and learning, which establishes and learns a model of the joint probability of the features and the class identity. In contrast, discriminative methods either directly model the class posterior probability, or learn the parameters of the joint-probability model discriminatively so as to minimize classification/recognition errors.

The main purpose of this book is to present an extensive account on the basic ideas behind the approaches and techniques on discriminative learning, especially those that discriminatively learn the parameters of joint-probability models [e.g., hidden Markov models (HMMs)]. In addition, we also desire to position our treatment of the related algorithms in a wider context of learning and building statistical classifiers/recognizers from a more general context of machine learning. The Bayes decision theory serves as the basic formalism of the classification and recognition processes for achieving the optimal decision boundaries. Hence, the goal of pattern recognition can be described as finding the parameters of the classifiers or recognizers that minimize the error rate by using the available training samples. Within the two main paradigms for designing and learning statistical classifiers/recognizers, the generative ones use the joint-probability model to perform the decision-making task based on the posterior probability of the class computed by Bayes rule [11, 43, 57]. The standard approach to learning (i.e., estimating) a generative model is maximum likelihood (ML). ML learning is considered a nondiscriminative approach because it aims at modeling the data distribution instead of directly separating class categories.

On the other hand, the discriminative classifiers/recognizers typically bypass the stage of building the joint-probability model while directly using the class posterior probability. This is exemplified by the celebrated argument that "one should solve the (classification/recognition) problem directly and never solve a more general problem as an intermediate step" [48]. This recognizer design philosophy is the basis of a wide range of popular machine learning methods including support vector machine [48], conditional random field [28, 37], and maximum entropy Markov models [15, 30], etc., where the "intermediate step" of estimating the joint distribution has been avoided. For example, in the recently proposed structured classification approach [15, 28, 30, 37] in machine learning and speech recognition, some well-known deficiencies of the HMM are addressed by applying "direct" discriminative learning, replacing the need for a probabilistic generative model by a set of flexibly selected, overlapping "features." Because the conditioning is made on the feature sequence and the "features" can be designed with long-contextual-span properties, the conditional-independence assumption made in the HMM is conceptually alleviated — provided that proper "features" can be constructed. How to design such features is a challenging research direction and it becomes a critical factor for the potential success of the structured discriminative approach, which departs from the "generative" component or joint distribution. On the other hand, local features can be much more easily designed that are appropriate for the generative approach, and many effective local features have been well established (e.g., cepstra, filter-bank outputs, etc. [10, 43]). Despite the high complexity of estimating joint distributions when the sole purpose is discrimination, the generative approach has important advantages of facilitating knowledge incorporation and of conceptually straightforward analyses of classifier/recognizer components and their interactions.

Analyses of the capabilities and limitations associated with the two general machine learning paradigms discussed above lead to a practical pattern recognition framework that will be pursued in this book. That is, we attempt to establish a simplistic joint-distribution or generative model, with the complexity lower than what is required to accurately "generate" samples from the true distribution. To make such low-complexity generative models discriminate well, it requires parameter learning methods that are discriminative in nature so as to overcome the limitations in the simplistic model structures. This is in contrast to the generative approach of fitting the intraclass data as the conventional ML-based methods intend to accomplish. This type of practical framework has been applied to and guiding much of the recent work in speech recognition research, where HMMs are used as the low-complexity joint distribution for the local acoustic feature sequences of speech and the corresponding underlying linguistic label sequences (sentences, words, or phones, etc.).

1.2 WHAT IS SPEECH RECOGNITION?

Speech recognition is the process and the related technology for converting a speech signal into a sequence of words (or other linguistic units) by means of an algorithm implemented as a computer program. Speech recognition applications that have emerged over the last few years include voice dialing, call routing, interactive voice response, voice search, data entry and dictation, command and control (voice user interface with the computer), hands-free computing (automotive applications), structured document creation (e.g., medical and legal transcriptions), appliance control by voice, computer-aided language learning, content-based spoken audio search, and robotics.

Modern general-purpose speech recognition systems are generally based on HMMs, which will be described in some detail in Chapter 2. One reason why HMMs are popular in speech recognition is that their parameters can be trained or learned automatically and they are simple and computationally feasible to use. In speech recognition, to give the very simplest setup possible, HMMs generate a sequence of multidimensional real-valued or symbolic/discrete acoustic features, each corresponding to about 10 ms. The real-valued vectors (or the discrete symbols) often consist of cepstral coefficients (or their vector-quantized codes), which are obtained by taking a Fourier transform of a short-time window of speech and decorrelating the spectrum by using a cosine transform. The continuous-density (CD) HMMs usually have, in each state, a probability distribution of a mixture of diagonal-covariance Gaussians. Discrete HMMs usually have, in each state, a nonparametric discrete distribution. Each word or phone will have different output distributions that are trained or learned automatically. An HMM for a sequence of words or phonemes is constructed by concatenating the individual trained HMMs for the separate words and phonemes.

Major developments in the technology of speech recognition over the past 50 years have been elegantly summarized in a recent keynote presentation at International Conference on Acoustics, Speech, and Signal Processing; the slides of that presentation can be found in http://www.ewh.ieee.org/soc/sps/stc/News/NL0704/furui-icassp2007.pdf. This long period has witnessed the field of speech recognition proceed from its infancy to its current coming of age. Although far from a "solved" problem, it now has a growing number of practical applications in many sectors. Further research and development will enable increasingly more powerful systems, deployable on a worldwide basis.

Let us summarize the major developments of speech recognition in four areas. First, in the infrastructure area, Moore's law, in conjunction with the constantly shrinking cost of memory, has been instrumental in enabling speech recognition researchers to develop and run increasingly complex systems. The availability of common speech corpora for speech system training, development, and evaluation, has been critical in creating systems of increasing capabilities. Speech is a highly variable signal, characterized by many factors, and thus large corpora are critical in modeling it well enough for automated systems to achieve proficiency. Over the years, these corpora have been created, annotated, and distributed to the worldwide community. The character of the recorded speech has progressed from limited, constrained speech materials to masses of progressively more realistic, spontaneous, and "found" speech. The development and adoption of rigorous benchmark evaluations and standards have also been critical in developing increasingly powerful and capable speech recognition systems.

Second, in the area of knowledge representation, major advances in speech signal representations have included perceptually motivated acoustic features of speech. Architecturally, the most important development has been the searchable unified graph representations allowing multiple sources of knowledge to be incorporated in a common probabilistic framework.

Third, in the area of modeling and algorithms, the most significant paradigm shift has been the introduction of statistical methods, especially of the HMM method. More than 30 years after the initial use of HMMs in 1970s, this methodology still predominates. The ML-based expectation–maximization (EM) algorithm and the forward–backward or Baum–Welch algorithm have been the principal means by which the HMMs are trained from data. Despite their simplicity, *N*-gram language models have proved remarkably powerful and resilient. Decision trees have been widely used to categorize sets of features, such as pronunciations from training data. Statistical discriminative learning techniques form the recent major innovations in speech recognition algorithms, which will be elaborated below and be the focus of the remainder of this book.

Fourth, in the area of recognition hypothesis search, key decoding or search strategies, originally developed in nonspeech applications, have focused on stack decoding (A^* search), Viterbi, *N*-best, and lattice search/decoding. Derived originally from communications and information theory, stack decoding was subsequently applied to speech recognition systems. Viterbi or dynamic-programming based search is at present broadly applied to search alternative recognition hypotheses in virtually all modern speech recognition systems.

1.3 ROLES OF DISCRIMINATIVE LEARNING IN SPEECH RECOGNITION

As we just highlighted above, statistical discriminative learning has become a major theme in recent speech recognition research (e.g., [8, 9, 12, 18, 25, 31, 36, 37, 40, 42]). In particular, much of the striking progress in large-scale automatic speech recognition over the past few years has been attributed to the successful development and applications of discriminative learning (e.g., [31, 33, 40, 41]). Although the ML-based learning algorithm (i.e., the Baum–Welch algorithm) has been highly efficient and practical, it limits the performance of speech recognition. This is because ML learning relies on the assumption that the correct functional form of the joint probability between the data and the class categories is known and that there are sufficient and representative training data, both of which are often not realistic in practice. In the case of speech recognition, the data are speech feature sequences and the class categories are word sequences. As we discussed earlier, the currently most popular functional form of the probability model for speech is the HMM. Given the knowledge gained from many years of research in speech science, the assumptions made by the HMM are in many ways incorrect for the realistic processes in human speech. This inconsistency motivates the development of discriminative learning methods for speech recognition and highlights their critical roles in improving speech recognition performance beyond the conventional ML-based learning techniques. The essence of discriminative learning as presented in this book is to learn the parameters of distribution models (e.g., HMMs) in such a way that the recognition errors or some measures of them are minimized directly via efficient and effective optimization techniques.

Two central issues in the development of discriminative learning methods for sequential pattern recognition and in particular for speech recognition are: (1) construction of the objective functions for optimization and (2) actual optimization techniques. There have been a wide variety of methods reported in the literature related to both of these issues (e.g., [8, 14, 18, 25, 31, 33, 34, 38, 42, 44, 46, 49]); however, their relationships have not been adequately understood. Because of the practical and theoretical importance of this problem, there is a pressing need for a unified account of the numerous discriminative learning techniques in the literature. This book aims to fulfill this need while providing insights into the discriminative learning framework for sequential pattern classification and for speech recognition. In presenting discriminative learning in this chapter, we intend to address the issues of how the various discriminative learning techniques are related to and distinguished from each other, and what may be a deeper underlying scheme that can unify various ostensibly different techniques. Although the unifying review provided in this book is on a general class of pattern recognition problems associated with sequential characteristics, we will focus most of the discussions on those related to speech recognition and to the HMM [10, 43, 47]. We note that the HMM as well as the various forms of discriminative learning have been used in many signal processing areas beyond speech; for example, in bioinformatics [5, 13], in text and image classification/recognition [29, 53, 56], in video object classification [54], in natural language processing [7, 9], and in telerobotics [55]. It is our hope that the unifying review and the insights provided in this book will foster more principled and successful applications of discriminative learning in a wide range of signal processing disciplines, speech processing or otherwise.

1.4 BACKGROUND: BASIC PROBABILITY DISTRIBUTIONS

In this section, we provide the mathematical background for several basic probability distributions that will be used directly or as building blocks for more complex distributions in the remaining chapters of this book. The basic probability distributions discussed first will include multinomial distribution (discrete), Gaussian, and mixture-of-Gaussian distributions (continuous). Then we will present a more general form of the distributions, exponential-family distributions, which subsume a large number of discrete and continuous distributions. The more complex distributions (e.g., HMMs) built from the basic distributions will be presented in subsequent chapters.

1.4.1 Multinomial Distribution

Frequently, we need to handle discrete random variables that may take one of K possible values. Among all possible ways to express such variables, there is a convenient representation that the variable is represented by a K-dimensional vector x in which one of the elements $x(k)$ is equal to 1, and all other elements are equal to 0. For example, if we have a variable that can take $K = 6$ possible values and a particular observation of the variable happens to correspond to the third value, that is, $x(3) = 1$, then x can be represented by

$$x = [\ 0, 0, 1, 0, 0, 0\ ]^{\mathrm{T}}$$

If we denote the probability of $x(k) = 1$ by the parameter v_k, then the distribution of x is given by

$$p(x|v) = \prod_{k=1}^{K} v_k^{x(k)} \tag{1.1}$$

where $v = [v_1, \ldots, v_K]^{\mathrm{T}}$ is the parameter vector. Because v_k is a probability, that is, it is the probability of that the random variable takes the kth value, $\{v_k\}$ are constrained to satisfy $v_k \geq 0$ and $\sum_k v_k = 1$.

Now consider a data set X of N independent observations $x_1, \ldots, x_N$. If we denote the counts of observations at state k by the value m_k, we can have the joint distribution of the quantities $m_1, \ldots, m_K$, conditioned on the parameter vector v, and the total number of observations N, which takes the following form:

$$p(m_1, \ldots, m_K | v, N) = \binom{N}{m_1, \ldots, m_K} \prod_{k=1}^{K} v_k^{m_k} \tag{1.2}$$

This is known as the multinomial distribution. The normalization coefficient is the number of ways of partitioning N objects into K groups of size $m_1, \ldots, m_K$ and is computed as

$$\binom{N}{m_1, \ldots, m_K} = \frac{N!}{m_1! m_2! \ldots m_K!}$$

where the variables m_k are subject to the constraint

$$m_k \geq 0 \quad \text{and} \quad \sum_k m_k = N.$$

Note that (1.1) is a special case of the multinomial distribution for a single observation; that is, $N = 1$.

1.4.2 Gaussian and Mixture-of-Gaussian Distributions

The Gaussian or normal distribution is a widely used model for the distribution of continuous variables. When the random variable x is a scalar, the Gaussian probability density function (PDF) is

$$p(x|\lambda) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right\} = N(x;\mu,\sigma^2) \tag{1.3}$$

where the parameter set λ includes μ (mean) and σ (standard deviation). For a D-dimensional vector x, the multivariate Gaussian PDF takes the form of

$$p(x|\lambda) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right\} = N(x;\mu,\Sigma) \tag{1.4}$$

where $\lambda = \{\mu, \Sigma\}$ includes μ (mean vector) and Σ (covariance matrix). The Gaussian distribution is commonly used in many engineering and science disciplines including speech processing. The popularity arises not only from its highly desirable computational properties, but also from its ability to approximate many naturally occurring real-world data due to the central limit theorem.

Mixture-of-Gaussian distributions. Unfortunately, in some speech processing problems including speech recognition, the Gaussian distribution is inadequate. The inadequacy comes from its unimodal property, whereas most speech features have multimodal distributions. The more appropriate distribution is the following mixture-of-Gaussian distribution with the desirable multimodal property:

$$p(x|\lambda) = \sum_{m=1}^{M} c_m N(x;\mu_m,\sigma_m^2) \tag{1.5}$$

where the variable x is a scale and $\lambda = \{c_m, \mu_m, \sigma_m; m = 1, 2 \ldots, M\}$ or

$$p(x|\lambda) = \sum_{m=1}^{M} c_m N(x;\ \mu_m,\Sigma_m)$$

where the variable x is a vector and $\lambda = \{c_m, \mu_m, \Sigma_m; m = 1, 2 \ldots, M\}$.

1.4.3 Exponential-Family Distribution

Both multinomial and Gaussian distributions (but not the mixture-of-Gaussians) discussed above are special cases of a broad class of distributions known as the exponential family, including both continuous and discrete distributions. This general family of distributions is defined by the following PDF:

$$p\,(x|\theta) = h(x)\cdot\exp\left\{\theta^{\mathrm{T}}T(x) - A(\theta)\right\} \tag{1.6}$$

where x can be scalar or vector, and may be discrete or continuous. Here, θ is called the natural parameters of the distribution, $T(x)$ is some function of x, $A(\theta)$ is the cumulative generating function, and $h(x)$ is the base measure, which is a function of x. To obtain a normalized distribution, we need to take integration of both sides of (1.6) and set it to one:

$$\int p\,(x|\theta)\mathrm{d}x = \exp\left(-A(\theta)\right)\int h(x)\cdot\exp\left(\theta^{\mathrm{T}}T(x)\right)\mathrm{d}x = 1 \tag{1.7}$$

Therefore,

$$\exp\left(A(\theta)\right) = \int h(x)\cdot\exp\left(\theta^{\mathrm{T}}T(x)\right)\mathrm{d}x \tag{1.8}$$

For a discrete random variable x, the integration above should be replaced by summation.

Convexity of the exponential-family distribution. Let us first consider the properties of $A(\theta)$. Examine the first-order derivative of $A(\theta)$. Taking the gradient of both side of (1.8) with respect to θ, we have

$$\exp[A(\theta)]\cdot\nabla A(\theta) = \int h(x)\cdot\exp\left[\theta^{\mathrm{T}}T(x)\right]\cdot T(x)\mathrm{d}x$$

Q1 Rearranging and making use of (1.6), we obtain

$$\nabla A(\theta) = \exp[-A(\theta)]\cdot\int h(x)\cdot\exp\left[\theta^{\mathrm{T}}T(x)\right]\cdot T(x)\mathrm{d}x = \mathbb{E}_{p(x|\theta)}[T(x)] \tag{1.9}$$

After using the chain rule and the matrix derivative formula of $\nabla(f(\theta)\cdot a) = \nabla(f(\theta)\cdot a^{\mathrm{T}}$, the second-order derivative of $A(\theta)$ can be obtained based on (1.9):

$$\begin{aligned}\nabla^2 A(\theta) = & -\exp[-A(\theta)]\cdot\nabla A(\theta)\cdot\int h(x)\cdot\exp\left[\theta^{\mathrm{T}}T(x)\right]\cdot T(x)^{\mathrm{T}}\mathrm{d}x\\ & +\exp[-A(\theta)]\cdot\int h(x)\cdot\exp\left[\theta^{\mathrm{T}}T(x)\right]\cdot T(x)\cdot T(x)^{\mathrm{T}}\mathrm{d}x\end{aligned}$$

Using (1.9) again, we have

$$\nabla^2 A(\theta) = -\mathbb{E}_{p(x|\theta)}[T(x)]\mathbb{E}_{p(x|\theta)}[T(x)]^{\mathrm{T}} + \mathbb{E}_{p(x|\theta)}\left[T(x)T(x)^{\mathrm{T}}\right] = \mathrm{Cov}_{p(x|\theta)}[T(x)] \succeq 0$$

That is, the second-order derivative of $A(\theta)$ is positive definite. Therefore, $A(\theta)$ is a convex function of θ.

Maximum likelihood estimation and sufficient statistic of the exponential-family distribution. Now, let us consider the problem of estimating the parameter vector $\boldsymbol{\theta}$ in the general exponential-family distribution (1.6) using the technique of maximum likelihood (ML). In maximum likelihood estimation, consider that there is a set of independent identically distributed data denoted by $\mathbf{X} = \{x_1, \ldots, x_n\}$, for which the likelihood function is given by

$$p(X|\theta) = \left(\prod_{n-1}^{N} h(x_n)\right) \cdot \exp\left[\theta^{\mathrm{T}} \sum_{n=1}^{N} T(x_n) - N \cdot A(\theta)\right]$$

Setting the gradient of $\ln(p\,(X|\theta))$ with respect to θ to zero, we obtain the following condition to be satisfied by the maximum likelihood estimate

$$\nabla A(\theta) = \frac{1}{N}\sum_{n=1}^{N} T(x_n) \qquad (1.10)$$

which can be solved to obtain θ_{ML}. Because $A(\theta)$ is convex, there is one global unique ML solution for θ_{ML}.

From (1.10), we observe that the solution to the ML estimate depends on the data only through $\sum_{n=1}^{N} T\,(x_n)$, which is therefore called sufficient statistic of the distribution (1.6). In computing the ML estimate, we only need to store the value of the sufficient statistic.

The above sufficiency property holds for discriminative learning, and we will defer the discussion to Chapter 4.

Exponential form of the multinomial distribution. It can be verified that the distributions discussed in the previous sections are members of the exponential family.

Let us consider the multinomial distribution that, for a single observation x, takes the form

$$p(x|v) = \prod_{k=1}^{K} v_k^{x(k)} \qquad (1.11)$$

There are parameter constraints of $v_k \geq 0$ and $\sum_{k=1}^{K} v_k = 1$ in this form of distribution, which is to be removed. Due to the sum-to-one constraint, there are a total of $K - 1$ free parameters. For instance, v_K can be expressed by the remaining $K - 1$ parameters through $\nu_k = 1 - \sum_{j=1}^{K} v_j$, thus leaving $K - 1$ free parameters. Note that these remaining $K - 1$ parameters are still subject to the constraints $v_{\mathrm{k}} \geq 0$ and $\sum_{j=1}^{K-1} v_j \leq 1$, $k = 1, \ldots, K - 1$. Note also that $\sum_{k=1}^{K} x(k) = 1$. We now rewrite the distribution

$$\begin{aligned}
\prod_{k=1}^{K} v_k^{x(k)} &= \exp\left\{\sum_{k=1}^{K} x(k)\ln v_k\right\} \\
&= \exp\left\{\sum_{k=1}^{K-1} x(k)\ln v_k + \left(1-\sum_{k=1}^{K-1} x(k)\right)\ln\left(1-\sum_{j=1}^{K-1} v_j\right)\right\} \\
&= \exp\left\{\underbrace{\sum_{k=1}^{K-1} x(k)\ln\left(\frac{v_k}{1-\sum_{j=1}^{K-1} v_j}\right)}_{\theta^{\mathrm{T}}T(x)} \underbrace{+\ln\left(1-\sum_{j=1}^{K-1} v_j\right)}_{-A(\theta)}\right\}
\end{aligned}$$

and then construct the $K-1$ dimensional natural parameter vector $\theta = [\theta_1,\ldots,\theta_{k-1}]^{\mathrm{T}}$ such that

$$\theta_k = \ln\left(\frac{v_k}{1-\sum_{j=1}^{K-1} v_j}\right) \tag{1.12}$$

After identifying the parameters (as well as the sufficient statistic) above in the standard form, we now need to express $A(\theta)$ in terms of the parameters of (1.12). To do this, we rewrite (1.12) as

$$\exp(\theta_k) = \frac{v_k}{1-\sum_{j=1}^{K-1} v_j} \tag{1.13}$$

Summing both sides of (1.13) over k, we have

$$\sum_{k=1}^{K-1} \exp(\theta_k) = \frac{\sum_{k=1}^{K-1} v_k}{1-\sum_{j=1}^{K-1} v_j}$$

After adding one on both sides, we obtain

$$1+\sum_{k=1}^{K-1} \exp(\theta_k) = \frac{1}{1-\sum_{j=1}^{K-1} v_j} \tag{1.14}$$

Then, substituting the right-hand side of (1.14) into (1.13), we have

$$v_k = \frac{\exp(\theta_k)}{1 + \sum_{j=1}^{K-1} \exp(\theta_j)} \quad \text{and} \tag{1.15}$$

$$\ln\left(1 - \sum_{j=1}^{K-1} v_j\right) = \ln\left(1 - \frac{\sum_{k=1}^{K-1} \exp(\theta_k)}{1 + \sum_{j=1}^{K-1} \exp(\theta_j)}\right) = -\ln\left(1 + \sum_{j=1}^{K-1} \exp(\theta_j)\right)$$

Therefore, comparing with the standard form (1.6) of the exponential family distribution, we identify:

$$\begin{aligned} h(x) &= 1 \\ T(x) &= \tilde{x} = [x_1, \ldots, x_{K-1}]^{\mathrm{T}} \\ A(\theta) &= \ln\left(1 + \sum_{j=1}^{K-1} \exp(\theta_j)\right) \end{aligned}$$

where $\tilde{x}$ is an observation vector that only contains the first $K-1$ elements of x. Furthermore,

$$\frac{\partial A(\theta)}{\partial \theta} = \frac{1}{1 + \sum_{j=1}^{K-1} \exp(\theta_j)} \begin{bmatrix} \exp(\theta_1) \\ \vdots \\ \exp(\theta_{K-1}) \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_{K-1} \end{bmatrix} = \tilde{v} \tag{1.16}$$

where we denote by $\tilde{v}$ the partial parameter vector that only contains the first $K-1$ parameters.

We now discuss ML parameter estimation. According to (1.10), the maximum likelihood estimation of θ should satisfy the following condition:

$$\tilde{v} = \frac{1}{N} \sum_{n=1}^{N} \tilde{x}_n \tag{1.17}$$

By summing both sides of (1.17) over $k = 1, \ldots, K-1$, we have

$$\sum_{k=1}^{K-1} v_k = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K-1} x_n(k)$$

Therefore, we have

$$v_K = 1 - \sum_{k=1}^{K-1} v_k = 1 - \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K-1} x_n(k) = \frac{1}{N}\sum_{n=1}^{N}\left(1 - \sum_{k=1}^{K-1} x_n(k)\right) = \frac{1}{N}\sum_{n=1}^{N} x_n(K) \qquad \textit{(1.18)}$$

Combining (1.17) and (1.18), we have the ML estimation formula for the multinomial distribution

$$v_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Exponential form of the univariate Gaussian distribution. Let us consider the single-variable Gaussian distribution:

$$p(x|\lambda\,) = \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

$$= \underbrace{\frac{1}{\sqrt{2\pi}}}_{h(x)}\exp\left\{\underbrace{-\frac{x^2}{2\sigma^2}+\frac{2x\mu}{2\sigma^2}}_{\theta^{\mathrm{T}}T(x)}\underbrace{-\frac{\mu^2}{2\sigma^2}-\ln(\sigma)}_{-A(\theta)}\right\}$$

Therefore, we identify:

$$h(x) = 1/\sqrt{2\pi}$$
$$T(x) = \left[x,x^2\right]^{\mathrm{T}}$$

$$\theta = [\theta_1,\theta_2]^{\mathrm{T}} = \left[\frac{\mu}{\sigma^2},\frac{-1}{2\sigma^2}\right]^{\mathrm{T}} \qquad \textit{(1.19)}$$

To express $A(\theta)$ in terms of the parameters in the form of (1.19), we rewrite (1.19) to obtain

$$\mu = \frac{\theta_1}{(-2\theta_2)}$$
$$\sigma = (-2\theta_2)^{-\frac{1}{2}}$$

And therefore

$$A(\theta) = \frac{\mu^2}{2\sigma^2} + \ln(\sigma) = -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\ln(-2\theta_2)$$

Exponential form of the multivariate Gaussian distribution. Let us consider the D-dimensional multivariate Gaussian distribution:

$$\begin{aligned} p(x|\lambda\,) &= \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right\} \\ &= \underbrace{\frac{1}{(2\pi)^{\frac{D}{2}}}}_{h(x)}\exp\left\{\underbrace{-\frac{1}{2}x^{\mathrm{T}}\Sigma^{-1}x+\frac{1}{2}x^{\mathrm{T}}\Sigma^{-1}\mu+\frac{1}{2}\mu^{\mathrm{T}}\Sigma^{-1}x}_{\theta^{T}T(x)}\underbrace{-\frac{1}{2}\mu^{\mathrm{T}}\Sigma^{-1}\mu-\frac{1}{2}\ln|\Sigma|}_{-A(\theta)}\right\} \end{aligned}$$

Therefore, we can identify

$$\begin{aligned} h(x) &= (2\pi)^{-\frac{D}{2}} \\ T(x) &= \begin{bmatrix} x' \\ x'' \end{bmatrix} \\ \theta &= \begin{bmatrix} \theta' \\ \theta'' \end{bmatrix} \end{aligned}$$

where we denote by

$$\begin{aligned} x' &= x \\ x'' &= \mathrm{Vec}\left(xx^{\mathrm{T}}\right) \end{aligned} \qquad (1.20)$$

and

$$\begin{aligned} \theta' &= \Sigma^{-1}\mu \\ \theta'' &= \mathrm{Vec}\left(-\frac{1}{2}\Sigma^{-1}\right) \end{aligned} \qquad (1.21)$$

In the above, we define Vec(•) as a function that converts a matrix into a column vector in the following manner: First, it concatenates rows of the matrix one by one to form a row vector, and then transport the result to a column vector. Correspondingly, we define the inverse function of Vec(•) as IVec(•), which converts a column vector to a matrix. For example,

$$\mathrm{Vec}\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\right) = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{bmatrix} \quad \text{and} \quad \mathrm{IVec}\left(\begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{bmatrix}\right) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

We now express $A(\theta)$ as an explicit function of the parameter θ as defined in (1.21). For simplicity of notation, we define matrix Θ as

$$\Theta = \mathrm{IVec}\left(\theta''\right)$$

Note Θ is a symmetric matrix; that is, $\Theta = \Theta^{\mathrm{T}}$ and $\Theta^{-1} = (\Theta^{-1})^{\mathrm{T}}$.

Then we obtain

$$\mu = -\frac{1}{2}\Theta^{-1}\theta'$$

$$\Sigma = -\frac{1}{2}\Theta^{-1} = (-2\Theta)^{-1}$$

And we can derive

$$\begin{aligned} A(\theta) &= \frac{1}{2}\mu^{\mathrm{T}}\Sigma^{-1}\mu + \frac{1}{2}\ln|\Sigma| \\ &= \frac{1}{2}\frac{-1}{2}\theta'^{\mathrm{T}}\Theta^{-1}\theta' - \frac{1}{2}\ln|-2\Theta| \\ &= -\frac{1}{4}\theta'^{\mathrm{T}}\Theta^{-1}\theta' - \frac{1}{2}\ln|-2\Theta| \end{aligned}$$

We now discuss ML-based parameter estimation. Using matrix calculus, we obtain

$$\frac{\partial A(\theta)}{\partial \theta'} = -\frac{1}{2}\Theta^{-1}\theta' = \mu \tag{1.22}$$

$$\frac{\partial A(\theta)}{\partial \theta''} = \mathrm{Vec}\left(\frac{\partial A(\theta)}{\partial \Theta}\right) = \mathrm{Vec}\left(\frac{1}{4}\Theta^{-1}\theta'\theta'^{\mathrm{T}}\Theta^{-1} - \frac{1}{2}(-2)(-2\Theta)^{-1}\right) = \mathrm{Vec}\left(\mu\mu^{\mathrm{T}} + \Sigma\right) \tag{1.23}$$

According to (1.10), the ML estimate of θ should satisfy the following condition:

$$\begin{bmatrix} \dfrac{\partial A(\theta)}{\partial \theta'} \\ \dfrac{\partial A(\theta)}{\partial \theta''} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{N}\sum_{n=1}^{N} x'_n \\ \dfrac{1}{N}\sum_{n=1}^{N} x''_n \end{bmatrix} \tag{1.24}$$

Therefore, after substituting (1.20), (1.22), and (1.23) into (1.24), we obtain

$$\begin{bmatrix} \mu \\ \text{Vec}\left(\mu\mu^{\text{T}} + \Sigma\right) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{N}\sum_{n=1}^{N} x_n \\ \dfrac{1}{N}\text{Vec}\left(\sum_{n=1}^{N} x_n x_n^{\text{T}}\right) \end{bmatrix}$$

After rearrangement and canceling out the Vec( ) function on both sides, we have the estimation formula:

$$\mu_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\Sigma_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^{\text{T}} - \mu_{\text{ML}}\mu_{\text{ML}}^{\text{T}}$$

In addition to the multinomial and Gaussian distributions that are commonly used in speech modeling, we here also introduce a few other members of the exponential family. As will be shown in the following chapters, discriminative training for the general exponential family distributions is applicable to all the distributions discussed here.

Exponential form of the Poisson distribution. Poisson distribution has the following conventional form for one dimensional discrete variable:

$$p(x|\lambda) = \frac{1}{x!}\lambda^{x}\exp(-\lambda) \qquad\qquad x = 0,\ 1,\ 2,\ \ldots$$

$$= \underbrace{\frac{1}{x!}}_{h(x)} \exp\left\{ \underbrace{x\ln(\lambda)}_{\theta T(x)} \underbrace{-\lambda}_{-A(\theta)} \right\}$$

Therefore, we identify the quantities in the standard form of the exponential family:

$$h(x) = 1 \neq x!$$
$$T(x) = x$$
$$\theta = \ln(\lambda)$$
$$A(\theta) = \lambda = e^{\theta}$$

Exponential form of the exponential distribution. Exponential distribution has the conventional form:

$$p(x|\lambda) = \lambda \exp(-\lambda x) \qquad x \in \mathbb{R}^{+}$$
$$= \exp\left\{ \underbrace{-\lambda x}_{\theta T(x)} + \underbrace{\ln(\lambda)}_{-A(\theta)} \right\}$$

from which we identify:

$$h(x) = 1$$
$$T(x) = x$$
$$\theta = -\lambda$$
$$A(\theta) = -\ln(\lambda) = -\ln(-\theta)$$

Exponential form of the Dirichlet distribution. Dirichlet distribution takes the following form

$$p(x|\alpha) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} x(k)^{\alpha_k - 1}$$

where $\alpha = [\alpha_1, \ldots, \alpha_K]^T$ is the parameter vector, $\Gamma(\cdot)$ is the Gamma function defined as

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-z} dt \tag{1.25}$$

and $x = [x(1), \ldots, x(K)]^T$ is a K-dimensional observation vector with the constraints: $0 \leq x(k) \leq 1$, $\sum_{k=1}^{K} x(k) = 1$. Rewriting (1.25), we have

$$p(x|\alpha) = \exp\left\{ \ln \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} + \sum_{k=1}^{K} (\alpha_k - 1) \ln(x(k)) \right\}$$

$$= \underbrace{\exp\left\{ -\sum_{k=1}^{K} \ln(x(k)) \right\}}_{h(x)} \exp\left\{ \underbrace{\sum_{k=1}^{K} \alpha_k \ln(x(k))}_{\theta^{\mathrm{T}} T(x)} + \underbrace{\ln \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)}}_{-A(\theta)} \right\}$$

from which we can identify:

$$h(x) = \exp\left\{ -\sum_{k=1}^{K} \ln(x(k)) \right\}$$

$$T(x) = \begin{bmatrix} \ln(x(k)) \\ \vdots \\ \ln(x(K)) \end{bmatrix}$$

$$\theta = \alpha$$

$$A(\theta) = -\ln \frac{\Gamma\left(\sum_{k=1}^{K} \theta_k\right)}{\prod_{k=1}^{K} \Gamma(\theta_k)}$$

1.5 BACKGROUND: BASIC OPTIMIZATION CONCEPTS AND TECHNIQUES

In this section, we provide the mathematical background for basic optimization concepts and pertinent techniques that will be used in the remaining chapters of this book. In particular, we will introduce the growth-transformation-based optimization technique that applies to specific, rational forms of object functions. All topics discussed in this section will be used as the basic material for the following chapters in this book.

1.5.1 Basic Definitions

Let a vector Λ be in a K-dimensional parameter space, $\Lambda \in R^K$, and let $O(\Lambda)$ be a real-valued function of Λ. When we want to optimize the function $O(\Lambda)$, we call it the objective function w.r.t. to the parameter set Λ.

The function $O(\Lambda)$ with its domain $\Lambda \in R^K$ is said to have a global minimum Λ^* if

$$O(\Lambda^*) \leq O(\Lambda)$$

for all $\Lambda \in R^K$. The function $O(\Lambda)$ is said to have a global maximum Λ if

$$O(\Lambda^{**}) \leq O(\Lambda)$$

for all $\Lambda \in R^K$.

The function $O(\Lambda)$ is said to have a local minimum Λ_0 if

$$O(\Lambda_0) \leq O(\Lambda)$$

for all Λ in the neighborhood of Λ_0.

Note that since

$$\min O(\Lambda) = -[\max(-O(\Lambda))]$$

a minimization problem is equivalent to a maximization one. We thus will treat both of these problems as the same optimization problem.

The vector of partial derivatives of $O(\Lambda)$ w.r.t. Λ is called the gradient vector, which is often denoted by $\nabla O(\Lambda)$. The matrix of second-order partial derivatives of $O(\Lambda)$ is called the Hessian matrix, denoted by H_Λ.

1.5.2 Necessary and Sufficient Conditions for an Optimum

A necessary condition for a function $O(\Lambda)$ to have a local optimum at Λ^* is that the gradient vector has all zero components:

$$\nabla O(\Lambda^*) = 0$$

as long as $\nabla O(\Lambda)$ exists and is continuous at Λ^*. This necessary condition can be directly proved using Taylor series expansion.

Note that $\nabla O(\Lambda^*) = 0$ is only a necessary condition; that is, a point Λ^* satisfying $\nabla O(\Lambda^*) = 0$ may be just a stationary or saddle point, not an optimum point.

However, in many optimization problems including those in speech processing, previous knowledge about the nature of the objective function in the problem domain can eliminate the possibility of having a stationary point.

To theoretically guarantee an optimum point (i.e., elimination of the possibility of a stationary point), we have the following sufficient condition: Let there exist continuous partial derivatives up to the second order for objective function $O(\Lambda)$. If the gradient vector $\nabla O(\Lambda^*) = 0$ and the Hessian matrix H_{A} is positive definite, then Λ^* is a local minimum. Similarly, if the gradient vector $\nabla O(\Lambda^*) = 0$ and the Hessian matrix H_{A} is negative definite, then Λ^* is a local maximum.

Again, the proof of the above condition comes also directly from applying Taylor series expansion.

The necessary and sufficient conditions discussed above are applied to optimization problems with no constraints. For the situation where constraints must be imposed, the related optimization problems are discussed next.

1.5.3 Lagrange Multiplier Method for Constrained Optimization

The Lagrange multiplier method is a popular method in speech processing, as well as in many other optimization problems, which converts constrained optimization problems into unconstrained ones. It uses a linear combination of the objective function and the constraints to form a new objective function with no constraints.

The constrained optimization problem, where the constraints are in the form of equalities, can be formally described as follows: Find $\Lambda = \Lambda^*$ that optimizes the objective function $O(\Lambda)$ subject to the M constraints:

$$\begin{gathered} g_1(\Lambda) = b_1, \\ g_2(\Lambda) = b_2, \\ \cdots \\ g_M(\Lambda) = b_M. \end{gathered}$$

The Lagrange multiplier method solves the above problem by forming a new objective function for the equivalent unconstrained optimization:

$$F(\Lambda, \lambda) = O(\Lambda) + \sum_{m=1}^{M} \lambda_m \left[g_m(\Lambda) - b_m \right]$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_M)$ are called the Lagrange multipliers.

Optimization of the new objective function $F(\Lambda,\lambda)$ proceeds by setting its partial derivatives to zero with respect to each vector component of Λ and λ. This produces a set of $K + M$ equations that determine the $K + M$ unknowns including the desired solution $\Lambda = \Lambda^*$ for optimization.

When the constraints are in the form of inequalities, rather than of equalities as discussed above, a common method for optimization is to transform the related variables so as to eliminate the constraints. For example, if the constraint is $\Lambda > 0$ (e.g., as required for estimating the variance, which is always positive, in a PDF), then we can transform Λ into $\Lambda' = \exp(\Lambda)$. Because Λ and Λ' are monotonically related, optimization of one automatically gives the solution to the other. However, when using this type of transformation techniques, one should be aware of the sensitivity problem in the solution.

1.5.4 Gradient Descent Method

One popular family of numerical methods for optimization is based on gradient descent. As discussed earlier, the gradient is a vector in a K-dimensional space where the objective function is defined. The effectiveness of these gradient-based methods derives from its important property: The gradient vector represents the direction of steepest ascent of the objective function, and the negative gradient vector represents the direction of steepest descent. That is, if we move along the gradient direction from any point in the K-dimensional space over which the objective function is defined, then the function value increases at the fastest rate.

Note that the direction of steepest ascent is a local and not a global property. Hence, all the optimization methods based on gradients give only local optimum, and not global optimum. Due to the steepest ascent or descent property associated with the gradient vector, any method that makes use of it can be expected to find an optimum point faster than the methods without using it.

In the steepest descent method, one uses the negative gradient vector, $\nabla O(\Lambda)$, as a direction for minimizing an objective function $O(\Lambda)$. In this method, an initial point $\Lambda^{(0)}$ is supplied, and it iteratively moves toward the optimal point using the updating equation:

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \alpha_{\min}^{(t)} \nabla O(\Lambda^{(t)})$$

where $\alpha_{\min}^{(t)}$ is called the step size, and in the strict steepest descent method, the step size is optimized along the search direction $\nabla O(\Lambda^{(t)})$. That is, in each iteration of steepest descent, $\alpha_{\min}^{(t)}$ is found that minimizes $O[\Lambda^{(t)} - \alpha_{\min}^{(t)} \nabla O(\Lambda^{(t)})]$. In practice, for large-scale optimization problems such as speech recognizer training, the above procedure is difficult and $\alpha_{\min}^{(t)}$ is often determined empirically.

1.5.5 Growth Transformation Method: Introduction

The gradient descent method discussed above can be applied to any objective function, as long as the gradient can be computed efficiently (analytically or numerically, especially analytically). There are many other optimization techniques that take advantage of higher-order gradients, such as Newton's method. However, if the objective function has a special structure, more efficient optimization techniques than the gradient-based ones can be used. In this section, we provide preliminaries to optimizing rational functions, a common type of structure in the objective function, by a nongradient-based technique called "growth transformation" (GT).

Many times, the objective function of discriminative training of HMM can be formulated into a rational function, thus enabling the use of GT techniques. This type of techniques is also called extended Baum–Welch (EBW) algorithm when the underlying statistical model is an HMM. GT is an iterative optimization scheme where if the parameter set Λ is subject to a transformation $\Lambda = T(\Lambda')$, then the objective function "grows" in its value $O(\Lambda) > (\Lambda')$ unless $\Lambda = \Lambda'$. Hence the name growth transformation. GT or EBW algorithm was initially developed for the homogeneous polynomial by Baum and his colleagues (e.g., [4]). It was later extended to optimizing nonhomogeneous rational functions as reported in [14]. EBW algorithm became popular for its successful use in discriminative training of HMM using the maximum mutual information (MMI; see Chapter 3) criterion after the extension of the MMI training was made from the discrete HMM in [14] to the CD HMM in [3, 17, 34, 50, 52].

The importance of the optimization technique based on GT/EBW algorithm lies in its effectiveness and closed-form parameter updating for large-scale optimization problems with difficult objective functions (i.e., training criteria). With the traditional ML training where the likelihood function as the optimization criterion is relatively simple, a fast method is often available, such as the expectation–maximization (EM) algorithm for the HMM. In contrast, for the discriminative training criteria that are more complex than the ML, optimization becomes more difficult. For them, two general types of optimization techniques are available for the HMM: (1) gradient-based method and (2) GT/EBW. The latter has the advantage of having closed-form parameter updating formulas while not explicitly requiring second-order statistics. In addition, it does not require the same type of special and often delicate care for tuning the parameter-dependent learning rate as in the gradient-based methods (e.g., [25, 44]).

Let $G(\Lambda)$ and $H(\Lambda)$ be two real valued functions on the parameter set Λ, and the denominator function $H(\Lambda)$ is positive valued. And let the objective function be the ratio of them, giving the rational function of

$$O(\Lambda) = \frac{G(\Lambda)}{H(\Lambda)} \tag{1.26}$$

a GT-based optimization algorithm exists to maximize $O(\Lambda)$.

An example of this rational function is the objective function for discriminative learning of HMM parameters, which will be discussed in greater details in Section 3.2.2, where

$$G(\Lambda) = \sum_{s} p(X, s|\Lambda)\ C(s) \text{ and } H(\Lambda) = \sum_{s} p(X, \mathrm{s}|\Lambda) \qquad (1.27)$$

and we use $S = S_1, \ldots, S_R$ to denote the label sequences for all R training tokens, and use $X = x_1, \ldots, x_R$ to denote the observation data sequences for all R training tokens.

As originally proposed in [14], for the objective function of (1.26), the GT-based optimization algorithm constructs the auxiliary function of

$$F(\Lambda; \Lambda') = G(\Lambda) - O(\Lambda')H(\Lambda) + D \qquad (1.28)$$

where D is a quantity independent of the parameter set Λ, and Λ' denotes the parameter set obtained from the immediately previous iteration of the algorithm.

The algorithm starts by initializing the parameter set as, say, Λ'. (This is often accomplished by the ML training using, for instance, EM or Baum–Welch algorithm for HMMs.) Then, the updating of the parameter set from Λ' to Λ proceeds by maximizing the auxiliary function $F(\Lambda; \Lambda')$, and the process iterates until convergence is reached. Maximizing the auxiliary function $F(\Lambda; \Lambda')$ is often easier than maximizing the original rational function $O(\Lambda)$. And the following is a simple proof that as long as D is a quantity not relevant to the parameter set Λ, an increase of $F(\Lambda; \Lambda')$ guarantees an increase of $O(\Lambda)$.

Substituting $\Lambda = \Lambda'$ into (1.28), we have

$$F(\Lambda'; \Lambda') = \underbrace{G(\Lambda') - O(\Lambda')H(\Lambda')}_{=0} + D = D$$

Hence,

$$\begin{aligned} F(\Lambda; \Lambda') - F(\Lambda'; \Lambda') &= F(\Lambda; \Lambda') - D = G(\Lambda) - O(\Lambda')H(\Lambda) \\ &= H(\Lambda)\left(\frac{G(\Lambda)}{H(\Lambda)} - O(\Lambda')\right) = H(\Lambda)\left(O(\Lambda) - O(\Lambda')\right) \end{aligned}$$

Because $H(\Lambda)$ is positive, we have $O(\Lambda) - O(\Lambda') > 0$ on the right-hand side if $F(\Lambda; \Lambda') - F(\Lambda; \Lambda') > 0$ on the left-hand side. That is, for optimizing a complicated rational function, we can turn the problem to optimizing $F(\Lambda; \Lambda')$, which is often simpler.

In later chapters of this book, we will provide details of optimizing $F(\Lambda; \Lambda')$ for discriminative training of speech recognizer parameters.

1.6 ORGANIZATION OF THE BOOK

The main content of this book is an extensive account of the discriminative learning techniques that are currently popular in training HMM-based speech recognition systems. In this introductory chapter, we first clarify the concepts of discriminative learning and speech recognition, and then we proceed to discuss the roles of discriminative learning in speech recognition practice. We then introduce several basic probability distributions that will be used in the remainder of this book and that also serve as the building blocks for the more complex distributions such as HMMs. Finally, we introduce some basic concepts and techniques of optimization including the definition of optima, a necessary condition for achieving the optima, Lagrange multiplier method, and gradient descent method. We also provide preliminaries to the growth-transformation based optimization technique that applies to specific, rational forms of the objective functions naturally fitting to those in discriminative learning of popular distributions such as HMMs used in speech recognition.

In Chapter 2, we will provide a tutorial on statistical speech recognition and on the state-of-the-art modeling techniques, setting up the context in which discriminative learning is motivated and applied to. In particular, HMMs are formally introduced. In Chapter 3, we provide a unified account for the several common objective functions for discriminative training of HMMs currently in use in speech recognition practice. We also compare our unified form of these objective functions with another form in literature. How to do discriminative parameter learning using the unified form of objective functions via the GT technique is discussed in Chapters 4 and 5; Chapter 4 deals with exponential family distribution parameters, and Chapter 5 focuses on more difficult HMM parameters. Some practical implementation issues of the GT technique for HMM parameter learning are discussed in Chapter 6. In Chapter 7, selected experimental results in speech recognition are presented. Finally, an epilogue and summary is given in Chapter 8.

• • • •

Author Query Form

(Queries are to be answered by the Author)

He – Chapter 1

The following queries have arisen during the typesetting of your manuscript. Please answer these queries.

Query Marker	Query	Reply
Q1	"Re-ranging" was changed to "Rearranging" – ok?	

Thank you very much.