Multi-Source Contingency Clustering

by

Jacob V. Bouvrie

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

Author ....
Department of Electrical Engineering and Computer Science
June 28, 2004

Certified by.... ....
Tomaso Poggio
Eugene McDermott Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Multi-Source Contingency Clustering

by

Jacob V. Bouvrie

Submitted to the Department of Electrical Engineering and Computer Science
on June 28, 2004, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis examines the problem of clustering multiple, related sets of data simultaneously. Given datasets which are in some way connected (e.g. temporally) but which do not necessarily share label compatibility, we exploit co-occurrence information in the form of normalized multidimensional contingency tables in order to recover robust mappings between data points and clusters for each of the individual data sources.

We outline a unifying formalism by which one might approach cross-channel clustering problems, and begin by defining an information-theoretic objective function that is small when the clustering can be expected to be good. We then propose and explore several multi-source algorithms for optimizing this and other relevant objective functions, borrowing ideas from both continuous and discrete optimization methods. More specifically, we adapt gradient-based techniques, simulated annealing, and spectral clustering to the multi-source clustering problem.

Finally, we apply the proposed algorithms to a multi-source human identification task, where the overall goal is to cluster grayscale face images according to identity, using additional temporally connected features. It is our hope that the proposed multi-source clustering framework can ultimately shed light on the problem of when and how models might be automatically created to account for, and adapt to, novel individuals as a surveillance/recognition system accumulates sensory experience.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor

Acknowledgments

I would firstly like to thank Tommy for making this thesis possible, and for maintaining a fantastic research environment at CBCL. It has been a pleasure to be a part of the group this past year.

To Yuri Ivanov, I owe my sincerest gratitude. He has gone truly above and beyond the ordinary obligations of an advisor from day one, and has always been available, whether to clarify a perplexing concept, or to keep the big picture in focus. Yuri's depth of knowledge and sense of humor make him a rare commodity, and it has been a honor to work with him.

I would also like to thank Ryan Rifkin for numerous helpful discussions and insightful comments, Thomas Serre for his generous help with the vision system and data collection, Bernd Heisele for helping me maintain a sense of humor throughout this project, and Jerry Jun Yokono for letting us crowd into his office and have loud conversations while he tried to get work done.

Lastly, I thank my parents for their unwavering support and encouragement during my years at MIT.

Contents

List of Figures

List of Tables

List of Algorithms

Notation

$\hat{x}$, $\hat{y}$, $\hat{z}$ Clusters, defined as either collections of original data points or as prototypes where appropriate.

C_x, C_y, C_z Mappings from sets of points $\{x\}$, $\{y\}$, $\{z\}$ to clusters $\{\hat{x}\}$, $\{\hat{y}\}$, $\{\hat{z}\}$, resp.

m, n Number of rows (m) and columns (n) in a co-occurrence matrix.
Also, the number of VQ-prototypes (codewords) defining the corresponding datasets $\{x\}$, $\{y\}$ resp.

m', n' Desired number of row and column clusters, resp.

$J(C_x, C_y)$ An objective function $J : (C_x, C_y) \rightarrow \mathbb{R}$ relating cluster mapping functions to a real-valued quantity indicating the quality of the clustering.

$J(\mathbf{w}^r, \mathbf{w}^c)$ An objective function $J : (\mathbb{R}^{m' \times m}, \mathbb{R}^{n' \times n}) \rightarrow \mathbb{R}$ relating real-valued row and column weights to a real-valued quantity indicating the quality of the clustering.

Chapter 1

Introduction

This thesis is primarily concerned with unsupervised learning, and in particular, clustering of multiple connected feature sets. Historically, the majority of research involving unsupervised clustering has been directed mainly towards single-source applications. That is, applications which involve a single set of unlabeled data and a single desired target function to be recovered. Both hard membership techniques, such as K-means clustering or hierarchical agglomerative clustering, as well as soft assignment statistical models such as mixtures of Gaussians, have been successfully applied in the course of solving single channel clustering tasks [12, 18].

However, many real-world problems to which one might want to apply unsupervised methods are inherently multi-source. For such "multimodal" applications, there are typically two or more distinct sets of measurements which describe the same underlying physical processes, but which do not necessarily cluster according to similar labels or even represent the same number of underlying clusters. The main challenge that we will consider then is how one might recover a specified target function using as much available data as possible, regardless of how that data might be organized. Assuming that the modalities are relatively decorrelated–redundant data will clearly not help–it is not immediately clear how any of the time-honored methods mentioned above can be applied to multiple datasets, designed as they were, for single-source circumstances. The main contribution of this thesis, therefore, will be in the form of clustering algorithms designed specifically for multi-source applications. As we will discuss in subsequent chapters, the core aim of each algorithm will be to take advantage of multiple datasets and the interactions between them, in order to recover one or more mappings from examples to clusters.

1.1 Road Map

The format of Chapter 1 is as follows: we first describe with broad strokes how and where our clustering problem fits within the general machine learning process. The second section discusses why the standard methods for single-source clustering can be expected to fail when applied to multi-source datasets. We then formally state the multi-source clustering problem, and introduce an objective function that can be applied to a broad range of clustering scenarios in which datasets from multiple sources are available. Finally, we conduct a brief review of previous work concerning clustering with multiple sets of features.

Beyond the current chapter, we will describe two classes of algorithms for clustering multi-source data: one in which we optimize over spaces of discrete, integer-valued solutions, and another based on continuous optimization with gradient-based methods. We then apply the proposed methods to a human identification task involving multiple high-dimensional databases of audio-visual features. Using multimodal clustering algorithms, we argue that an unsupervised solution to this traditionally supervised problem is both feasible and tractable. We finally compare the strengths and weaknesses of the proposed algorithms and discuss the applicability of multi-source clustering to other problem domains.

1.2 Characterization of the Clustering Problem

Consider the generalized machine learning formulation shown in Figure 1-1, adapted from [31]. Here, an observational *generator* (the underlying physical phenomena) produces independent samples $\mathbf{x}$ from an unknown distribution $P(\mathbf{x})$, an *oracle* provides output labels y for each input $\mathbf{x}$ according to the unknown distribution $P(y|\mathbf{x})$,

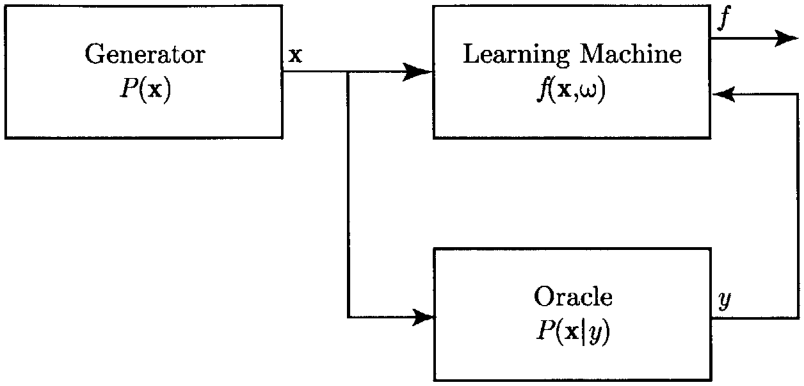

Figure 1-1: The generalized machine learning problem.

and a *learning machine* implements collections of abstractly parameterized functions $f(\mathbf{x}|\omega \in \Omega)$. The variable ω might function as an explicit model parameter, or as an index into a class of functions. In a supervised learning setting, the learning machine observes the example-label pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, and attempts to approximate as best as it can the output of the oracle given previously unseen examples. Given a loss function $L(f(\mathbf{x}, \omega), y) \geq 0$ that measures how good of an approximation f is to the oracle's outputs y, we can calculate the average amount of loss the learning machine will incur for a given ω via the expected risk:

$$R(\omega) = \int L(f(\mathbf{x}, \omega), y) P(\mathbf{x}, y) \, d\mathbf{x} \, dy.$$

The learning task is then to find the approximation $f^*(\mathbf{x}, \omega)$ that minimizes this risk functional.

In the unsupervised learning setting, the labels y_i are not observed and we must find an approximation f given only the data examples $\mathbf{x}_i$. The learning goal in this case is to find the best mapping function or from examples to clusters. In the signal processing literature, the mapping function is often interpreted as a vector quantizer or "codebook" mapping data points to cluster prototypes or "codewords". As one possible choice of loss function for clustering problems, we could summarize distortion (energy loss) in going from points to cluster center coordinates:

$$L(f(\mathbf{x}, \omega)) = \|\mathbf{x} - f(\mathbf{x}, \omega)\|,$$

where $\|\cdot\|$ is a suitably chosen norm. Indeed, most implementations of the K-means clustering algorithm [12] attempt to minimize the following risk functional:

$$R(\omega) = \int \|\mathbf{x} - f(\mathbf{x}, \omega)\|_2^2 P(\mathbf{x})\, d\mathbf{x}\, dy.$$

For most of the algorithms presented in this thesis however, we will attempt to minimize a risk functional based on an information-theoretic loss that is better suited to clustering applications than the energy loss. Within this context, we discuss the nature of the function $f(\mathbf{x}, \omega)$ and loss $L(\cdot)$ in sections 1.4.1 and 1.4.2 below, and in section 1.4.3 compare the information-theoretic loss to the more common distortion-based loss function just described.

1.3 Difficulties in Applying Classical Clustering Techniques

A note concerning terminology and assumptions: we will use the terms "multimodal", "multi-source", and "multi-channel" to refer to multiple sets of distinct features, all describing the same underlying physical process. We will further assume that the sets of features are connected in at least one dimension (e.g. space or time), and suppose that the overarching goal is to recover the target function that clusters one of the datasets. We thus assume the presence of a "primary" data channel, supplemented by other data channels that describe the same physical process, but do not necessarily share label compatibility. The hope is that, by somehow incorporating additional related data, the primary channel can be clustered better than if we had only the primary channel to work with (e.g. a single set of data).

We'll now consider the obstacles that arise when applying traditional clustering methods to multi-source datasets. But if we want to somehow graft existing single-source methods onto multimodal problems, the definitions above immediately create several substantial difficulties. If the modalities are connected temporally, for example, then we must ensure that the samples in each channel are aligned in time or suitably binned. Furthermore, it is not clear how to normalize each channel or how to choose a distance/similarity measure such that direct comparisons between points from separate channels can be made. For most non-generative hierarchical or partitional methods (e.g. K-means or agglomerative clustering), this question is of central importance.

In most cases, it is not feasible to concatenate together features from the modalities and run either hard or soft membership clustering algorithms out of the box. In high-dimensional feature spaces the data requirements for generative algorithms can quickly become impossibly large. In addition, for models based on statistical mixture densities, computation of the parameterized distributions can become unwieldy. In the case of a Gaussian mixture model, computing the determinant or inverse of a 1500x1500 covariance matrix, even if constrained to be diagonal, is neither straightforward nor speedy.

Similarly, in the case of distortion-based hard membership algorithms, many choices for the distance measure have poor discriminability in high dimensions due to averaging effects. The Minkowski metrics are particularly susceptible to this problem [1]. Even with prior dimensionality reduction (and the associated loss of information), it is not clear how to appropriately weight each dimension a priori. As an example of this predicament, consider an application where we have scalar measurements of an individual's height in one channel, and 50x50 pixel face images in another. Concatenating the two into a single long feature vector would effectively erase the influence of the height measurement. Concatenating features is also not necessarily guaranteed to give a better clustering with respect to the primary modality's target function either; we could simply be better off using only the primary dataset. Thus, we can reasonably assume that combining features and clustering the resulting vectors is not in general a viable option, given the limitations of most popular single-source clustering algorithms.

In the following section, we will propose a clustering framework that can avoid most of these difficulties by, in a nutshell, focusing not on the data itself but on the frequency of co-occurrences among observations.

1.4 The Co-Clustering Paradigm

To summarize our goal, we want to treat features jointly and cluster simultaneously in order to take advantage of interactions between datasets. To that end, we consider the space of co-occurrences between data points in each modality along the shared dimension. This particular formulation of the clustering problem will form the basis for several algorithms proposed in later sections.

1.4.1 Co-Clustering Formulation

Assuming that the data points in each of the N modalities have been suitably quantized, we define the N-variable joint probability distribution $P(x, y, z, ...)$ to be the normalized N-way contingency table over all (connected) modalities. Let the row and column dimensions m, n of the co-occurrence distribution equal the number of bins resulting from quantization of two datasets, and let m', n' denote the desired number of row/column clusters respectively. For simplicity (and without loss of generality), from here on we will consider clustering 2-way co-occurrences resulting from intersections of points falling into each row and column quantization bin. Following [11, 18], the multimodal clustering problem can be formally stated as follows:

Definition 1. *Given the joint distribution $P(x, y)$, we wish to cluster the distinct feature sets $\{x\}$ and $\{y\}$ into clusters $\{\hat{x}\}$ and $\{\hat{y}\}$ via the separate mappings C_x and C_y:*

$$C_x : \{x_1, x_2, ..., x_m\} \longrightarrow \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_{m'}\}$$

$$C_y : \{y_1, y_2, ..., y_n\} \longrightarrow \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_{n'}\}$$

where it is assumed that $m' < m$ and $n' < n$.

Given this definition, the joint probability distribution over clusters can be written as:

$$P(\hat{x}, \hat{y}) = \sum_{x \in \hat{x}, y \in \hat{y}} P(x, y). \tag{1.1}$$

Because the mappings C_x and C_y operate along a single dimension, note that the joint cluster distribution corresponds to sums over "blocks" of the original joint. The blocks are comprised of cells which are at the intersection of sets of rows and columns: a given "joint" cluster $(\hat{x}, \hat{y})$ cannot simply claim arbitrary cells from the original joint.

1.4.2 A Co-Clustering Objective Function

In order to evaluate the quality of a given assignment of data points to clusters, we choose an information-theoretic objective function that measures the loss of mutual information between the original (unclustered) modalities and the resulting clustered form. This particular objective has been used in recent unsupervised learning research, including document classification and other clustering applications [11, 28, 13].

Let X, Y be random variables describing the original data points, and let $\hat{X}, \hat{Y}$ describe the clusters (that is, the empirical data and resulting clusters are samples from the unknown distributions governing these random variables). The objective function is then,

$$J(C_x, C_y) = I(X; Y) - I(\hat{X}; \hat{Y}). \tag{1.2}$$

We want to minimize this objective with respect to the mappings C_x and C_y, which corresponds to maximizing just $I(\hat{X}, \hat{Y})$, since $I(X; Y)$ is fixed for a given $P(x, y)$ and $I(\hat{X}, \hat{Y}) \leq I(X, Y)$ by Theorem 1 below. We now show that this objective can be written concisely as a Kullback-Leibler divergence:

Theorem 1. *The objective* (1.2) *can be written as the KL-divergence between the original joint probability distribution* $P(x, y)$*, and a "compressed" approximation* $Q(x, y)$ *resulting from clustering.*

Proof. (see e.g. [11, 8]):

$$\begin{aligned} I(X;Y) - I(\hat{X};\hat{Y}) &= \sum_{\hat{x},\hat{y}} \sum_{x\in\hat{x},y\in\hat{y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \\ &- \sum_{\hat{x},\hat{y}} \sum_{x\in\hat{x},y\in\hat{y}} P(x,y)\log\frac{P(\hat{x},\hat{y})}{P(\hat{x})P(\hat{y})} \\ &= \sum_{\hat{x},\hat{y}} \sum_{x\in\hat{x},y\in\hat{y}} P(x,y)\log\frac{P(x,y)}{P(\hat{x},\hat{y})\frac{P(x)}{P(\hat{x})}\frac{P(y)}{P(\hat{y})}} \\ &= D_{KL}(P||Q) \end{aligned}$$

where the approximation distribution $Q(x,y) = P(\hat{x},\hat{y})\frac{P(x)}{P(\hat{x})}\frac{P(y)}{P(\hat{y})}$. □

As [11] points out, from the decomposition

$$D_{KL}(P||Q) = H(\hat{X},\hat{Y}) + H(X|\hat{X}) + H(Y|\hat{Y}) - H(X,Y),$$

we can see that this objective function explicitly takes advantage of row/column interactions via the joint entropy term $H(\hat{X},\hat{Y})$.

Unfortunately, the objective function (1.2) does not depend smoothly on the mappings C_x and C_y, making direct continuous optimization difficult. We will address this problem and explore a reworked version to which continuous optimization methods can be applied in Chapter 3.

1.4.3 Co-Clustering as a Low-Rank Approximation

For a large majority of clustering problems, the desired number of clusters is significantly smaller than the number of original data points. Given that we are concerned here with 2-way joint co-occurrence matrices, it is helpful and informative to think of co-clustering as a process by which to obtain low-rank approximations to a given contingency matrix. Indeed, the approximation $Q(x,y) = P(\hat{x},\hat{y})\frac{P(x)}{P(\hat{x})}\frac{P(y)}{P(\hat{y})}$ discussed in the previous section will have rank less than or equal to $\min(m', n')$.

If we take the singular value decomposition of the rank r matrix P so that

$$P = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

then the best approximation of P by a rank $\nu < r$ matrix P_ν is

$$P_\nu = \sum_{i=1}^{\nu} \sigma_i u_i v_i^T. \tag{1.3}$$

This approximation is optimal in the sense that it minimizes the quantity $\|P - P_\nu\|_2$, and thus preserves as much of the energy in P as possible (we could have just as easily used $\|\cdot\|_F$ here as well, see e.g. [30] for details and a proof of the above statement).

The information-theoretic objective (1.2), however, does *not* seek to capture in Q as much of the energy of P as possible! This objective instead seeks to emphasize low-rank approximations which maximize the mutual information between the variables upon which P depends. Thus, an energy preserving approximation will not necessarily correspond to a good clustering: we can make it our primary goal to minimize the loss in overall energy, but if the loss in mutual information is suboptimal, then the resulting clustering will be suboptimal as well. We can conclude then that an objective which attempts to minimize the loss in mutual information directly will be more effective for clustering than distortion based objectives. Because distortion-minimizing objectives typically summarize the distances between observations and cluster prototypes, such objective functions amount to a form of energy maximization and are therefore suboptimal for co-clustering.

Consider the following concrete example. We are given the rank 4 matrix P along with an optimal co-clustering satisfying (1.2), yielding the rank 2 approximation Q:

$$P = \begin{pmatrix} .1 & .1 & 0 & 0 \\ .1 & .2 & 0 & 0 \\ 0 & 0 & .05 & .05 \\ 0 & 0 & .15 & .25 \end{pmatrix} \quad Q = \begin{pmatrix} .08 & .12 & 0 & 0 \\ .12 & .18 & 0 & 0 \\ 0 & 0 & .04 & .06 \\ 0 & 0 & .16 & .24 \end{pmatrix}.$$

Define P_2 to be the rank 2 maximum-energy approximation of P defined by (1.3). That the co-clustering approximation Q is better than P_2 in terms of loss in mutual information, but worse in terms of energy preservation, can be seen from the following computations:

$$\begin{array}{ccc} D_{KL}(P||Q) = 0.0137b & & D_{KL}(P||P_2) = 0.0154b \\ \|P - Q\|_2 = 0.0400 & \text{as compared to} & \|P - P_2\|_2 = 0.0382. \end{array}$$

Thus it is not generally the case that energy minimization will give a good co-clustering as defined by equation (1.2), or conversely that the best cluster assignments will give an optimal approximation in terms of an energy-based criterion.

1.5 Previous Work

Little work has been done to address multimodal clustering as a general learning problem, however researchers have thus far tackled a host of closely related issues arising in several specific problem domains. We give a brief summary of research with connections to the thesis problem, and assess the applicability of the techniques to clustering multi-modal data.

1.5.1 Simultaneous Clustering

Simultaneous clustering concerns the concurrent clustering of related but separate datasets through the cross-modal exchange of information in one form or another. That is, over the course of a given simultaneous algorithm, information regarding the current state of a clustering process in one dataset is passed to and utilized by another. It is then a matter of how to represent this cross-modal information, how to exchange it, and then what to do with it.

Ivanov and Blumberg [16] indirectly provide a framework for incorporating knowledge about one EM-based mixture model clustering process into another by setting posterior class membership probabilities proportional to a Boltzmann distribution over possible pairings of observations and their memberships. This approach presents an opportunity for information exchange via the free Boltzmann parameter β, which one might vary according to another clustering process.

Other simple schemes involving the EM algorithm exist, whereby at each iteration the posterior class membership distribution for each point is taken from one of the clustering processes and used for all of the respective maximization steps. Approaches where posterior probabilities are taken from the clustering process which has recently given the largest gain in likelihood (e.g. "follow-the-leader"), or where the posterior distribution with maximum entropy is chosen, are examples of such schemes.

Experiments with the multimodal datasets used in Chapter 4 showed that these methods unfortunately allow weaker channels to corrupt stronger ones. That is, if a "strongly clusterable" dataset is combined with a "weakly clusterable" dataset, the results are somewhere in between for both datasets: the strong channel is worse off than if we had clustered it alone, while the weak channel clusters better than if considered alone. Thus in this situation the question of which channel to trust more is of central concern. Unfortunately, it is not always the case that *a priori* information to this extent is available or generally trustworthy.

1.5.2 CoTraining

First introduced by Blum and Mitchell [4], *cotraining* combines ideas from semi-supervised learning and simultaneous clustering. In [4] the authors consider the task of learning to classify web pages described by separate sets of features: word frequencies from the page itself, and words occurring in links to that page. The goal of cotraining is then to use both sets of web page features to learn strong classifiers from relatively few "expensive" labeled examples augmented by many "inexpensive" unlabeled examples. The suggested strategy trains separate classifiers for each modality, but uses predictions on new unlabeled data in one channel to modify the training set for another. Other researchers have pointed out the complication that some applications require far more labeled data than others in order to achieve reasonably small classification error rates. Typically problems for which simple clustering methods fare poorly fall into this category. Pierce and Cardie [25] report difficulty applying cotraining to natural language processing tasks, and resort to a more supervised variation in which a human corrects mistakes during the "automatic" labeling phase. Nevertheless, Nigam and Ghani [24] provide a theoretical analysis of cotraining, and justify the intuition that, under suitable conditions, one clustering process will be able to benefit another.

Unfortunately, cotraining rests on two critical assumptions. The first requires target function compatibility. That is, the target functions corresponding to each set of features must predict roughly the same labels. The second assumption requires conditional independence of the feature sets. Even assuming the availability of partially-labeled data, the features available in many applications (e.g. to an identification system) are radically different in both scale and dimensionality, and often do not cluster according to a single universal target set. Cotraining is therefore only appropriate for certain classes of problems that satisfy the above assumptions. Conversely, the information-theoretic co-clustering objective, coupled with an appropriate optimization algorithm (as we shall see), is applicable under a much wider range of conditions, and does not require target function compatibility.

1.5.3 Information-Theoretic Clustering

Tishby et. al. [29] have proposed an general information-theoretic method for single-source data clustering which attempts to maximize the mutual information between clustered and unclustered variables $I(X;\hat{X})$, using an intermediate "auxiliary" variable which generates soft partitions over a dataset X subject to another informative variable Y. The tradeoff between clustering (compression) and maximization of the relevant information $I(X;Y)$ is controlled by a positive Lagrange multiplier β giving the Lagrangian $L = I(X;\hat{X} - \beta I(\hat{X};Y)$. The authors propose a deterministic annealing procedure for optimization of this Lagrangian.

Dhillon et. al. [11] have proposed an iterative algorithm based on the information-theoretic framework due to Tishby et. al. that monotonically decreases the objective function (1.2), but is restricted to the 2-channel case and relies on explicitly representing the entire joint distribution. By expressing the loss in mutual information in terms of Kullback-Liebler divergence, the authors alternate between assigning rows and columns of a contingency matrix to clusters on the basis of (KL-)distance to row/column prototypes. The method generally converges quickly and is computationally efficient, but is substantially susceptible to local minima in that the set of random initial mappings leading to a "good" minimum is small (if not null). In practice the algorithm must be run over many trials and averaged to obtain a single "average" clustering, which is often inadequate. Empirically, there is also large variability in the quality of the results. For the human identification problem explored later in this thesis, however, Dhillon's algorithm offers moderately good performance and we will use it as a standard to which the performance of other algorithms based on the same objective can be compared.

Optimization of various information-theoretic quantities has recently become popular in other learning domains as well. Jaakkola et. al. [17] have attempted to merge the accuracy of discriminative classifiers with the flexibility offered by generative models, while Viola et. al. [32] apply maximization of mutual-information to image alignment problems. We will, however, restrict our attention to clustering applications only.

Chapter 2

Algorithms Involving Discrete Solution Spaces

In this chapter we will primarily discuss algorithms for multi-source clustering that search over solution spaces of discrete-valued cluster membership functions. The first of which will borrow ideas from *simulated annealing*, a technique based on the process by which physical systems settle into stable low energy configurations. This algorithm attempts to directly optimize the information-theoretic objective function (1.2), while avoiding poor local minima. The second algorithm clusters data points in an embedding resulting from the process by which contingency matrices are built, and represents the application of a traditionally single-source tool (spectral clustering) to a multi-source problem. Before discussing these algorithms however, we first offer a general technique by which arbitrarily large multimodal joint distributions can be clustered by considering only 2-dimensional marginals.

2.1 Clustering N-way Multimodal Datasets

In this section we will describe a flexible framework for handling the general multimodal case of N datasets and the associated N-variable co-occurrence matrix. For most of the algorithms presented in this thesis, the full joint distribution must be stored in order to conduct co-clustering on all sets of features simultaneously. But since storage space requirements grow exponentially as we consider more modalities, it would be desirable to instead work with smaller, more manageable subsets. One possible technique is summarized in Algorithm 1. We assume that each given feature set $\mathcal{D}^1, \dots, \mathcal{D}^N$ is represented by codebooks $\{z_i^1\}_{i=1}^{m_1}, \dots, \{z_i^N\}_{i=1}^{m_N}$ (respectively), and denote by $\{D_i^\mu\}$ the set of points in $\mathcal{D}^\mu$ assigned to codeword z_i^μ. The size of a set is denoted by $|\cdot|$, and COCLUSTER represents any 2-way co-clustering algorithm that requires initial mappings in some form or another.

Algorithm 1 Generalized N-way Co-Clustering

```
Randomly initialize $C_1(x_1) \in \{1, \dots, m'_1\}\ \forall x_1, \dots, C_N(x_N) \in \{1, \dots, m'_N\}\ \forall x_N$,
  where $m'_1, \dots, m'_N$ are the desired number of clusters for each modality.
Given N quantized sets of features, form contingency matrices for
  all $\binom{N}{2}$ unique pairs of datasets:
  $\mathbf{P}_{ij}^{\mu\nu} = |\{D_i^\mu\} \bigcap \{D_j^\nu\}|\ , \mu = 1, \dots, N-1,\ \nu = \mu + 1, \dots, N,\ \forall i, j.$
Normalize: $\mathbf{P}_{ij}^{\mu\nu} \leftarrow \frac{\mathbf{P}_{ij}^{\mu\nu}}{\sum_{p,q} \mathbf{P}_{pq}^{\mu\nu}}\ \forall \mu, \nu.$
do
    for $\mu = 1, \dots, N-1$
        for $\nu = \mu + 1, \dots, N$
            $C_\mu, C_\nu \leftarrow$ COCLUSTER($\mathbf{P}^{\mu\nu}, C_\mu, C_\nu$)
until convergence criterion met, or maximum iterations reached
```

The idea behind this algorithm is to take advantage of interactions of order 2 at the most (and ignore higher order information) by performing interleaved clustering on 2-way marginals of the full N-variable joint distribution. Figure 2-1 illustrates the concept in the three-dimensional joint case, where we assume three hypothetical datasets A, B, and C. Each left-hand cube in the figure represents the 3-way, rank 2 co-occurrence tensor rotated to show summation along each of the dimensions. After summing out one of the three variables in each case, we are left with the three 2-way marginals shown respectively on the right-hand side. The example 2-way distributions illustrated in the figure were borrowed from the human identification dataset discussed in Chapter 4.

Ultimately, by interleaving clustering processes we aim to realize an improvement by biasing the solution of a given clustering sub-problem towards a specific part of the solution space. In Algorithm 1, biasing is accomplished by initializing each subprocess with shared partitions from previous clusterings. It can immediately be seen that this particular approach eliminates the exponential storage requirement that would otherwise be necessary (and eventually intractable) with a full joint. A given single, large clustering problem is thus made into a set of small tractable sub-problems that can be handled faster and more efficiently by a wide range of clustering algorithms. The division of an N-way problem into smaller self-contained tasks also opens the door to parallelization or network distribution in the case of massive data sets with many samples and/or hundreds of modalities. Lastly, interleaving of clustering processes also suggests a host of hybrid algorithms that could be tailored to the specific interactive nature of each marginal, and the corresponding manifolds on which the rows and columns live. We might choose one algorithm for clustering the first marginal, another for the second, and so on. For many application domains, it is not unreasonable to assume that different modalities interact differently with one another.

There are, however, two major drawbacks to the proposed scheme. Firstly, we do not take advantage of possibly informative higher order interactions and instead try to do our best with the set of all 2-way co-occurrences. Second, because each sub-problem can only influence the others through initial conditions, there is the possibility that one clustering process can corrupt another depending on the nature of the particular problem.

2.2 Stochastic Simulated Annealing

In this section we describe an iterative algorithm for direct optimization of the co-clustering objective (1.2) described in Chapter 1. Based on stochastic simulated annealing, the algorithm seeks to reduce the sensitivity to initial conditions exhibited by Dhillon's algorithm [11], and it is easily applied in the general case to problems with N modalities. Despite the fact that the algorithm can handle an arbitrary number of modalities, in situations where N is large the algorithm can be also used within a framework such as Algorithm 1 to avoid problematic storage requirements or to capture interactions up to a specified order only. Most importantly however, the annealing approach poses an attractive optimization tool for multi-source clus-

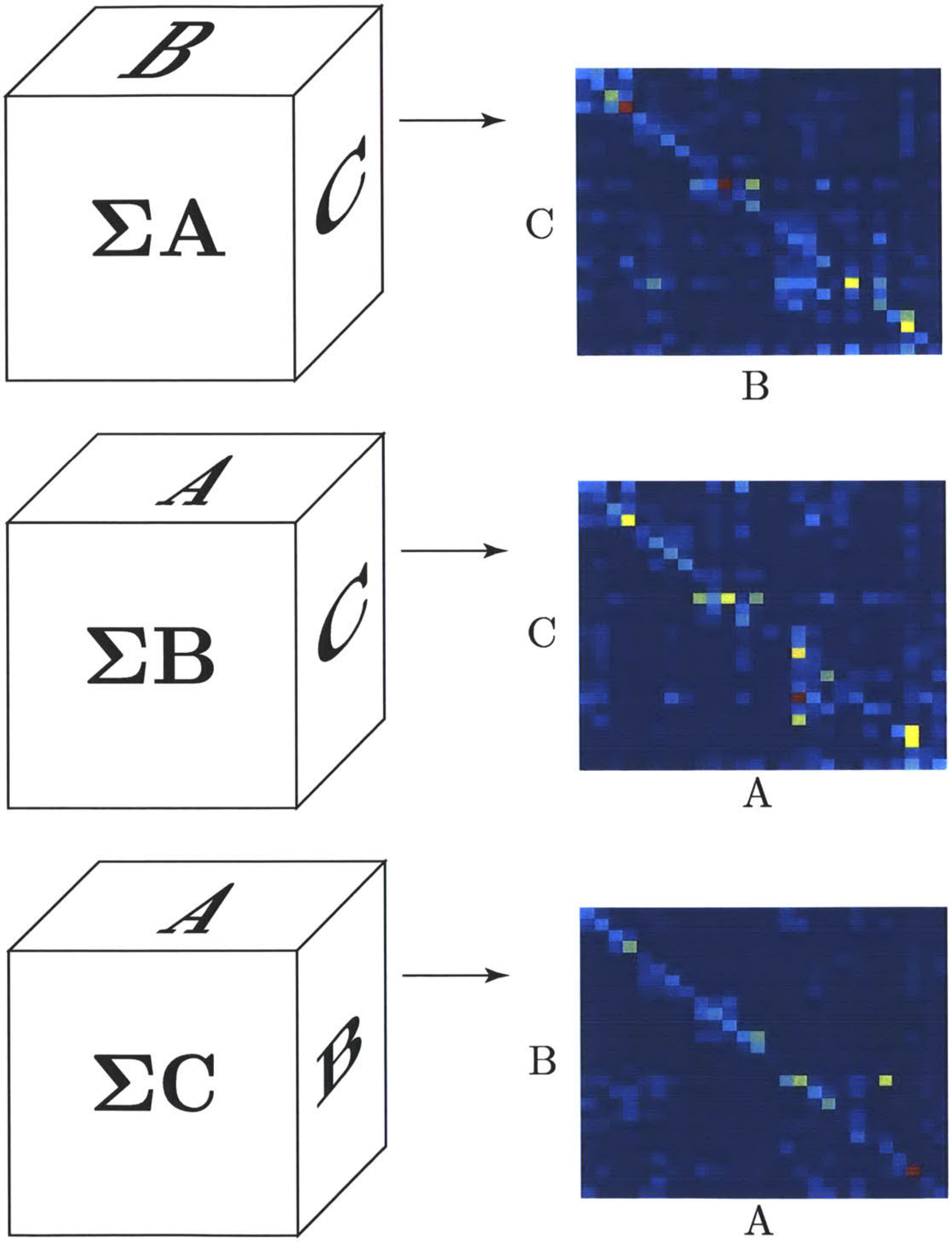

Figure 2-1: A 3-dimensional joint distribution resulting from the three hypothetical datasets A, B and C, is broken down into three 2-variable marginals to which a 2-way co-clustering algorithm can be applied. Interleaving repeatedly clusters each 2-way distribution in succession.

tering because it does not require that the optimization variables depend smoothly on the objective. In the formulation that follows, we directly search over the space of mappings $C_{\hat{x}}, C_{\hat{y}}$ by stochastic sampling.

In general, clustering is an NP-hard problem and a globally optimal solution cannot be guaranteed by any algorithm that attempts to minimize distortion. Similarly, the co-clustering objective we have chosen to optimize (1.2) is also non-convex, and is plagued with poor local minima. *Simulated Annealing* is a technique borrowed from physics that aims to avoid such local optima and settle on a solution that is close or equal to the global optimum by allowing the solution process to jump out of local minimums with probability determined by a time dependent "temperature" parameter. If a random change causes a decrease in the objective, we keep the change. If it increases the objective, then we still keep the change with probability

$$p = e^{-\Delta J/T},$$

where ΔJ is the change in the objective due to a random modification applied to the optimization variables. Over the course of optimization, the temperature parameter T is slowly reduced from a large initial value to a finishing value near zero. The effect is that the optimization process initially has a chance to explore what at the time appears to be unfavorable areas of the search space. When the temperature is reduced, randomness in the system is eliminated and the optimization process converges onto a solution that could be better than what would have resulted had we simply followed the best looking initial minimum. The interested reader is referred to [12, 20] for further details, and the connection to physical annealing.

Because simulated annealing depends only on computing the change in the objective function, we can directly apply it to a large range of optimization problems for which the gradient cannot be computed analytically. Thus, for the co-clustering objective as stated in (1.2) we can apply simulated annealing without further modification; stochastic changes to the partition functions can be made directly, and we need only evaluate the objective itself.

2.2.1 Multi-Source Stochastic Simulated Annealing

Bearing in mind that the joint clusters $(\hat{x}, \hat{y})$ are constrained to claim only cells at the intersection of rows and columns, in Algorithm 2 we give a simulated annealing implementation for multi-source clustering based on reassignments of entire rows and columns. Here, the objective $J(\cdot)$ can be any function which is small when the clustering is good, however for the experiments in Chapter 4 the information-theoretic objective (1.2) is assumed.

Algorithm 2 Multi-Source Stochastic Simulated Annealing (2-way case)

```
1.  Randomly initialize $C_{\hat{x}}(x) \in \{1, \dots, m'\}\ \forall x$, $C_{\hat{y}}(y) \in \{1, \dots, n'\}\ \forall y$.
    Initialize $T \leftarrow T_0$, $\gamma \in [0.8, 0.99]$.
2.  while $T > T_{stop}$
3.          while all rows and columns have not been visited several times
4.                $E_a \leftarrow J(\mathbf{C}_{\hat{x}}, \mathbf{C}_{\hat{y}})$
5.                $r \leftarrow \text{rand}[0,1]$, $\tilde{\mathbf{C}}_{\hat{x}} \leftarrow \mathbf{C}_{\hat{x}}$, $\tilde{\mathbf{C}}_{\hat{y}} \leftarrow \mathbf{C}_{\hat{y}}$
6.                if $r > 1/2$
7.                   randomly select a row $u \in \{1, \dots, m\}$
                              and cluster ID $v \in \{1, ...m'\}$
8.                   $\tilde{C}_{\hat{x}}(u) \leftarrow v$
9.                else
10.                  randomly select a column $u \in \{1, \dots, n\}$
                              and cluster ID $v \in \{1, ...n'\}$
11.                  $\tilde{C}_{\hat{y}}(u) \leftarrow v$
12.               $E_b \leftarrow J(\tilde{\mathbf{C}}_{\hat{x}}, \tilde{\mathbf{C}}_{\hat{y}})$
13.               if $(E_b < E_a)$ OR $(e^{-(E_b - E_a)/T} > \text{rand}[0,1])$
14.                  $\mathbf{C}_{\hat{x}} \leftarrow \tilde{\mathbf{C}}_{\hat{x}}$, $\mathbf{C}_{\hat{y}} \leftarrow \tilde{\mathbf{C}}_{\hat{y}}$.
15.         $T \leftarrow \gamma \cdot T$
16. return $\mathbf{C}_{\hat{x}}, \mathbf{C}_{\hat{y}}$
```

Assuming a suitable N-way objective function is used, Algorithm 2 can also be applied to the general N-dataset co-clustering problem by randomly selecting any one of the N dimensions, changing a mapping entry in place of steps 6-11, and appropriately updating the additional mappings in steps 5 and 14. Typically, the closer γ is to 1, the better the results, but at an increased cost in computation time. Fortunately, the number of inner iterations required at step 3 grows linearly as more modalities are added, since we need only touch each entry of each mapping a fixed number of times (3-10 is a good rule of thumb). The algorithm does however require evaluation of the objective, which in turn requires storage of the full joint distribution. The space requirements thus grow exponentially. As mentioned above, we can avoid this difficulty by compromising on higher order interactions through the application of Algorithm 1.

Finally, we acknowledge that Friedman et. al. [13] have proposed a heuristic annealing procedure to optimize a related but single-source information-theoretic clustering objective. In their formulation, the objective function is optimized over an auxiliary partitioning variable T and a Lagrange multiplier β, which controls the tradeoff between compression and maximization of relevant information. The annealing "temperature" in this case is characterized by $1/\beta$. In comparison, Algorithm 2 takes a stochastic, as opposed to deterministic, approach to annealing, and directly optimizes over the relevant mapping functions. The procedure proposed above is both conceptually and algorithmically simpler, and is easily modified to handle multi-way co-clustering problems, whereas Friedman's technique is designed for single-source clustering applications. On the downside, it is more computationally intensive than the algorithm due to Friedman et. al.

2.3 Multi-Source Spectral Clustering

Given a 2-way contingency table of the sort defined in Section 1.4.1, we can apply spectral methods as well. If we consider the joint probability of occurrence between two random variables to be a sort of similarity measure, then the normalized contingency table can be clustered directly as if it were a Gram matrix. Furthermore, for many applications we are concerned only with clustering a primary dataset using additional sources of information (for which a target function or the correct number of "classes" may not be known). Spectral clustering thus offers a reasonable means by which to cluster joint occurrence data with respect to a single dimension. Following the treatment in [23], we propose Algorithm 3 for spectral clustering of 2-way joint distributions.

Algorithm 3 Multi-Source Spectral Clustering (2-way case)

```
1. Given a contingency table $P \in \mathbb{R}^{m \times n}$, compute the row sums
   $d_i = \sum_j P_{ij}\ i = 1, \ldots, m$.
2. Normalize the rows and columns so that $L_{ij} = \frac{P_{ij}}{\sqrt{d_i d_j}}$.
3. Find orthogonal eigenvectors $v_1, \ldots, v_{m'}$ corresponding to the $m'$ largest
   eigenvalues $\lambda_1 \geq \cdots \geq \lambda_{m'}$ satisfying $Lv_i = \lambda_i v_i,\ i = 1, \ldots, m'$.
4. Collect the embedded points $y_k = (v_{1k}, \ldots, v_{m'k}),\ k = 1, \ldots, m$, and cluster
   them with the K-means algorithm.
5. Set the mapping entry $C_x(i) = j$ if the point $y_i$ was assigned to cluster $j$.
6. Repeat from Step 1 given the contingency table $P^T$ to get the mapping $C_y$ if
   desired.
```

Note that because we perform spectral clustering on the contingency matrix, the corresponding eigenproblem is drastically simpler than in the case where the input is a true Gram matrix consisting of kernel products between all pairs of data points. In that situation we require $\mathcal{O}(n^2)$ distance or kernel product computations just to get the Gram matrix, and approximately $\mathcal{O}(n^3)$ floating point operations to compute the eigenvectors for an $n \times n$ matrix. If we have many high dimensional data points, as is typically the case with, say, image features, then this difference in computation time is enormous. The fact that contingency tables are in general non-symmetric, does not substantially impact this computational advantage. For the human identification experiments conducted in Chapter 4, full spectral clustering based on kernel products is effectively intractable.

As before, we observe that the application of Algorithm 3 to N-way problems can be approximated by interleaved clustering of the 2-way marginals of the N-way joint. In addition, if both the row and column mappings of a 2-way marginal $\mathbf{P}$ are desired, then Algorithm 3 can be carried out on both $\mathbf{P}$ and $\mathbf{P}^T$ separately to obtain C_x and C_y respectively. We will discuss further how one might interleave the spectral co-clustering algorithm in Chapter 4.

The drawbacks to Algorithm 3 are few but important. Firstly, spectral clustering embeds the training points in a space that is sensitive and specific to the chosen kernel. If the chosen embedding cannot map the original points (ostensibly on a nonlinear manifold) into linear subspaces, then we won't necessarily be any better off with spectral clustering. The effectiveness thus rests on the kernel, and in particular on any kernel parameters which must be carefully tuned. In the case of contingency data, we do not have the option of selecting from a variety of kernels, and must therefore ensure that the given co-occurrence matrix is suitably normalized. Secondly, it is worth noting that Algorithm 3 applied to contingency data is an inherently 2-way mechanism: we cannot explicitly take advantage of interactions higher than order two.

Chapter 3

Multimodal Data Clustering by Continuous Optimization

In this chapter we will explore optimization of the nonlinear information-theoretic objective function discussed in section 1.4.2, using classical multivariable techniques which require differentiability of the problem to be solved. The goal here will not be to propose the world's best algorithm for co-clustering, but rather, to formulate and examine direct continuous optimization applied to co-clustering. The following analysis will provide insight into the nature of multi-source clustering, and we will make the constraints and limitations specific to clustering of contingency data explicit. The optimization framework presented below is noteworthy in that it facilitates the development of tailor-made co-clustering algorithms that can incorporate almost any problem-specific criteria. In this regard the technique is far more general than any of the methods presented in Chapter 2. Furthermore, by formulating co-clustering as a continuous optimization problem, we are able to additionally evaluate the effectiveness of a unique class of co-clustering algorithms that represent a departure from the algorithmic themes which dominate classical clustering techniques. Unlike many popular clustering methods, the optimization problem described below does not involve cluster prototypes or minimization of a distortion metric over individual points. In Chapter 4, we will evaluate empirically the viability of this approach, and discuss its strengths and weaknesses therein.

3.1 A Weight-Based Formulation

In this section we will develop a real-valued weight based approach to co-clustering, in which we optimize the information-theoretic objective function over a factored set of real-valued weights using standard gradient descent procedures. We will additionally discuss the time and space requirements of the weight-based approach and give matrix-vector definitions to facilitate implementation.

Because many clustering objective functions do not depend smoothly on the cluster mappings, continuous optimization techniques such as gradient descent cannot be used unless the optimization problem is appropriately reformulated. Recall that, because $I(X;Y)$ is fixed, minimizing (1.2) is equivalent to

$$\max_{C_x, C_y} \sum_{\hat{x},\hat{y}} P(\hat{x},\hat{y}) \log \frac{P(\hat{x},\hat{y})}{P(\hat{x})P(\hat{y})}.$$

In words, this says that a good clustering should maximize the mutual information between the modalities. The problem, however, lies in the definition of $P(\hat{x},\hat{y})$ (shown in (1.1)), since we can't optimize in a continuous fashion over the integer-valued mappings C_x, C_y. If we can write down a definition of $P(\hat{x},\hat{y})$ that depends smoothly on a function from which cluster memberships can be recovered, then continuous optimization can proceed.

We'll begin with a modification of the joint cluster distribution (1.1) involving a weight factorization that allows recovery of individual modality mapping functions:

$$P(\hat{x},\hat{y}) = \sum_{x,y} w^r_{\hat{x},x} w^c_{\hat{y},y} P(x,y). \tag{3.1}$$

The summations are now over all rows x and columns y in the joint distribution $P(x,y)$, and the members of a given cluster $(\hat{x},\hat{y})$ are picked out by the sets of real-valued row and column weights $\{w^r_{\hat{x},x}\}$ and $\{w^c_{\hat{y},y}\}$ respectively. Given the co-clustering definition in section 1.4.1, the number of weights to be found given the factorization (3.1) is equal to $m \cdot m' + n \cdot n'$, as compared to $m \cdot m' \cdot n \cdot n'$ in the general case. The (preliminary) continuous maximization problem can now be stated as:

$$\max_{\mathbf{w}^r,\mathbf{w}^c} \sum_{\hat{x},\hat{y}} P(\hat{x},\hat{y}) \log \frac{P(\hat{x},\hat{y})}{P(\hat{x})P(\hat{y})} \tag{3.2}$$
$$\text{subject to } 0 \le w^r_{\hat{x},x}, w^c_{\hat{y},y} \le 1,\ \forall \hat{x},\hat{y},x,y.$$

The constraints indicated in Equation (3.2) are incomplete however: we will additionally need to further constrain the weights $\mathbf{w}^r$ and $\mathbf{w}^c$ in order to bias the solution towards something from which we can recover the cluster mappings C_x and C_y. The following clustering-specific criteria offer some clues as to what kind of constraints we might want to include:

1. Each point can only be assigned to one cluster.

2. Each point must be assigned to a cluster.

3. Each cluster should have at least one point.

While it would seem ideal from a clustering standpoint to ensure that all of the rules are strongly enforced, in practice we may need only one or two weakly enforced constraints to give sufficient interpretability of the solution. In addition, when applying gradient methods to non-convex optimization problem such as (3.2), the introduction of multiple constraints can at times give poorer solutions and prevent convergence. When additional constraints are combined with the original objective via Lagrange multipliers, we arrive at what's called the *dual* problem. It could also be the case that the optimal solution to a dual program incorporating many constraints may not correspond to an optimal solution for the original unconstrained objective due to the existence of a large duality gap (dependent on the nature of the constraints, see e.g. [3]). We will show how to quantify each of the clustering criteria in terms of constraints on the row and column weights and develop the algorithm in full generality. However, depending on the application, it may not be necessary or desirable to include them all, and in those cases the unwanted constraint terms may be dropped from the following development.

We can encode the first condition in the form of equality constraints which state that only one weight for a given point can be non-zero across row/column clusters:

$$\sum_{i,j>i} w^r_{\hat{x}=i,x} w^r_{\hat{x}=j,x} = 0,\ \forall x \tag{3.3}$$

$$\sum_{i,j>i} w^c_{\hat{y}=i,y} w^c_{\hat{y}=j,y} = 0,\ \forall y. \tag{3.4}$$

The second condition is enforced by the previous constraints, plus the additional requirement that the weights for a given point must sum to one across row/column clusters:

$$\sum_{\hat{x}} w^r_{\hat{x},x} = 1,\ \forall x \tag{3.5}$$

$$\sum_{\hat{y}} w^c_{\hat{y},y} = 1,\ \forall y. \tag{3.6}$$

This constraint also has the added benefit of encouraging the weights to take on binary values since we require that the summations equal 1. Lastly, the third clustering condition can be transformed into constraints similar to the previous set, except with summations over the other (x,y) dimension of the weights, and the possibility that more than one point can be assigned to a cluster:

$$\sum_{x} w^r_{\hat{x},x} \geq 1,\ \forall \hat{x} \tag{3.7}$$

$$\sum_{y} w^c_{\hat{y},y} \geq 1,\ \forall \hat{y}. \tag{3.8}$$

Combining these constraints with the original problem (3.2) gives the following Lagrangian:

$$\begin{aligned}
L(\mathbf{w}^{r/c}, \boldsymbol{\lambda}^{r/c}, \boldsymbol{\xi}^{r/c}, \boldsymbol{\zeta}^{r/c}, \boldsymbol{\alpha}^{r/c}) = & \sum_{\hat{x},\hat{y}} P(\hat{x},\hat{y}) \log \frac{P(\hat{x},\hat{y})}{P(\hat{x})P(\hat{y})} \\
& - \sum_{x=1}^{m} \lambda^r_x \Big(\sum_{i,j>i} w^r_{i,x} w^r_{j,x} \Big) - \sum_{y=1}^{n} \lambda^c_y \Big(\sum_{i,j>i} w^c_{i,y} w^c_{j,y} \Big) \\
& - \sum_{x=1}^{m} \xi^r_x \Big(\sum_{\hat{x}} w^r_{\hat{x},x} - 1 \Big) - \sum_{y=1}^{n} \xi^c_y \Big(\sum_{\hat{y}} w^c_{\hat{y},y} - 1 \Big) \\
& - \sum_{\hat{x}=1}^{m'} \zeta^r_{\hat{x}} \Big(\sum_{x} w^r_{\hat{x},x} - 1 \Big) - \sum_{\hat{y}=1}^{n'} \zeta^c_{\hat{y}} \Big(\sum_{y} w^c_{\hat{y},y} - 1 \Big), \\
& - \sum_{\hat{x}=1}^{m'} \alpha^r_{\hat{x}} \zeta^r_{\hat{x}} - \sum_{\hat{y}=1}^{n'} \alpha^c_{\hat{y}} \zeta^c_{\hat{y}}
\end{aligned} \tag{3.9}$$

which is maximized with respect to the row/column weights $\mathbf{w}^{r/c}$, and the row/column constraint Lagrange multipliers $\boldsymbol{\lambda}^{r/c}, \boldsymbol{\xi}^{r/c}, \boldsymbol{\zeta}^{r/c}$, and $\boldsymbol{\alpha}^{r/c}$. A word on notation: from here on we will drop the tedious notation $\hat{x} = u, x = v$ and simply assume that first (hatted) subscripts index clusters and second subscripts index rows or columns.

In order to proceed with optimization, we take derivatives with respect to the row and column weights. Focusing just on the first (risk) term in equation (3.9), if we substitute the full definition (3.1) in for $P(\hat{x}, \hat{y})$ we see that

$$I(\hat{X}, \hat{Y}) = \sum_{\hat{x},\hat{y},x,y} w^r_{\hat{x},x} w^c_{\hat{y},y} P(x,y) \log \frac{\sum_{x',y'} w^r_{\hat{x},x'} w^c_{\hat{y},y'} P(x',y')}{\sum_{\hat{y}',x',y'} w^r_{\hat{x},x'} w^c_{\hat{y}',y'} P(x',y') \sum_{\hat{x}',x',y'} w^r_{\hat{x}',x'} w^c_{\hat{y},y'} P(x',y')}.$$

Skipping a great deal of algebra and simplifying, the derivative of $I(\hat{X}, \hat{Y})$ with respect to the row weights is then

$$\begin{aligned}
\frac{\partial\, I(\hat{X}, \hat{Y})}{\partial w^r_{\hat{u},v}} &= \sum_{\hat{y},x,y} \frac{w^r_{\hat{u},x} w^c_{\hat{y},y} P(x,y)}{P(\hat{u},\hat{y})} \Big(\sum_{y'} w^c_{\hat{y},y'} P(v,y') \Big) \\
&+ \sum_{\hat{y},y} w^c_{\hat{y},y} P(v,y) \log P(\hat{u},\hat{y}) \\
&- \sum_{\hat{y},y} w^c_{\hat{y},y} P(v,y) \log P(\hat{u}) P(\hat{y}) \\
&- \sum_{\hat{x},\hat{y},x,y} \frac{w^r_{\hat{x},x} w^c_{\hat{y},y} P(x,y)}{P(\hat{y})} \Big(\sum_{y'} w^c_{\hat{y},y'} P(v,y') \Big) \\
&- \sum_{\hat{y},x,y} \frac{w^r_{\hat{u},x} w^c_{\hat{y},y} P(x,y)}{P(\hat{u})} \Big(\sum_{\hat{y}',y'} w^c_{\hat{y}',y'} P(v,y') \Big) \\
&= \sum_{\hat{y},y} w^c_{\hat{y},y} P(v,y) \left[\log \left(\frac{P(\hat{u},\hat{y})}{P(\hat{u})P(\hat{y})} \right) - 1 \right].
\end{aligned} \tag{3.10}$$

Similarly, the derivative with respect to the column weights reduces to

$$\frac{\partial\, I(\hat{X}, \hat{Y})}{\partial w^c_{\hat{u},v}} = \sum_{\hat{x},x} w^r_{\hat{x},x} P(x,v) \left[\log \left(\frac{P(\hat{x},\hat{u})}{P(\hat{x})P(\hat{u})} \right) - 1 \right]. \tag{3.11}$$

Turning to the constraint terms, which we will collectively denote by "C", the other derivatives needed for optimization are

$$\frac{\partial C}{\partial w^r_{\hat{u},v}} = \lambda^r_v \sum_{\hat{\imath} \neq \hat{u}} w^r_{\hat{\imath},v} + \xi^r_v + \zeta^r_{\hat{u}}$$
$$\frac{\partial C}{\partial w^c_{\hat{u},v}} = \lambda^c_v \sum_{\hat{\imath} \neq \hat{u}} w^c_{\hat{\imath},v} + \xi^c_v + \zeta^c_{\hat{u}}$$

for the weights, and

$$\begin{aligned} \frac{\partial C}{\partial \lambda^r_v} &= \sum_{\hat{\imath},\hat{\jmath}>\hat{\imath}} w^r_{\hat{\imath},v} w^r_{\hat{\jmath},v} & \qquad \frac{\partial C}{\partial \lambda^c_v} &= \sum_{\hat{\imath},\hat{\jmath}>\hat{\imath}} w^c_{\hat{\imath},v} w^c_{\hat{\jmath},v} \\ \frac{\partial C}{\partial \xi^r_v} &= \sum_{\hat{x}} w^r_{\hat{x},v} - 1 & \qquad \frac{\partial C}{\partial \xi^c_v} &= \sum_{\hat{y}} w^c_{\hat{y},v} - 1 \\ \frac{\partial C}{\partial \zeta^r_{\hat{u}}} &= \sum_{x} w^r_{\hat{u},x} - 1 & \qquad \frac{\partial C}{\partial \zeta^c_{\hat{u}}} &= \sum_{y} w^c_{\hat{u},y} - 1 \\ \frac{\partial C}{\partial \alpha^r_{\hat{u}}} &= \zeta^r_{\hat{u}} & \qquad \frac{\partial C}{\partial \alpha^c_{\hat{u}}} &= \zeta^c_{\hat{u}} \end{aligned}$$

for the clustering constraint Lagrange multipliers.

Lastly, we enforce the inequalities $0 \leq w^r_{\hat{x},x}, w^c_{\hat{y},y} \leq 1$, $\forall \hat{x}, \hat{y}, x, y$ by setting each weight (generically denoted by w) equal to the logistic function,

$$w = \frac{1}{1 + e^{-\gamma}} = g(\gamma),$$

and performing unconstrained optimization over the alternate logistic variables γ. Given the above derivatives, we can now define the gradient descent optimization procedure shown in Algorithm 4, where we have noted with "r/c" that the row and column versions of the update rules are separate but similar. As mentioned before, it should be noted that although the algorithm shows updates for each of the constraints, in practice some may be unnecessary or even counterproductive.

For the convergence criterion, one reasonable rule might simply terminate the iteration when the absolute improvement in the objective is less than a prespecified tolerance value, e.g. when

$$\left| J\big(\mathbf{w}^r(n+1), \mathbf{w}^c(n+1)\big) - J\big(\mathbf{w}^r(n), \mathbf{w}^c(n)\big) \right| < \epsilon.$$

Algorithm 4 Gradient Descent for Weight-Based Multi-Source Clustering (2-way case)

```
Randomly initialize the weights and Lagrange multipliers. Set $n \leftarrow 0$.
do
    for all weights and Lagrange multipliers
        $\gamma_{\hat{u},v}^{r/c}(n+1) = \gamma_{\hat{u},v}^{r/c}(n) + \eta_\gamma \left( \frac{\partial I(\hat{X},\hat{Y})}{\partial w_{\hat{u},v}^{r/c}} - \frac{\partial C}{\partial w_{\hat{u},v}^{r/c}} \right) w_{\hat{u},v}^{r/c}(n) \big(1 - w_{\hat{u},v}^{r/c}(n)\big)$
        $\lambda_{\hat{u},v}^{r/c}(n+1) = \lambda_{\hat{u},v}^{r/c}(n) + \eta_\lambda \frac{\partial C}{\partial \lambda_{\hat{u},v}^{r/c}}$
        $\xi_v^{r/c}(n+1) = \xi_v^{r/c}(n) + \eta_\xi \frac{\partial C}{\partial \xi_v^{r/c}}$
        $\zeta_{\hat{u}}^{r/c}(n+1) = \zeta_{\hat{u}}^{r/c}(n) + \eta_\zeta \frac{\partial C}{\partial \zeta_{\hat{u}}^{r/c}}$
        $\alpha_{\hat{u}}^{r/c}(n+1) = \alpha_{\hat{u}}^{r/c}(n) + \eta_\alpha \frac{\partial C}{\partial \alpha_{\hat{u}}^{r/c}}$
        $w_{u,v}^{r/c}(n+1) = \left(1 + e^{-\gamma_{u,v}^{r/c}(n+1)}\right)^{-1}$
    $n \leftarrow n + 1$
until convergence or $n > n_{max}$
```

In terms of time requirements, the weight based algorithm is relatively tractable. Because the gradient greatly simplifies, each gradient descent iteration in the 2-way case requires $\mathcal{O}\big((m' \cdot m)^2 + (n' \cdot n)^2\big)$ operations. The time requirements of the algorithm thus grows essentially quadratically as we increase the number of rows and columns or as we increase the number of desired clusters. As we add more modalities, the time requirements grow linearly. Unfortunately, storage requirements grow exponentially with higher order co-occurrences since the full joint must be represented. In the case of many modalities, the algorithm could possibly operate on a limited number of dimensions at a time via an interleaving scheme such as Algorithm 1, but where successive applications of the algorithm inherit initial weights from previous runs in place of mappings. As we will see empirically in Chapter 4 however, the performance of weight-based clustering is somewhat sensitive to the size of the input matrix, and it is likely that results would be unacceptably poor if the input matrix came close to exhausting the memory capabilities of most modern desktop computers anyhow. The algorithm is not particularly sensitive to the desired number of clusters along any dimension or to the number of original data-points used to construct the co-occurrence matrix. In the latter case however, for a fixed number of prototype vectors, quantization distortion is likely to increase as more data is used to build the contingency matrix.

3.1.1 Matrix-Vector Definitions

Computation of the update rules is dominated by the derivatives (3.10) and (3.11), while evaluation of the Lagrangian (3.9) is dominated by computation of the primal objective (3.2). We give the relevant equations in matrix-vector form to facilitate efficient computation with a software package such as MATLAB. First, define the weight vectors

$$\begin{aligned} \mathbf{w}^r(\hat{x}) &= [w^r_{\hat{x},1} \ \cdots \ w^r_{\hat{x},m}]^T, \ \forall \hat{x} \\ \mathbf{w}^c(\hat{y}) &= [w^c_{\hat{y},1} \ \cdots \ w^c_{\hat{y},n}]^T, \ \forall \hat{y} \end{aligned}$$

so that the matrix $\mathbf{W}(\hat{x},\hat{y}) \in \mathbb{R}^{m\times n}$ is the outer product $\mathbf{W}(\hat{x},\hat{y}) = \mathbf{w}^r(\hat{x})\mathbf{w}^c(\hat{y})^T$. For the development to follow, $||\cdot||_F$ denotes the Frobenius norm, $\circ$ denotes elementwise multiplication (Hadamard product), and the square-root and log operations are to be taken elementwise as well. The joint $m \times n$ distribution matrix is denoted $\mathbf{P}$.

We can express the the joint cluster distribution terms as:

$$\begin{aligned} P(\hat{x},\hat{y}) &= \mathbf{w}^r(\hat{x})\mathbf{P}\mathbf{w}^c(\hat{y}) \\ P(\hat{x})P(\hat{y}) &= \sum_{i,j} \left(\mathbf{P}^T\mathbf{w}^r(\hat{x})(\mathbf{P}\mathbf{w}^c(\hat{y}))^T\right)_{ij} = \sum_{i,j} \left(\mathbf{P}^T\mathbf{W}(\hat{x},\hat{y})\mathbf{P}^T\right)_{ij} \\ &= \left\|\sqrt{\mathbf{P}^T\mathbf{W}(\hat{x},\hat{y})\mathbf{P}^T}\right\|_F^2 = \mathrm{trace}\Big(\sqrt{\mathbf{P}\mathbf{W}(\hat{x},\hat{y})^T\mathbf{P}}\sqrt{\mathbf{P}^T\mathbf{W}(\hat{x},\hat{y})\mathbf{P}^T}\Big). \end{aligned}$$

If we then define

$$\begin{aligned} &\hat{\mathbf{P}}_{ij} = P(\hat{x}=i,\hat{y}=j) \ \text{ and} \\ &\mathbf{U}_{ij} = \mathrm{trace}\Big(\sqrt{\mathbf{P}\mathbf{W}(\hat{x}=i,\hat{y}=j)^T\mathbf{P}}\sqrt{\mathbf{P}^T\mathbf{W}(\hat{x}=i,\hat{y}=j)\mathbf{P}^T}\Big), \end{aligned}$$

the primal objective can be written as

$$I(\hat{X};\hat{Y}) = \sum_{i,j} \hat{\mathbf{P}}_{ij} \log \frac{\hat{\mathbf{P}}_{ij}}{\mathbf{U}_{ij}} = \left\|\sqrt{\hat{\mathbf{P}} \circ \log \hat{\mathbf{P}}}\right\|_F^2 - \left\|\sqrt{\hat{\mathbf{P}} \circ \log \mathbf{U}}\right\|_F^2,$$

which can be also be computed as a difference of traces if desired. Lastly, using these definitions, the derivatives (3.10) and (3.11) can be expressed similarly. First, form the matrices $\mathbf{W}^r \in \mathbb{R}^{m' \times m}$ and $\mathbf{W}^c \in \mathbb{R}^{n' \times n}$ by inserting as rows the weight vectors $\mathbf{w}^r(\hat{x})\ \forall \hat{x}$ and $\mathbf{w}^c(\hat{y})\ \forall \hat{y}$ respectively. Then,

$$\frac{\partial I(\hat{X},\hat{Y})}{\partial w^r_{\hat{u},v}} = \mathbf{W}^r \mathbf{p}_v^T \cdot \left(\log \hat{\mathbf{p}}_{\hat{u}}^T - \log \mathbf{u}_{\hat{u}}^T - \mathbf{1}\right)$$
$$\frac{\partial I(\hat{X},\hat{Y})}{\partial w^c_{\hat{u},v}} = \mathbf{W}^c \mathbf{p}_v \cdot \left(\log \hat{\mathbf{p}}_{\hat{u}} - \log \mathbf{u}_{\hat{u}} - \mathbf{1}\right),$$

where $\mathbf{p}_i$ denotes the i-th column of $\mathbf{P}$, $\mathbf{p}_i^T$ denotes the i-th row, and a similar convention is followed for $\mathbf{U}$ and $\hat{\mathbf{P}}$. We finally note that since the matrices $\mathbf{P}$ and $\mathbf{W}$ are typically small, the matrix products, matrix-vector products, and norms above are all relatively cheap computationally.

Chapter 4

Experimental Application to Multi-Source Human Identification

4.1 Motivation

The application of co-clustering to human identification is motivated by the observation that many machine systems for human identification (e.g. [22]) suffer from the significant limitation that they rely on human "experts" to label descriptive training data for the subjects to be identified. While a supervised approach allows for accurate detection and classification from the onset, it also necessarily hinders the system's ability to accommodate new individuals and adapt to changing world conditions autonomously. For many realistic applications, subjects may not be known to a system prior to deployment, while the pool of individuals we might want to recognize is often in a constant state of flux. It would thus be highly desirable for a system to be able to independently learn separate subject models from little or even zero prior knowledge, and to additionally recognize when incoming measurements correspond to a hitherto undiscovered class (e.g. a new person).

Here, we will cast multi-source human identification as an unsupervised co-clustering problem that can contribute to the larger goal of classification and autonomous construction of subject models in the absence of data normalization. More specifically, by applying multi-source clustering we hope to identify coherent structures in the data that could be used to assign labels according to distinct groups which, ideally, correspond to subject identities. The automatic construction of user models and subsequent classification of unseen data is then made possible: after augmenting the original data with labels returned by the co-clustering process, we can simply train classifiers using a broad range of supervised learning methods.

If the particular application requires the labels to be semantically grounded in the real world, the system can ask a human expert a small set of "questions" in order to attach meaning to the labels. When the data is accurately clustered according to subject and the number of desired clusters is either known or close to the true value, the number of questions will be small. Conversely, if we are forced to overestimate the number of subjects because the true number is not known, or if the clustering does not bear sufficient correspondence to the true classes, then more cluster labels will need to be tied to (possibly repeated) physically significant descriptions. In general, multi-source clustering can facilitate active learning by significantly reducing the number of questions put to a human supervisor, as compared to simply labeling an entire dataset.

In the section that immediately follows, we will describe the identification system that served as the experimental platform in this chapter. We then discuss briefly the difficulties one faces in identifying individuals from multi-source features. Our description of the experimental setup is concluded with a discussion of multi-source clustering within the context of human identification. Finally, and most importantly, we conduct several experiments which compare and analyze performance of the algorithms outlined in Chapters 2 and 3, as applied to clustering tasks of varying difficulty and complexity. We provide a comparison to the co-clustering algorithm proposed by Dhillon et. al. [11], and finish with an evaluation of how noise affects the performance of contingency-based clustering.

4.2 Experimental Platform

The experiments to follow make use of multi-source sensory data collected by an identification system designed by Kim, Ivanov, and Poggio [19]. The data is organized according to four audio and visual sensory channels:

- Subject's *height*: measured using a single stationary, calibrated color camera.
- Subject's *clothing* preferences: accumulated by computing separate 16x16x3 color histograms from the top and bottom halves of body images within each day. We will refer to these data channels as the *upper* and *lower* modalities, respectively.
- Subject's *face*: 50x50 grayscale patches detected and extracted when available using a technique proposed by Heisele et. al. [14].
- Subject's *voice*: recorded with a stationary microphone array and encoded by MEL-scale frequency cepstral coefficients (MFCCs) in order to capture the defining frequencies of each subject's voice. Because voice information is not always present, and can be unreliable due to background interference, we will not use this channel in the experiments to follow.

At each sampling step, the system computes and records upper/lower histograms and height measurements as long as a person is present and unobscured. If the subject's face is detected (that is, the face is visible and the pose is sufficiently frontal), a patch is extracted, normalized to a canonical scale, and stored. As a rudimentary form of temporal alignment, we take samples from only those time steps where *all* modalities are present. The data available to be clustered can therefore be organized into four distinct subsets, each specific to a sensory channel, but related by time step. A subset of the dataset along with examples from each audio/visual modality is illustrated diagrammatically in Figure 4-1. The time steps $t = 0, 1, 2, ...$ shown correspond to temporally aligned intervals resulting from the alignment process described above, while heights along the last row represent single calibrated measurements taken at that instant. Columns in the figure have been partitioned according to sequences which were known to have been recorded at different times, and bright patches seen in the example histograms correspond to clothing colors that particularly dominated the subject's wardrobe that day. The fact that two visibly different subjects are shown in the sequences is coincidental; it is almost always the case that multiple sequences describe the same person and we could have easily shown two sequences of the same person taken from two separate days.

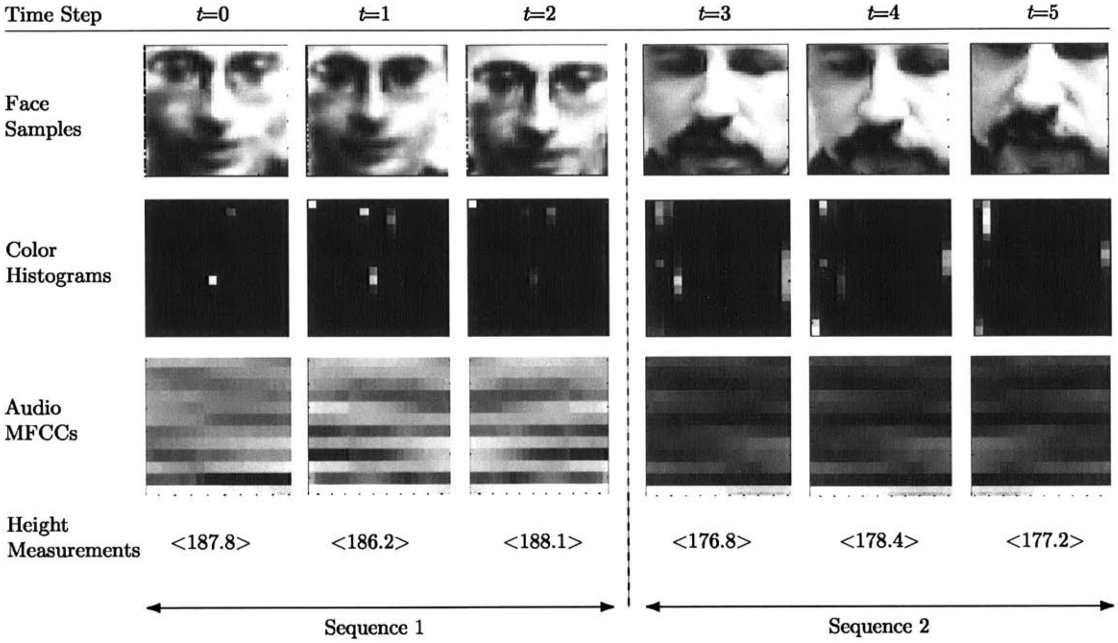

Figure 4-1: Examples of audio and visual feature channels are shown temporally aligned and partitioned into sequences corresponding to two different individuals presented before the identification system.

4.2.1 Challenges

Several potential difficulties immediately stand out: many of the modalities we wish to cluster cannot be expected to naturally group according to subject, and include overlapping or oddly shaped clusters. For example, clothing histograms may not cluster according to individual as we would expect faces to. Furthermore, data recorded from the surveillance system tends to include a certain degree of noise in the form of errors in the sensory measurements: heights can be inaccurate due to camera calibration issues, while spurious patches are occasionally extracted as faces. In addition, the presence of multiple individuals in a single frame can incorrectly commingle modalities during a given sampling interval, while subjects holding objects such as papers or cups effectively cause corrupting noise to be added to clothing histograms. We have attempted to minimize such sources of noise somewhat, but do not systematically weed out every error. It is our hope to approximate a reasonably authentic surveillance scenario, and we thus allow some noise to remain.

A final difficulty worth considering is the problem of "who to trust." A priori, we cannot rule out the possibility that weak channels might corrupt stronger ones when clustered simultaneously. Knowing precisely when to trust or distrust a given modality is a challenge that is beyond the scope of this thesis. However, the empirical results to follow suggest that, as far as co-clustering is concerned, this knowledge may not be as critical as one might expect.

4.2.2 Multi-Source Clustering and Human ID

Without doing any experiments, we can reasonably expect that clustering massive collections of high-dimensional video features will be difficult using almost any method. It is therefore critically important that we exploit all available information regarding the circumstances under which the data was collected, as well as from the data itself. We therefore aim to include two additional pieces of information:

- All data points (regardless of sensory channel) belong in the same respective cluster if they are from the same temporal sequence. For example, if we extract 100 faces over two minutes of video, then all of those faces ought to be part of the same cluster since we assume that there can only be one subject under surveillance at a time. Examples of such sequences are marked near the bottom of Figure 4-1. In practice, we would not know that two sequences correspond to two separate individuals per say, however information designating preliminary partitionings of temporally aligned features into sequences is available and exploited in the experiments that follow.

- We allow information from strong channels to help weaker ones by allowing modalities to interact during clustering (thanks to multi-source clustering). This concept is illustrated in Figure 4-2.

Clustering by sequence also has the important added benefit that it avoids feature resolution and temporal alignment difficulties which would arise in a joint feature space, or if all data points were simply clustered together without regard to temporal co-occurrence. Experiments in which faces alone were hierarchically clustered by sequence showed a dramatic improvement in accuracy (and speed), compared to simply clustering a giant pool of individual faces.

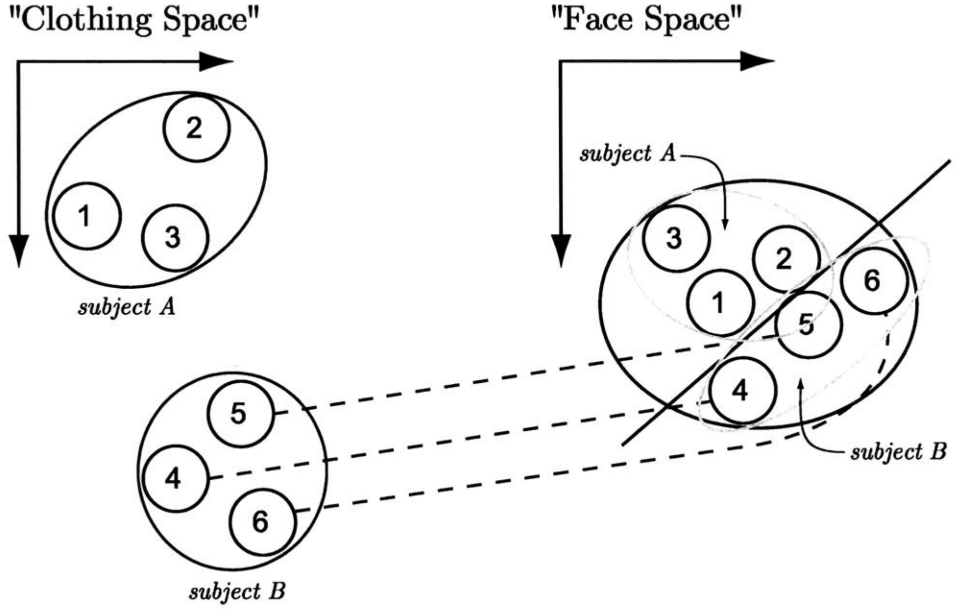

Figure 4-2: In this idealized example, sequences 4,5, and 6 in the space of clothing histograms provide information about how faces ought to be clustered. The dashed lines indicate temporal correspondences between modalities, and suggest that the large cluster enclosing all of the face sequences should be split and re-labeled into two smaller clusters.

4.3 Experiments

In the present section we show the results of the algorithms discussed in this thesis when applied to several human identity mining tasks of varying difficulty, and additionally compare performance to Dhillon's algorithm [11]. The goal of the experiments that follow will be to realize an encoding of the data that groups points into classes which correspond to the true subject classes. For simplicity we will assume that the correct number of subjects is known a priori, and will not address cluster selection or investigate algorithms for determining the number of clusters. Because faces can reasonably be assumed to group according to subject, we will evaluate the accuracy of a given encoding by calculating the extent to which clusters in the face channel represent subject classes. The reported "error" incurred by a clustering will therefore summarize the fraction of points known to represent a particular subject that are grouped together by the encoding.

4.3.1 Datasets

For most of the experiments in this chapter we have used the same two datasets. The first represents features corresponding to 10 different subjects, and consists of 19,136 points divided into 32 sequences, while the second is a smaller, 5 subject database consisting of 7,379 points and 27 sequences. The data occurs in three channels: faces, upper-body histograms, and lower-body histograms. In the future will simply refer to them as the *"10 subject dataset"* and the *"5 subject dataset"*.

4.3.2 Seeded Agglomerative Clustering

As a helpful basis for comparison, we attempted to cluster faces into both 5 and 10 categories using single-source techniques, while working in knowledge of the sequences. Before we even begin clustering, we know from the sequences that certain collections of points belong together. Hierarchical agglomerative clustering provides a natural interface for incorporating exactly this sort of information: we can simply start the agglomeration process from initial clusters that correspond to the sequences. In this case, most of the clustering has already been done for us, and it is as if we are only performing the last few merges of a clustering process that begins with individual points and ends with clusters that hopefully correspond to subject identities.

Figure 4-3 illustrates the process schematically, where we show the agglomeration process starting with 6 initial sequences, and ending at 2 desired clusters. Because the seeded agglomerative procedure is top-down, this method rigidly enforces the sequence groupings in the sense that points from the same sequence will always be in the same final cluster together. In addition, it is usually the case that the number of sequences is far less than the number of original datapoints – for these experiments, approximately 100-1500 points fall into each sequence. This fact implies that the seeded agglomerative clustering technique is markedly faster than simply clustering the individual data points without any initial groupings. In general, if we have N points and K desired clusters, then hierarchical clustering will require $N - K$ mergings. Thus, the computation time is reduced by a factor roughly equal to the ratio of original points to sequences.

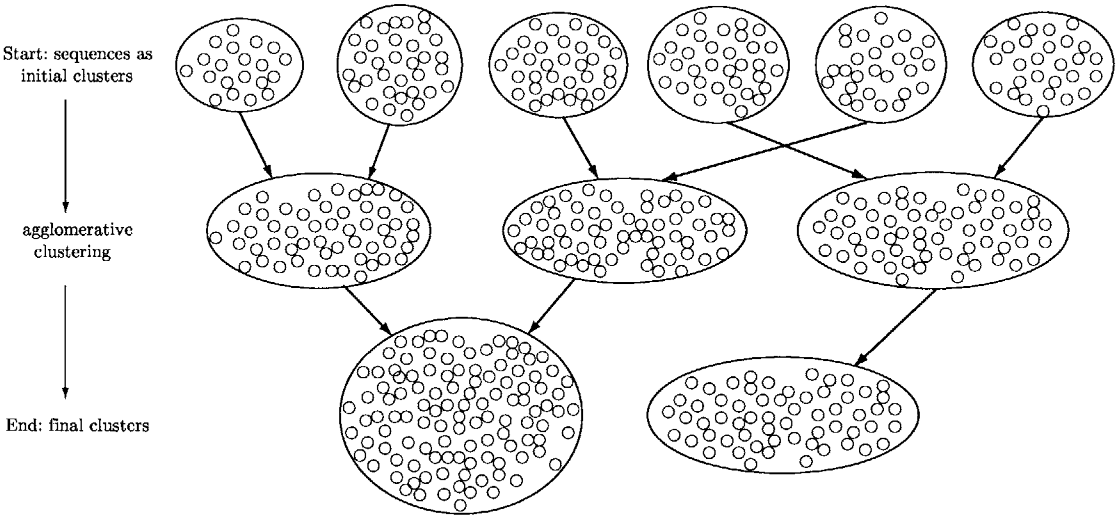

Figure 4-3: Illustration of seeded agglomerative clustering. In the example shown, we start with 6 initial clusters corresponding to sequences, and merge until finishing with 2 final clusters.

In Table 4.1 we show the results of several agglomerative clustering trials applied to face images. The first section reports the results when applied to the 10 subject database, where each data point was reduced from 2500 to 30 features with Principal Components Analysis (PCA). The clustering process was seeded with all 32 sequences as the initial clusters. The second section shows performance when applied to the 5

Dataset	Linkage	Assignment Error
5 subject	single (min)	59.1% (4359)
5 subject	complete (max)	36.2% (2668)
5 subject	average (mean)	**13.1%** (963)
10 subject	single (min)	**22.8%** (4359)
10 subject	complete (max)	57.0% (10900)
10 subject	average (mean)	30.7% (5868)

Table 4.1: Seeded agglomerative clustering applied to PCA face components ($N = 30$) alone. Performance is shown for both the 5 and 10 subject datasets, and for three common choices of the distance metric.

subject database, where we now have 27 initial sequences. From the results shown, the average-distance metric fared the best for the 5 subject dataset with approximately 13% error, while the minimum point-to-point distance (single linkage) gives the smallest number of assignment mistakes, at 22.8% of the total, on the 10 subject dataset. Interestingly, in the single-linkage case the same errors are made on both the 10 and 5 user databases, indicating that some of the sequences are much more prone to grouping by subject identity than others when using this linkage. In addition, it is worth noting that the best linkage choice is different for the two datasets, and that there is no clear best distance one can choose a priori. Because labels are rarely available in practice, the experimenter might thus investigate the resulting intra-cluster distortion or a similar cluster coherence measure in order to determine which linkage is giving better results for a given dataset.

4.3.3 Pre-Processing, Post-Processing, and Error Accounting

Quantization and Construction of the Contingency Matrix

Unfortunately, for the co-clustering algorithms discussed in this chapter, valuable sequence information cannot be explicitly incorporated as is possible with agglomerative clustering. To see why, recall that the normalized contingency table functions as a discrete approximation to the joint distribution over random variables describing the input feature sets. If we simply built a table of co-occurrences between sequences (a diagonal matrix in the 2-way case), then we would not have an even-handed discretization of the space in which the input points lie. Counting co-occurrences between sequences alone says nothing about how the feature sets are distributed. As an example, consider two sequences that "sit" almost on top of one another in the space of faces, but are cleanly "separated" in the height space. When we go to count the number of co-occurrences involving each of the height sequences, we compare intersections only between full sequences, resulting in values of zero for every comparison that does not involve the same sequence and the number of points in the sequence otherwise. In doing so, we neglect the information that the two sequences overlap *in space* and thus do not provide an accurate characterization of that space. In building the joint probability distribution, we must therefore tradeoff sequence knowledge against coverage of the input spaces. On the other end of the spectrum, we could simply perform K-means clustering with K chosen large as a means by which to discretize a feature space.

With this tradeoff in mind, we incorporate sequence knowledge into the clustering process in two "implicit" ways, and emphasize that this difference should be remembered when comparing co-clustering results to seeded agglomerative performance figures. Firstly, when building the contingency matrix, we organize each set of features into temporal sequences as described above, and take the mean of points within each sequence to be the set of vector prototypes or "codebook" for that modality. Each point, irrespective of sequence membership, is then assigned to the prototype that is closest in Euclidean distance. Because we are using 30 PCA components and not the full high-dimensional points, the Euclidean distance is a reasonable choice here. In the 2-way case, the input joint distribution matrices will thus be square and will have dimensions m and n equal to the number of sequences. Each entry in the co-occurrence matrix is then the number of points lying at the intersection of the two sets of points assigned to each corresponding pair of prototypes, for all pairs of row and column prototypes. The table of co-occurrences is then normalized to unity in order to become an admissible probability distribution.

It should be noted that, because each point is associated with a prototype regardless of sequence membership, we will incur quantization error: whenever sequences overlap, there is an increased possibility of quantization mistakes in the form of points assigned to the wrong sequence prototype. Despite the addition of these quantization effects, we have found that this technique yields better results empirically compared to either performing vector quantization on the original data using no sequence information, or compared to using sequence correspondences alone. This indicates that we have indeed struck a balance between accurate discretization of the input spaces, and utilization of sequence knowledge.

Post-Processing with Sequences

The second way in which sequence information is exploited comes after clustering of the co-occurrence matrix. When co-clustering has completed, we reassign each point to the cluster containing the majority of samples from the sequence to which the point belongs. In the case where quantization as described above misdirects less than half of the points in a sequence, reassignment will effectively "undo" each of those mistakes. If more than half of the points have been assigned incorrectly, then reassignment will only make matters worse. In practice, however,this post-processing step often greatly reduces the error initially incurred via quantization rather than exacerbate it.

Error Definitions

In the trials that follow, we will incur four kinds of "error", where error is loosely defined as the extent to which points group according to subject:

1. Before we cluster anything we will see error due to quantization alone, or *quantization error* (QE). This error is defined as the number of points that have been assigned to the wrong sequence prototype (i.e. to a prototype that does not correspond to the sequence from which a given point came from).

2. After clustering, but before reassignment post-processing, we will observe the *total error* (TE). The total error is derived from a "confusion matrix" where

entry (i, j) corresponds to the number of points from class i assigned to class j by the clustering. Because the actual numerical value of the cluster IDs is irrelevant and typically out of alignment, we need only ensure that this confusion matrix resemble a permutation of a diagonal matrix. If we reorganize the columns so that the largest counts are on the diagonal, then the total error is defined as the sum of all the off-diagonal elements.

3. The total error can also be broken down into the sum of the error due to quantization and the error due to what we will call the *raw error* (RE): $TE = QE + RE$. Here, the raw error summarizes the additional error (as a count of incorrect assignments) due to clustering alone. This error captures how well the clustering algorithm has performed, regardless of quantization effects, sequence knowledge, or the current application in general.

4. After clustering and after reassignment post-processing, we arrive at the *final error* (FE). This error represents the final tally of errors assuming sequence post-processing, and summarizes the number of original data points which have been assigned to incorrect subject classes by again summing the off-diagonal entries of a confusion matrix that is generated after reassignment has been performed.

At times, the final error will be greater than the raw error, indicating that sequence-based reassignment after clustering was unable to erase the initial quantization error, or even incurred additional error. However, for this application domain, the final error is always less than the total error $(FE < QR + RE)$, demonstrating that the sequence information utilized during post-clustering reassignment is still beneficial, even if it cannot improve on the raw clustering performance RE.

4.3.4 Multi-Source Spectral Clustering

We now apply the spectral co-clustering algorithm (Algorithm 3) described in section 2.3 to both the 5 and 10 user datasets. For the generic single-source clustering step of the algorithm, we perform K-means clustering 5 times with random initial conditions. The final clustering is returned as the best of the 5 replicates. The resulting performance, averaged over 100 trials, is shown in Table 4.2 for three different configurations of feature co-occurrences: the face channel vs. upper body histograms, faces vs. lower body histograms, and faces vs. height measurements.

Figure 4-4 gives a more detailed picture of performance over the trials, where we show boxplots for both the raw clustering error and the final error resulting from all three feature relations. In the case of faces vs. lower-body histograms from the 10 subject dataset, the solutions typically took on only one out of a handful of possible forms, giving many repeated error counts and a multi-modal distribution of errors. Though the boxplots do not capture this kind of behavior accurately, they do tell us more than the sample mean and variance, and we have opted to show them in these circumstances anyhow. In the case of faces vs. upper-body histograms applied to the 5 subject dataset, the variance was zero (every trial incurred the same number of mistakes), and the "boxes" have become single lines at the respective error counts.

These results collectively say that the nonlinear transformation spectral co-clustering applies to the space of input points is better suited to some co-occurrence matrices than others. Faces vs. upper histograms data gives the best results after sequence information is used for both 10 and 5 subject scenarios, whereas other co-occurrence configurations do not become sufficiently separated in the embedding to give as impressive results. We might further conclude that the raw error is also lowest when using faces vs. upper-body histogram features, since the advantage of lower- over upper-body histograms in the 10 subject case is statistically insignificant.

Interleaved co-clustering in the spirit of Algorithm 1 was also evaluated using the multi-source spectral algorithm as the generic 2-way co-clustering scheme. Unfortunately, spectral clustering is not easily biased since the initial conditions do not take on the form of partitioning functions. Thus, to facilitate interleaving we attempted to influence the behavior of successive applications of the algorithm via the initial cluster coordinates required during the single-source clustering step. Given a mapping function resulting from a previous clustering and using K-means as the single-source

Dataset	Input Matrix	Avg. Raw Error (RE)	Avg. Final Error (FE)
5 subject	face/upper	**8.5**% (627) $\sigma = 0$	**1.2**% (88) $\sigma = 0$
5 subject	face/lower	16.9% (1246) $\sigma = 274$	24.9% (1839) $\sigma = 411$
5 subject	face/height	18.2% (1346) $\sigma = 309$	21.8% (1610) $\sigma = 723$
10 subject	face/upper	7.4% (1412) $\sigma = 1034$	**8.4**% (1602) $\sigma = 1163$
10 subject	face/lower	**7.2**% (1387) $\sigma = 599$	9.2% (1756) $\sigma = 617$
10 subject	face/height	13.8% (2643) $\sigma = 556$	15.0% (2877) $\sigma = 603$

Table 4.2: Spectral clustering applied to three 2-way joint distributions involving face information, for both 10 and 5 subject datasets.

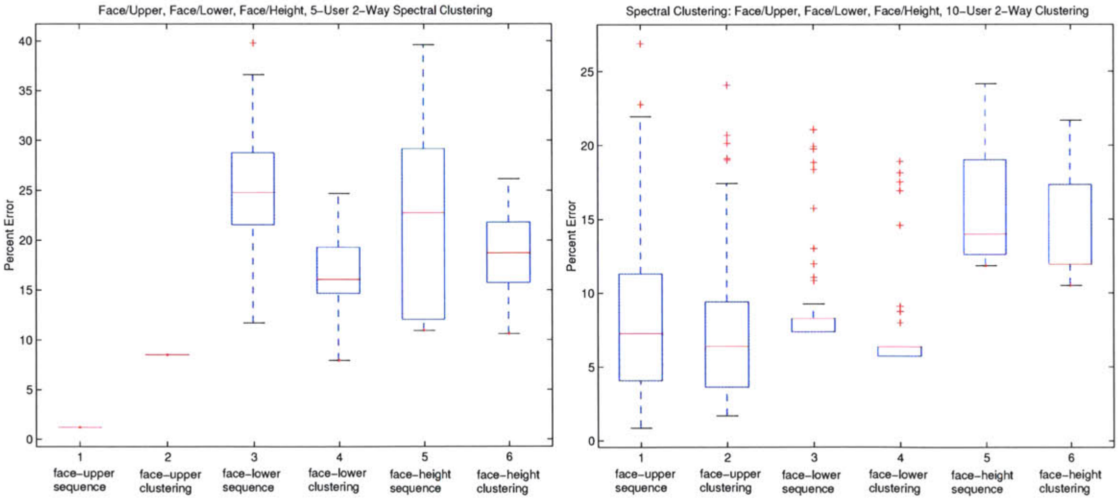

Figure 4-4: Two-way multi-source spectral clustering. (Left) Performance on the 5 subject dataset, and (Right) performance on the 10 subject dataset. Along the horizontal axis, "sequence" refers to the final error, computed using sequence knowledge, while "clustering" refers to the raw clustering error alone.

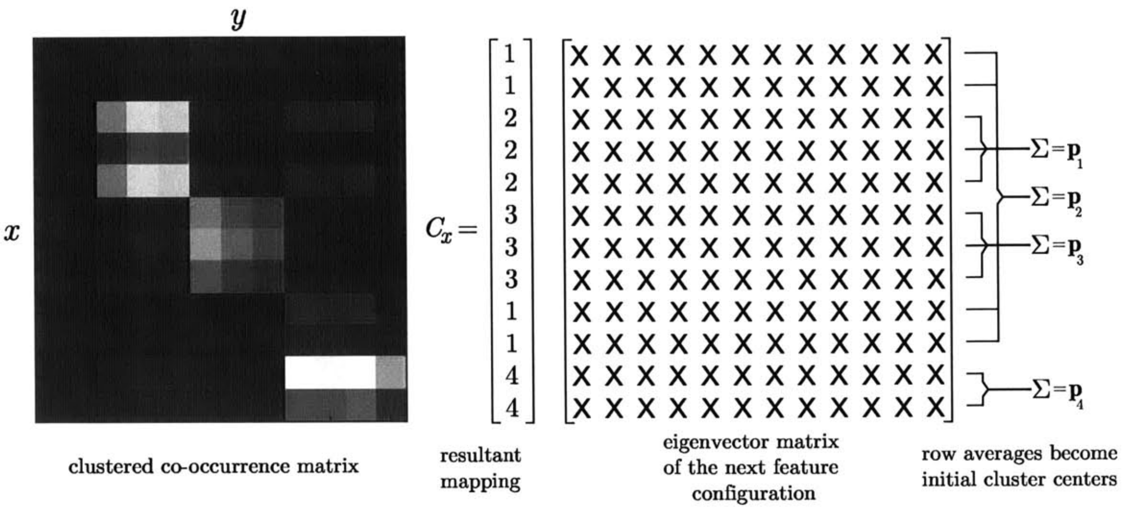

Figure 4-5: Illustration of the spectral clustering seeding process. A mapping C_x from a previous clustering process selects rows from a matrix of stacked eigenvectors corresponding to another contingency configuration. The averaged rows become initial cluster coordinates during the K-means clustering step of the multi-source spectral clustering algorithm.

clustering algorithm, we set the initial cluster coordinates for K-means to the average of the rows of the eigenvector matrix assigned to the same cluster according to the previous mapping. The idea is illustrated in Figure 4-5, where the clustered left-hand co-occurrence matrix generates the row mapping C_x (center). The mapping is shown selecting rows of a hypothetical eigenvector matrix from a subsequent clustering trial, on the basis of cluster membership. The rows are then averaged to give the initial K-means seeds $\mathbf{p}_1, ..., \mathbf{p}_4$. For the interleaving experiments we conducted, each iteration of the interleaving algorithm sequentially clusters all three of the co-occurrence configurations mentioned in Table 4.2, while passing the mapping from rows (faces) to clusters as initial conditions to subsequent clusterings. The algorithm was run for 15 iterations, and 100 trial runs of the algorithm were recorded.

Dataset	Input Matrix	Avg. Raw Error (RE)	Avg. Final Error (FE)
5 subject	face vs. up./lo./ht.	8.5% (627) $\sigma = 0$	1.2% (88) $\sigma = 0$
10 subject	face vs. up./lo./ht.	1.7% (323) $\sigma = 0$	0.9% (165) $\sigma = 0$

Table 4.3: Interleaved multi-source spectral clustering applied to the 5 and 10 subject datasets. The marginals faces vs. heights, faces vs. lower histograms, and faces vs. upper histograms were clustered sequentially at each iteration.

The results of the technique are shown in Table 4.3. For the 5 subject dataset, each of 100 trials gave performance equal to the best average 2-way co-occurrence error shown in the bottom half of Table 4.2: that of faces vs. upper body histograms. Thus, while interleaved clustering of the different co-occurrence matrices did not improve on the best we could do with any single co-occurrence matrix, it did not allow weaker configurations to corrupt the strongest one. When applied to the 10 subject data, interleaving helped steer the solution towards the best mapping observed among *all* the trials conducted for the top half of Table 4.2. Each of the 100 trials gave the same answer: 323 raw clustering mistakes, and 165 total mistakes. This particular solution was found during only a handful of the trials where we applied spectral clustering without interleaving to faces vs. upper-body co-occurrences, as can be verified from Figure 4-4. In this case, interleaving thus served to guide our solution towards the best possible solution over the various feature combinations, and eliminated much of the variation seen in the figures reported in the top section of Table 4.2.

4.3.5 Weight-Based Clustering

We next applied the weight-based algorithm (Algorithm 4) described in section 3.1 to three different datasets. In addition to the 10 and 5 person datasets defined above, we also tested the algorithm on a simpler 3 subject dataset described by 3610 points and 13 sequences. For the datasets examined in this chapter, the weight-based algorithm generates results that are typically more sensitive to the desired number of clusters or the size of the contingency matrix than the other methods. In general, it behooves the experimenter to have fewer sequences with more points, rather than many small sequences, and this is particularly true for weight-based co-clustering. The smaller 3 subject dataset will therefore serve to better illustrate how the present algorithm scales with problem complexity.

In each of the experiments, we included *only* the constraint stipulating that each "point" (row or column) can be assigned to at most one cluster. The weight-based scheme was tested with additional constraints, but typically showed poor convergence and gave inferior results, even for the 3 category problem. We believe that the two

Dataset	Input Matrix	Avg. Raw Error (RE)	Avg. Final Error (FE)
3 subject	face/upper	22.6% (815) $\sigma = 218$	30.0% (1080) $\sigma = 389$
3 subject	face/lower	23.0% (830) $\sigma = 222$	29.9% (1076) $\sigma = 380$
3 subject	face/height	**21.7%** (782) $\sigma = 237$	**27.6%** (997) $\sigma = 400$
5 subject	face/upper	**28.8%** (2067) $\sigma = 287$	34.7% (2559) $\sigma = 576$
5 subject	face/lower	29.5% (2117) $\sigma = 332$	35.5% (2616) $\sigma = 631$
5 subject	face/height	28.8% (2072) $\sigma = 283$	**33.9%** (2503) $\sigma = 587$
10 subject	face/upper	**30.2%** (5772) $\sigma = 975$	**35.0%** (6681) $\sigma = 1175$
10 subject	face/lower	31.1% (5954) $\sigma = 948$	36.1% (6899) $\sigma = 1189$
10 subject	face/height	32.0% (6124) $\sigma = 1079$	37.4 % (7164) $\sigma = 1352$

Table 4.4: Weight-based clustering applied to all 2-way co-occurrence configurations, with 3, 5, and 10 subject datasets. (First section): Performance on the 10 subject dataset, (Middle section) Performance on the 5 subject dataset, and (Bottom section) Performance on the 3 subject dataset.

clustering-specific constraints (3.3) and (3.5) either cause conflicts during the (non-convex) optimization, or impose too many additional variables over which to optimize. If we remove the constraints which enforce the condition that every point is assigned to a cluster (3.5), the algorithm does significantly better and usually assigns a great majority of the points to a cluster anyhow since the main objective (3.2) weakly enforces this condition. Whenever a row or column was left unassigned (e.g. all weights for the point were identically zero), we simply assigned it to a randomly selected cluster. The constraint enforcing the rule that each cluster should get at least one point was largely unnecessary here, since the unique assignment constraint (3.3) effectively forces at least one point into all clusters. This behavior was empirically observed in all trials, and third constraint (3.7) was left out entirely. Finally, we did not use an adaptive rate parameter or conduct line searches for the optimal step size, but instead simply chose a single crude value ($\eta_0 = 10$ for the 10 and 5 subject datasets and $\eta_0 = 20$ in the 3 subject case) that was applied to all gradient update rules in Algorithm 4. Updating was performed in an online fashion for 1800 epochs, and the distribution $P(\hat{x}, \hat{y})$ was recomputed after each weight adjustment.

The resulting performance averaged over 100 trials for each of the datasets is shown in Table 4.4. In each case, clustering was performed on the same three feature co-occurrence configurations as before: faces vs. upper body histograms, faces vs.

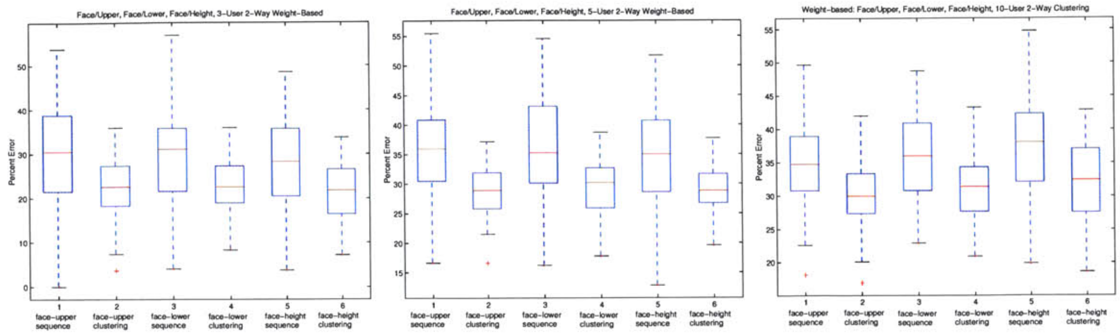

Figure 4-6: (Left) Weight-based performance on a 3 subject dataset with 3610 points, and 13 sequences. Middle: Weight-based performance on the 5 subject dataset. (Right) Performance on the 10 subject dataset. Along the horizontal axis, "sequence" refers to the final error, computed using sequence knowledge, while "clustering" refers to the raw clustering error alone.

lower body histograms, and faces vs. height measurements. In Figure 4-6 we show boxplots for each configuration and each dataset, summarizing performance in terms of raw error and final error tallies, marked "clustering" and "sequence" respectively along the horizontal axes. Unfortunately, for these datasets weight-based clustering is not as competitive as the other methods investigated in this chapter. However, the trend among errors over the different datasets suggests that the weight-based approach is viable for small problems with fewer sequences that have been balanced in size, and for such clustering tasks we have found that the approach is competitive with Dhillon's algorithm. In addition, the algorithm is a true co-clustering technique in that it optimizes and returns both row and column mapping functions. For applications where all mappings are desired, weight-based clustering might be an attractive solution.

Interestingly, the results summarized in Table 4.4 show that for smaller problems the height channel was most informative. In light of the feature preferences of the spectral algorithm above, this suggests that different algorithms might exploit co-occurrence information in distinct ways. Despite the fact that the weight-based scheme does not offer particularly stellar results, weight-based clustering might prove helpful when used in concert with other algorithms that exploit different structures in the data. The large standard deviations in the final error are not entirely surprising,

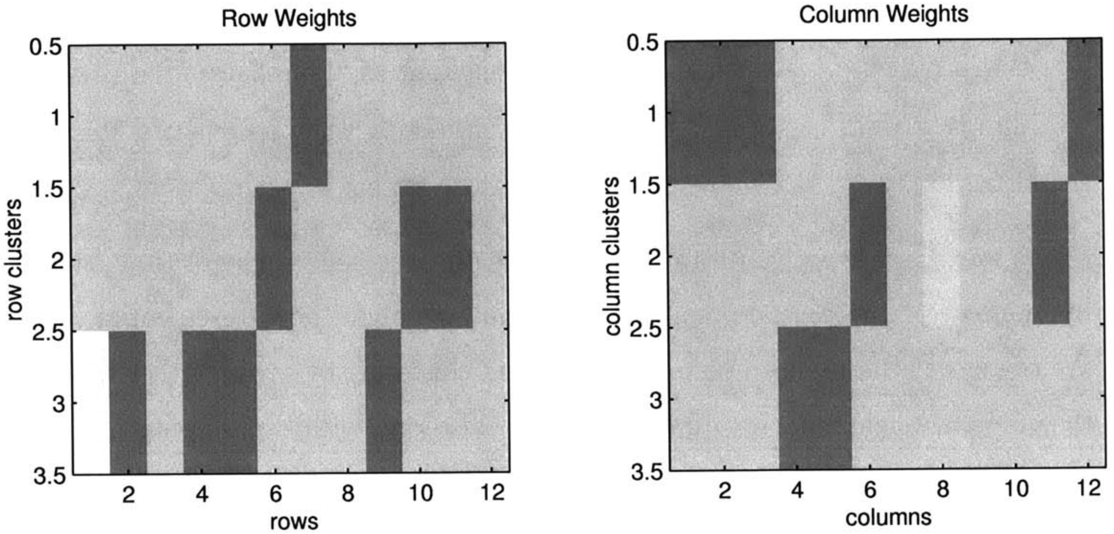

Figure 4-7: (Left) Typical converged row weights for the 3 subject case, where cluster indices run along the vertical axis and joint distribution row indices run along the horizontal axis. Dark cells denote weights near 1, while bright cells indicate weights near 0. Where a given column has no dark areas, the corresponding row was not assigned to any cluster by the algorithm. (Right) Typical converged column weights.

given that we reassign possibly large groups of points during the sequence-based post-processing, and variance in the raw errors were, on average, smaller with weight-based clustering in comparison to the spectral method.

For each trial we initialized the weights and constraint Lagrange multipliers to random values in the interval $[0, 1]$. Figure 4-7 shows sample converged row (left) and column (right) weights after a typical run of the weight-based algorithm on the 3-class problem, where dark patches correspond to values near unity. Each row in the plots correspond to a cluster, while the columns are the original row/column co-occurrence matrix indices. Brighter areas denote weights near zero, and indicate that the corresponding rows or columns are not members of the cluster. The mappings are thus recovered from the weights by recording the index of the largest weight for that row or column:

$$C_{row}(v) = \arg\max_{\hat{u}} w^{r}_{\hat{u},v}, \; v = 1, \ldots, m'$$

$$C_{col}(v) = \arg\max_{\hat{u}} w^{c}_{\hat{u},v}, \; v = 1, \ldots, n'.$$

From the weight patterns, it can also be seen that the unique assignment constraint was successfully enforced, as was the third constraint (3.7) implicitly. Since we chose to omit the second clustering condition, not all points were assigned to clusters by the algorithm.

As was the case with spectral co-clustering, the weight-based algorithm is not easily interleaved since initial mappings tell us nothing about how to set initial weights. We therefore attempted to perform interleaved clustering by saving either the row or the column weights $\mathbf{w}^r, \mathbf{w}^c$ from a previous clustering with overlapping modalities, and passing those weights as an initial starting point for a subsequent trial. While this technique ought to bias the new weights towards a portion of the solution space derived from previous results, in practice interleaving did not cause successive clustering runs to escape inherited local minima. Thus, once a set of weights had converged to a local-optimum, later applications of the algorithm did not significantly improve the objective function. Interleaving with weight-based clustering was thus unsuccessful in improving performance.

4.3.6 Annealing Methods

Turning to the stochastic simulated annealing algorithm (Algorithm 2) described in section 2.2.1, we attempted to label faces given both 2-way and 3-way joint distributions generated from the 5 and 10 user datasets. For the 2-way trials, we tested the algorithm on the same set of feature co-occurrence configurations as before: faces vs. upper-body histograms, faces vs. lower-body histograms, and faces vs. heights. The two 3-way contingency tensors were generated by looking at faces vs. upper vs. lower histograms, and faces vs. upper histograms vs. heights. In all 2-way cases we applied multi-source annealing to the information-theoretic objective function (1.2) discussed in Chapter 1. The three-source optimization objective takes on a similar form: we simply extended the relevant distributions to three variables, and optimized

Dataset	Input Matrix	Avg. Raw Error (RE)	Avg. Final Error (FE)
5 subject	face/upper	**6.3%** (463) $\sigma = 60$	**8.2%** (603) $\sigma = 206$
5 subject	face/lower	8.5% (621) $\sigma = 170$	14.7% (1081) $\sigma = 230$
5 subject	face/height	7.8% (572) $\sigma = 200$	10.6% (781) $\sigma = 269$
5 subject	face/upper/lower	**6.7%** (494) $\sigma = 130$	10.1% (742) $\sigma = 286$
5 subject	face/upper/height	7.1% (523) $\sigma = 110$	**9.6%** (708) $\sigma = 284$
10 subject	face/upper	**1.2%** (221) $\sigma = 335$	**1.6%** (313) $\sigma = 361$
10 subject	face/lower	4.1% (781) $\sigma = 219$	6.6% (1264) $\sigma = 406$
10 subject	face/height	12.3% (2347) $\sigma = 434$	13.3% (2552) $\sigma = 381$
10 subject	face/upper/lower	5.2% (989) $\sigma = 319$	6.2% (1186) $\sigma = 518$
10 subject	face/upper/height	**4.5%** (861) $\sigma = 261$	**5.7%** (1087) $\sigma = 484$

Table 4.5: Multi-source simulated annealing applied to three 2-way joint distributions and two 3-way configurations. (Top section) 2-way trials using the 5 subject dataset, (2nd section) 3-way trials using the 5 subject dataset, (3rd section) 2-way trials using the 10 person dataset, (Last section) 3-way trials using the 10 person dataset.

over the mapping functions C_x, C_y, and C_z:

$$\begin{aligned} J(C_x, C_y, C_z) &= I(X;Y;Z) - I(\hat{X};\hat{Y};\hat{Z}) \\ &= \sum_{\hat{x},\hat{y},\hat{z}} \sum_{\substack{x\in\hat{x},y\in\hat{y},\\ z\in\hat{z}}} P(x,y,z) \log \frac{P(x,y,z)}{P(\hat{x},\hat{y},\hat{z})\frac{P(x)}{P(\hat{x})}\frac{P(y)}{P(\hat{y})}\frac{P(z)}{P(\hat{z})}} \\ &= D_{KL}(P(x,y,z)||Q(x,y,z)), \end{aligned}$$

For each trial, the mappings functions were randomly initialized to integers in the range 1, ..., $N_{subjects}$. The temperature was initialized to $T_0 = 5$, and was reduced after completion of each outer iteration according to the update rule $T_n \leftarrow 0.98 \cdot T_{n-1}$. The number of inner iterations at step 3 of the algorithm was chosen such that each entry of the mapping functions would be touched approximately 5 times on average. The algorithm was stopped when the temperature dropped below $T_{stop} = 0.005$.

The resulting performance over 55 trials for each of the clustering problems is shown in Table 4.5, while Figures 4-8 and 4-9 show boxplots summarizing the raw and final error counts for all 2- and 3-way configurations respectively. As was the case with the spectral clustering experiments, for some configurations the multi-source annealing algorithm gave error counts which mainly took on only a handful of values, and disregarding outliers, gave an effective variance smaller than the values reported

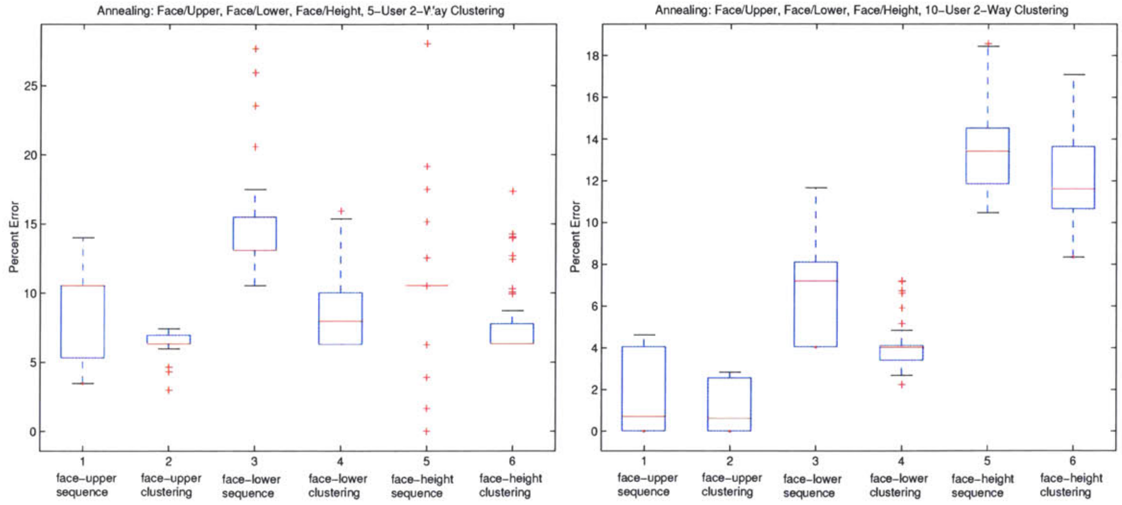

Figure 4-8: Multi-source simulated annealing applied to three 2-way co-occurrence matrices. (Left) Performance on the 5 subject dataset, and (Right) performance on the 10 subject dataset. Along the horizontal axis, "sequence" refers to the final error, computed using sequence knowledge, while "clustering" refers to the raw clustering error alone.

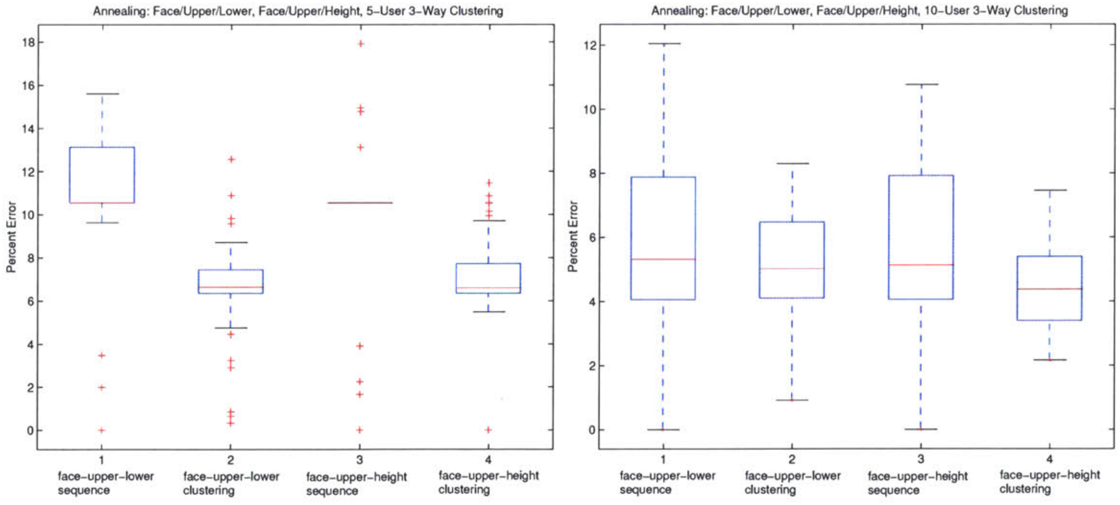

Figure 4-9: Multi-source simulated annealing applied to two 3-way joint distributions. (Left) Performance on the 5 subject dataset, and (Right) performance on the 10 subject dataset. Along the horizontal axis, "sequence" refers to the final error, computed using sequence knowledge, while "clustering" refers to the raw clustering error alone.

in Table 4.5 (which includes outliers). Considering the fact that annealing is in part designed to reduce variability of the solution, this behavior is desirable. Unfortunately, it makes for poor looking box-plots. Nevertheless, we feel that these "degenerate-looking" box-plots still offer useful information, in the form of outlier counts and range statistics, and we include them anyhow. Looking at these results, we can see that annealing gives the best clusterings when applied to faces vs. upper body histogram data, but does not sufficiently exploit third order interactions to give lower error rates in the 3-way case *on average*. However, over the trials we conducted, there was at least one case where performance on the 3-way co-occurrence tables gave the minimum error, while for the 2-way examples this minimum was not always reached. This implies that we are likely to find the best possible clustering by conducting several 3-way trials and taking the one for which the objective is smallest as the final mapping. For the 2-way configurations however, the results suggest that we cannot always be sure we are finding the best clustering: faces vs. upper-body histograms is clearly the strongest combination of features, but in the absence of such knowledge we would not know to prefer this particular set of co-occurrences over the others. Clustering all of the modalities together while paying no attention to which are "strong" and which are "weak" can possibly find the best mapping automatically, but at the expense of increased computation time.

Dataset	Input Matrix	Avg. Raw Error (RE)	Avg. Final Error (FE)
5 subject	face vs. up./lo./ht.	7.0% (513) $\sigma = 60$	7.8% (574) $\sigma = 200$
10 subject	face vs. up./lo./ht.	1.1% (216) $\sigma = 216$	1.3% (247) $\sigma = 295$

Table 4.6: Interleaving of the multi-source annealing algorithm applied to the 5 and 10 subject datasets. The marginals faces vs. upper histograms, faces vs. lower histograms, and faces vs. heights were clustered sequentially at each iteration.

Interleaving of the 2-way co-occurrence configurations can give improved results when using the annealing algorithm as well. Faces vs. heights, faces vs. lower histograms, and faces vs. upper histograms were sequentially clustered (in that order) 5 times for each interleaving trial, while directly passing previous mappings as initial conditions. Table 4.6 shows the average performance over 50 such trials. For the 10 subject dataset, interleaving gave groupings which reduced the best 2-way final error count by about 0.3%, or 66 errors, and gave a lower variance. The raw error performance remained about the same. For the 5 subject dataset, interleaving also reduced the final error count, resulting in 29 fewer errors, or a 0.4% reduction. While these margins may not be large, the experiments do suggest that interleaving with annealing can give results equal to or better than any single 2-way result.

4.3.7 Dhillon's Algorithm

In order to help evaluate the performance of the techniques proposed in this thesis, the information-theoretic co-clustering algorithm due to Dhillon et.al. [11] discussed in section 1.5.3 was also applied to the 5 and 10 subject datasets. The algorithm was started with random initial mappings, and allowed to run until the objective improved by less than 10^{-10} bits. The results from 100 trials are summarized in Table 4.7 and Figure 4-10. For the 5-subject clustering task, Dhillon's algorithm performed equally well (a statistical tie) given faces vs. upper-body histograms and given faces vs. heights. In the 10-subject case, faces vs. upper-body histograms proved to be the most informative co-occurrence configuration as well.

Following Algorithm 1, we also interleaved clustering of the 2-way co-occurrence configurations. In this case, the initial mappings passed to Dhillon's algorithm corresponded to the results from previous applications of the algorithm. Each trial iterated over all three feature configurations 15 times, and we again terminated each sub-clustering process when the objective improved by less than 10^{-10} bits. The average performance over 100 trials is shown in Table 4.8. Comparing these error counts to those obtained from the individual 2-way matrices in Table 4.7, we can see that, for this problem, interleaving all available 2-way marginals yields a substantial improvement. For the 5 person dataset, the final error drops by approximately 3 percentage points, while for the 10 subject data the final error is reduced by about 2 percentage points. The variance of the error increased slightly for the 10 subject dataset, but was roughly the same for 5 people.

Dataset	Input Matrix	Avg. Raw Error (RE)	Avg. Final Error (FE)
5 subject	face/upper	**15.0%** (1075) $\sigma = 413$	20.3% (1460) $\sigma = 650$
5 subject	face/lower	18.7% (1346) $\sigma = 448$	27.9% (2003) $\sigma = 697$
5 subject	face/height	15.1% (1084) $\sigma = 368$	**19.2%** (1379) $\sigma = 638$
10 subject	face/upper	**13.3%** (2538) $\sigma = 966$	**17.0%** (3249) $\sigma = 1089$
10 subject	face/lower	14.2% (2708) $\sigma = 1093$	18.9% (3610) $\sigma = 1210$
10 subject	face/height	18.8% (3588) $\sigma = 941$	22.2% (4247) $\sigma = 1075$

Table 4.7: Performance of Dhillon's algorithm applied to the 5 subject (Top section) and 10 subject (Bottom section) datasets.

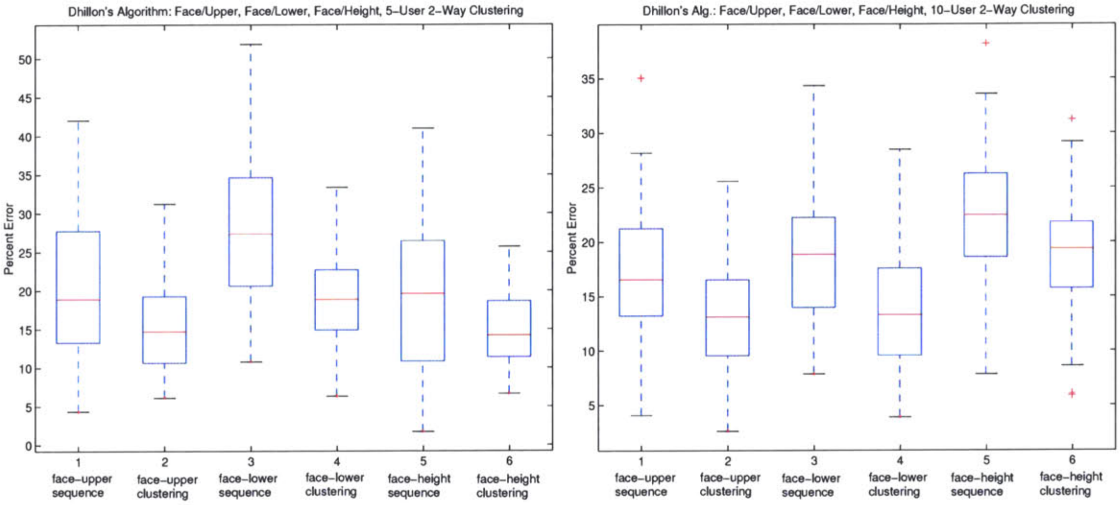

Figure 4-10: Boxplots showing performance of Dhillon's algorithm on the 5 subject (Left) and 10 subject (Right) datasets. Along the horizontal axis, "sequence" refers to the final error, computed using sequence knowledge, while "clustering" refers to the raw clustering error alone.

Dataset	Input Matrix	Avg. Raw Error (RE)	Avg. Final Error (FE)
5 subject	face vs. up./lo./ht.	12.2% (893) $\sigma = 405$	16.3% (1194) $\sigma = 660$
10 subject	face vs. up./lo./ht.	12.7% (2433) $\sigma = 1259$	15.3% (2928) $\sigma = 1370$

Table 4.8: Interleaved Dhillon's algorithm applied to the 5 and 10 subject datasets. The marginals faces vs. heights, faces vs. lower histograms, and faces vs. upper histograms were clustered sequentially at each iteration.

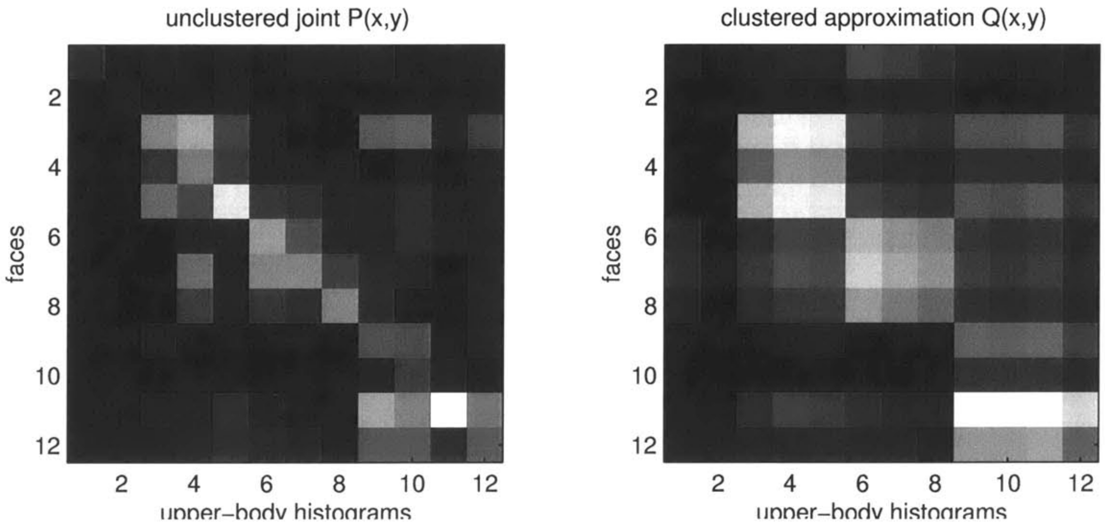

Figure 4-11: (Left) An example 12x12 unclustered joint probability distribution $P(x, y)$ over faces and upper-body histograms. Bright areas indicate larger entries. (Right) A typical approximation $Q(x, y)$ after clustering has been performed.

4.4 Discussion

As a brief visual example of the clustering occurring above, in Figure 4-11 we show the original distribution $P(x, y)$ of faces vs. upper histograms and a typical compressed approximation $Q(x, y)$ resulting from a clustering, where larger probability values are indicated by brighter areas. Note that we observe in $Q(x, y)$ the block-like structure discussed in Chapter 1. On the left, this figure shows both the form of most of the input matrices used in the experiments above, and on the right, also illustrates pictorially the sort of clustering results we would like to obtain in the form of the approximation matrix $Q(x, y)$. We also take this opportunity to note that the quality of the clustering resulting from any optimization process based on the information theoretic-objective is immediately available during the optimization itself. If, for a given trial the loss in mutual information is large, then it might make sense to adapt algorithm parameters or restart from different initial conditions.

In Tables 4.9 and 4.10 we compare the algorithms and feature sets respectively. The former table gives the best feature configuration and error percentage for each algorithm, along with approximate running times for each of the methods, while the latter table shows the best algorithm and the best error percentage for each co-

Dataset	Algorithm	Run Time	Best 2-way Features	Best Final Error
5 subject	**spectral**	seconds	face vs. upper	**1.2%** $\sigma = 0$
5 subject	annealing	hours	face vs. upper	8.2% $\sigma = 206$
5 subject	Dhillon	seconds	face vs. height	19.2% $\sigma = 638$
5 subject	wt-based	minutes	face vs. height	33.9% $\sigma = 587$
10 subject	**annealing**	hours	face vs. upper	**1.6%** $\sigma = 361$
10 subject	spectral	seconds	face vs. upper	8.4% $\sigma = 1163$
10 subject	Dhillon	seconds	face vs. upper	17.0% $\sigma = 1089$
10 subject	wt-based	minutes	face vs. upper	35.0% $\sigma = 1175$

Table 4.9: For each dataset and over all feature sets, we show the best performance for each algorithm. The fourth column corresponds to the particular 2-way matrix which gave the best error listed in the final column. Approximate running times for the algorithms are given for reference in column three.

Dataset	2-way Feature Set	Best Avg. Final Error (FE)	Algorithm
5 subject	face vs. upper	1.2% $\sigma = 0$	spectral
5 subject	face vs. lower	14.7% $\sigma = 230$	annealing
5 subject	face vs. height	10.6% $\sigma = 269$	annealing
10 subject	face vs. upper	1.6% $\sigma = 361$	annealing
10 subject	face vs. lower	6.6% $\sigma = 406$	annealing
10 subject	face vs. height	13.3% $\sigma = 381$	annealing

Table 4.10: For each data set and over all algorithms, we show the best performance for each feature set. The last column shows the particular algorithm which gave the best error listed in column three.

Dataset	Algorithm	Avg. Raw Error Std. Dev. (%)
5 subject	**annealing**	$\langle\sigma\rangle = \mathbf{2.0\%}$
5 subject	spectral	$\langle\sigma\rangle = 2.7\%$
5 subject	wt-based	$\langle\sigma\rangle = 4.1\%$
5 subject	Dhillon	$\langle\sigma\rangle = 5.6\%$
10 subject	**annealing**	$\langle\sigma\rangle = \mathbf{1.7\%}$
10 subject	spectral	$\langle\sigma\rangle = 3.8\%$
10 subject	wt-based	$\langle\sigma\rangle = 5.2\%$
10 subject	Dhillon	$\langle\sigma\rangle = 5.2\%$

Table 4.11: Variability in the results from each algorithm. The final column lists the standard deviation of the raw error as a percentage of the total number of points in the datasets, averaged over all 2-way feature configurations for each respective algorithm.

occurrence configuration. Out of the algorithms tested above, the data presented in these tables indicate that multi-source spectral clustering applied to faces vs. upper-body histograms gave the best performance on the 5 subject dataset, while annealing applied to the same set of features gave the best results for the 10 person dataset. Overall, the multi-source annealing algorithm gives the best results on both datasets, followed by the spectral algorithm. Annealing and spectral clustering also give superior results for all feature configurations compared to seeded agglomerative clustering for the 10 subject dataset, while for the 5 person problem, spectral clustering gave better mappings only for faces vs. upper-body histograms. The annealing algorithm was always better than seeded agglomerative clustering for the 5 subject dataset. Weight-based clustering and Dhillon's method were both always worse than the agglomerative algorithm, for all datasets. It must be remembered however, that comparing co-clustering algorithms to any single-source clustering scheme is a bit like comparing "apples to oranges": the seeded agglomerative algorithm, for instance, does not provide multiple mappings, and exploits sequence knowledge to a greater extent than the multi-source algorithms.

We can also conclude that faces vs. upper-body histograms was the most informative co-occurrence relation for all algorithms in the case of 10 subjects. For the 5 subject dataset, faces vs. heights proved to be the most useful for Dhillon's algorithm and the weight-based approach. The other two methods preferred faces vs. upper-body histograms again. Faces vs. lower-body histograms gave the worst results for all algorithms and all datasets, but still gave results far lower than random assignment: the largest difference in error between these features and the best feature combination was only about 11 percentage points in the worst case. It is not surprising however that lower-body histograms were relatively uninformative, as most of the subjects wore jeans of one shade or another. The results therefore collectively suggest that faces, upper-body histograms, and heights are all good features to include in a recognition or surveillance system.

It is also helpful to look at the variability of the results shown in the preceding section. That variance figures for the raw error are typically large is unsurprising;

the sequence-based post-processing step reassigns great numbers of points, and can easily change the final error count by hundreds of points. If a clustering puts just a bare majority of points from a sequence into a cluster, then all points go into that cluster. Likewise, if a slight minority are placed into a cluster, then all points are removed. The standard deviations corresponding to the raw clustering error thus give a far better measure of the variability among results for a given algorithm, and in Table 4.11 we give standard deviations as a percentage of the total number of points in the datasets for each algorithm, averaged over the raw error results for all 2-way feature configurations.

These variance data show that the annealing algorithm gives the most "stable" results, as would be expected given the nature of the technique. While the weight-based algorithm gives the worst clusterings, the variance of the error incurred by the mappings is comparatively small. Conversely, Dhillon's algorithm performed somewhere in the middle of the pack error-wise, but shows the largest variability.

On average, the annealing algorithm was not able to take significant advantage of three-way interactions when clustering the two 3-way joint distributions. It is possible that when all modalities are clustered jointly, weaker datasets corrupt stronger ones. In this case, it could be that lower body histograms tend to reduce discriminability in other channels when clustered together. While the average performance of 3-way clustering was below the best 2-way average, clustering of the 3-way co-occurrences did at times give the lowest observed error count. This indicates that we can ensure a good clustering by repeating the algorithm several times, and taking the mapping giving the smallest corresponding objective value. In addition, clustering multiple sources at once does not require prior knowledge of strong and weak modalities. Thus, 3-way clustering is one possible way to obtain good results given ample computational resources and very little prior information.

Interleaving, however, seemed to promote the exchange of 2-way information while preventing significant cross-corruption. In most cases where we applied interleaving, some benefit was realized–up to several percentage points at times. However, it should be noted that when there are substantially weaker modalities combined with strong ones, the effectiveness of interleaving can be dependent on the order in which the marginals are clustered. If the weakest 2-way configuration is clustered last, some corruption is possible. On the other hand, if the strongest 2-way co-occurrence table is the final matrix to be clustered, the results can be better than of any individual 2-way configurations alone. In this sense, weaker channels are more likely to help stronger ones, but stronger channels are less likely to help weaker ones.

Overall, for this particular problem, the simulated annealing algorithm appears to be the most powerful of the methods. The technique combines high raw clustering accuracy and low variance, to give generally good results for all of the clustering problems. The benefits of annealing however must be weighed against the computational requirements. For all datasets, the algorithm took longer than any other method. Considering running time then, spectral clustering and Dhillon's algorithm are competitive options, as they are both computationally fast, and offer similar performance. In summary, the experiments conducted in this chapter collectively suggest that a reliable recognition system with autonomous training and model building capabilities is a very real possibility.

4.5 A Final Noise Tolerance Experiment

In practice, data collected automatically by surveillance and recognition systems are often substantially noisy. It is not unreasonable to expect that on occasion multiple people might appear in front of the camera simultaneously, contaminating a sequence of recordings with ambiguous information. In addition, face detection algorithms will always have some probability of error, and it is likely that a few random spurious patches in an image will be extracted as faces. There are many possible sources of "noise" that the experimenter can encounter, and in most real-world situations we can expect that all modalities will sustain some amount of corruption. A system's robustness to noise is thus an important quality to evaluate if we are to have some idea of how it will perform in practice. In order gain some insight into the extent to which multi-source clustering algorithms can handle noise, we conducted

Dataset	Algorithm (linkage)	Avg. Final Error
10 subject corrupted	spectral	14.3% (2735) $\sigma = 1449$
10 subject corrupted	seeded agglom. (single)	58.0% (11076)
10 subject corrupted	seeded agglom. (average)	30.7% (5868)
10 subject corrupted	seeded agglom. (complete)	57.5% (10996)
10 subject original	spectral	8.4% (1602) $\sigma = 1163$
10 subject original	seeded agglom. (single)	22.8% (4359)
10 subject original	seeded agglom. (average)	30.7% (5868)
10 subject original	seeded agglom. (complete)	57.0% (10900

Table 4.12: Multi-source and single-source clustering performance with a noisy dataset (top section), and on the original 10 subject dataset (bottom section).

a brief experiment in which the spectral co-clustering algorithm was applied to to a deliberately corrupted dataset. We believe that robustness to noise depends more on the fact that we are using contingency data than on the particular algorithm used to cluster that data. Multi-source spectral clustering was therefore chosen to represent co-clustering in general as a good tradeoff between speed and accuracy.

For this experiment, we started with the same 10 subject dataset described above, but then randomly corrupted data entries in all modalities and all sequences to generate a "noisy" dataset. The noise data points were taken from actual recordings of other subjects not among the original 10. Some are "correct" in the sense that they accurately capture what they are supposed to (e.g. faces are faces), while others are incorrect spurious measurements. Examples of such incorrect measurements include images of the back of a person's head as a "face", or patches of rug as upper-body histograms. The noise points were randomly distributed among sequences, and replaced randomly selected (original) entries. In all, we corrupted 1094 points, corresponding to a noise level of 5.7%, in addition to any noise already present in the dataset. In Table 4.12 we show the results of 100 spectral clustering trials applied to faces vs. upper-body histograms co-occurrence data generated from the corrupted dataset. The table also shows the performance obtained on the original 10 subject dataset, and the performance of the single-source seeded agglomerative algorithm, which is not contingency based, on both corrupted and original data. As before, agglomerative clustering was applied to 30 PCA components per face.

As would be expected, both algorithms showed an increase in final error when applied to the corrupted dataset. In the case of seeded agglomerative clustering, however, the optimal linkage choice has changed, while the increase in error for all linkages excepting the average linkage was far more substantial than the increase in final error due to the spectral algorithm. These results collectively suggest that contingency clustering is more robust to noise and other corrupting factors than traditional single-source clustering techniques.

Chapter 5

Conclusion

This thesis began by asking the question, "How can we best cluster multiple connected datasets?". After arguing that classical clustering techniques were inappropriate, we proceeded to formally define the multi-source clustering problem, and showed that a good clustering minimizes the loss in mutual information among data sources between a contingency (co-occurrence) matrix and a clustered approximation. We then went on to propose several algorithms by which one might minimize the information theoretic objective, involving both spaces of discrete mappings from data points to clusters, and spaces of continuous, real-valued weights.

In the continuous case, we considered optimization over sets of weights designed to pick out cells from the original joint probability matrix. Using a factored representation involving only row and column variables, we developed a gradient descent formulation and outlined constraints on the weights that were necessary to recover meaningful cluster mappings. In the case of discrete solution spaces, we focused first on a multi-source simulated annealing approach, where the information-theoretic objective was directly minimized by a stochastic sampling process applied to the point-to-cluster mappings. We additionally gave a multi-source spectral algorithm that performs traditional distortion based clustering, but in an embedding resulting from co-occurrence information.

Because most of the algorithms required the full joint distribution of co-occurrences to be explicitly stored, we also offered an algorithm for interleaved clustering of two dimensional marginals of an arbitrarily large joint. Any 2-way co-clustering function susceptible to biasing through initial conditions was designated as acceptable for interleaving.

Finally, we evaluated the algorithms on a real-world human identification problem, consisting of several large datasets of visual features. The experiments conducted in Chapter 4 showed roughly how the algorithms compared to one another on a moderately difficult clustering task. It was found that the annealing algorithm had the lowest error and variance on average, but also required the most computation time. The spectral algorithm required little computation time, but also performed well, giving the lowest single average clustering error on a 5 person dataset. The weight-based approach required a moderate amount of computation and gave results that did not scale well with problem complexity, although it did boast low variability in the results. That gradient methods gave *reasonable* results however, serves to explicitly illustrate the interaction of modalities through contingencies. Whereas other techniques might appear like black boxes, a good understanding of co-clustering can be gleaned from the weight-based formulation. Compared to the algorithm due to Dhillon et. al. [11], all but the weight-based approach performed better overall in terms of raw clustering error. While Dhillon's algorithm was computationally more efficient than annealing or gradient descent, that algorithm had the largest error variance.

The major conclusions of this thesis can be summarized by the following bullets:

- Multi-source co-clustering can greatly improve a mapping from points to clusters (Chapter 4) by exploiting additional related data in a principled manner (Chapter 1). This mapping can then be used to label a given dataset for subsequent learning of classifiers and evaluation of unseen examples.

- Under suitable conditions, interleaved co-clustering of the 2-way marginals of an arbitrary joint distribution (Section 2.1) can further improve a desired mapping by utilizing all available 2-way interactions, but at the expense of increased computation time. It is not clear, however, that when given one or more relatively uninformative feature sets among stronger feature sets, co-clustering of entire joint distributions is as beneficial as simply interleaving with 2-way marginals (Chapter 4).

- Clustering applied to contingency data can be more robust to noise than clustering original data points with traditional single-source algorithms (Section 4.5).
- Overall, the experiments presented in Chapter 4 suggested that it may be possible to construct a recognition system capable of training itself automatically or with little human intervention, compared to supervised analogues. Such an unsupervised surveillance system would better approximate identity recognition in humans, and can additionally facilitate continual self-adaptation that would otherwise be impossible in most supervised settings.

That multi-source clustering and labeling of faces was demonstrably successful further suggests that the algorithms discussed in this thesis can be successfully applied to a host of other difficult problem domains. The datasets used in the preceding chapter were hardly contrived synthetic examples; noisy samples and other difficulties made the clustering task very real indeed. Other inherently multi-source applications therefore have much to gain through the application of multi-source clustering methods.

In conclusion, the algorithms presented in this thesis collectively provide a framework by which to approach a large range of difficult multi-source clustering tasks. For many application domains, it is often the case that more data is better [6, 33]. Multi-source contingency clustering thus provides a powerful answer to integrating and exploiting additional sources of information under minimal constraints.

5.1 Future Work

The work presented in this thesis is far from exhaustive, and several future directions based on both the topics discussed herein and on other ideas exist. Firstly, it might be possible to take advantage of higher-order co-occurrences within a spectral clustering framework using the method of De Lathauwer et. al. [10] to compute the spectrum of an N-dimensional contingency table. Second, the annealing approach discussed in Chapter 2 might be extended to deterministic methods [26, 15], possibly involving a redesigned information-theoretic objective written in terms of row and column prototypes instead of explicit mapping functions.

In addition, there is much exploration to be done concerning clustering with functions in kernel defined spaces. In particular, kernel design to enforce clustering constraints, regularization of factored functions for clustering, and selection of an appropriate clustering-specific penalty term stand out. We will therefore briefly sketch a optimization framework for which the space of solutions of the information-theoretic objective (1.2) is a Reproducing Kernel Hilbert Space (RKHS) [2], and discuss some conditions that should be met in order to cast clustering as a Tikhonov regularization problem with interpretable solutions.

5.1.1 Continuous Optimization with Regularization

Consider for a moment the continuous analog to the problem formulation given in section 1.4.1, where all of the distributions defined are continuous functions of several variables, and summations turn into integrals. Then one reasonable definition of $P(\hat{x}, \hat{y})$ could place a weighted Gaussian bump on every region in the joint distribution corresponding to cells in the discrete case:

$$P(\hat{x}, \hat{y}) = \int\limits_{X \times Y} \int\limits_{U \times V} f(\hat{x}, \hat{y}, u, v) G([x\ y]^T; \mu = [u\ v]^T, \sigma) P(x, y)\, d\Omega,$$

where $f(\cdot)$ denotes a weighting function intended to select or deselect Gaussian bumps for a given cluster $(\hat{x}, \hat{y})$, $G(\cdot)$ denotes the Gaussian function with fixed variance parameter σ, and Ω is the differential volume $dx\, dy\, du\, dv$. An example shown Figure 5-1 illustrates selection of points with Gaussians in a two dataset (2D joint) situation.

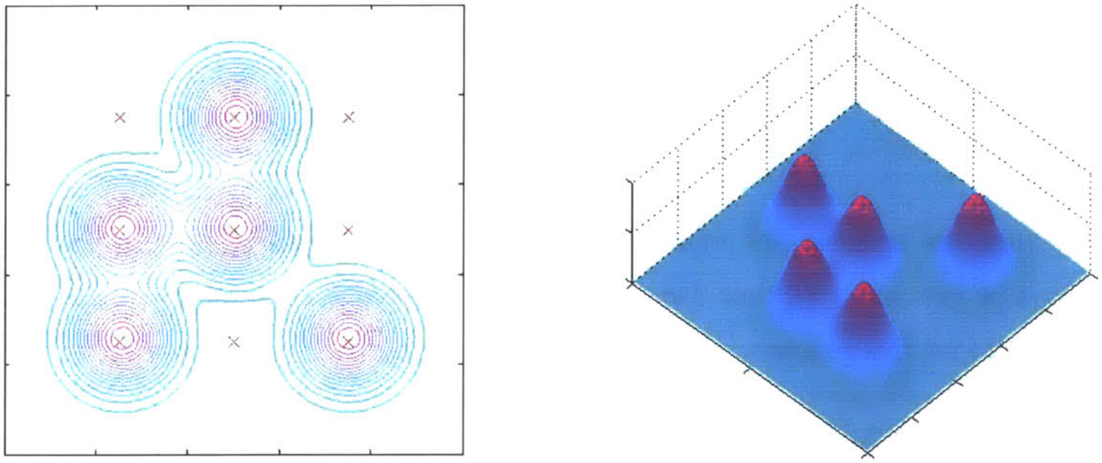

Figure 5-1: Weighted Gaussians are shown selecting "cells" from a joint distribution matrix for a given cluster $(\hat{x}, \hat{y})$. This cluster includes the points at coordinates (2,1),(3,1),(1,2),(2,2), and (3,3).

Using this definition for $P(\hat{x}, \hat{y})$, the objective is now

$$\max_{f} \int\limits_{\hat{X} \times \hat{Y}} P(\hat{x}, \hat{y}) \log \frac{P(\hat{x}, \hat{y})}{P(\hat{x})P(\hat{y})} \, d\hat{x} \, d\hat{y}.$$

The motivation behind choosing this particular representation for $P(\hat{x}, \hat{y})$ is that we would hope to be able to recover the cluster mappings from the final solution by simply looking at which bumps are selected by $f(\hat{x}, \hat{y}, \cdot, \cdot)$ for a given cluster $(\hat{x}, \hat{y})$.

Alternatively, we might instead just try to learn the function

$$h(\hat{x}, \hat{y}, x, y) = \int\limits_{U \times V} f(\hat{x}, \hat{y}, u, v) \, G([x \; y]^T; \mu = [u \; v]^T, \sigma) \, du \, dv,$$

and enforce smoothness with a regularization term. This regularization penalty would then effectively constrain the optimization over a class of functions lying in a ball within a reproducing kernel Hilbert space (RKHS) defined by an appropriate kernel K [2, 9]. In this case $P(\hat{x}, \hat{y})$ becomes

$$P(\hat{x}, \hat{y}) = \sum_{x,y} h(\hat{x}, \hat{y}, x, y) P(x, y)$$

and we can now write the new optimization problem as

$$\max_{h\in\mathcal{H}} \sum_{\hat{x},\hat{y}} P(\hat{x},\hat{y}) \log \frac{P(\hat{x},\hat{y})}{P(\hat{x})P(\hat{y})} + \lambda \|h\|_K^2. \tag{5.1}$$

We note that both the risk term in (5.1) and $h(\cdot)$ can be bounded below by zero, and above by one, implying that β stability and the associated convergence results in [5] may hold for this problem.

Interestingly, in the case of the weighting function $f(\cdot)$ above, we would *not* necessarily want the learned function to be smooth, since the function is intended to pick out points which are to be included in a given cluster. Function values corresponding to cells in the joint probability distribution may rapidly step between zero or one in the ideal case. For the function $h(\cdot)$ however, we can allow a great deal more smoothness since in this case we are looking at block-like groups of Gaussians which may, on the whole, be smoothed over large regions of the joint approximation. The question of smoothness therefore seems to argue against using regularization, and considering smoothness alone, we would not want to impose a complexity penalty. The regularization term is exceedingly important, however, when viewed from the perspective of stability: if we slightly perturb the joint distribution $P(x, y)$ we certainly would not want the resulting clustering to change much. Thus, we will sacrifice complexity (and increase smoothness) in exchange for stability. In addition, it may be the case that *some* degree of complexity reduction is desirable if the inherent complexity of $h(\cdot)$ is actually quite low. This is indeed the case if we allow only row/column intersection points to fall into the clusters, in which case h can be written as:

$$h(\hat{x}, \hat{y}, x, y) = h_{row}(\hat{x}, x) \cdot h_{col}(\hat{y}, y).$$

In fact, in order to recover independent mappings along each modality's dimension of the joint, h must factor in this way. In the absence of such a prior knowledge, however, we would hope that this factorization could be implicitly realized by constraining the solution to lie within an RKHS over which we have control.

Optimization Concerns

Optimization of the functional (5.1) can be accomplished by applying Wahba's representer theorem [21], which states that under certain conditions the solution to a regularized risk functional can be written in terms of the data. Let us temporarily combine the variables of interest into a vector: $\mathbf{x} = [\hat{x}\ \hat{y}\ x\ y]^T$. The fact that the risk term in (5.1) admits a solution of the form

$$h(\mathbf{x}) = \sum_i c_i K(\mathbf{x}_i, \mathbf{x}) \tag{5.2}$$

can be seen a bit more clearly by bringing the $I(X;Y)$ term in (1.2) into the summation in (5.1), and minimizing the loss in mutual information:

$$\min_{h\in\mathcal{H}} \sum_{\hat{x},\hat{y}} \Big(I(X;Y) - P(\hat{x},\hat{y}) \log \frac{P(\hat{x},\hat{y})}{P(\hat{x})P(\hat{y})} \Big) + \lambda \|h\|_K^2.$$

This regularized risk functional is of the generic form

$$c\big((x_1, y_1, h(x_1)), ..., (x_N, y_N, h(x_N))\big) + g(\|h\|),$$

and the Representer Theorem thus applies [27]. Substituting the representation (5.2) into the risk functional (5.1) gives an equivalent continuous optimization problem over the set of real-valued weights $\{c_i\}$.

While the optimization may be straight forward at this point, there is one remaining difficulty to be addressed: in general, we cannot be sure that the solution will allow recovery of the cluster mappings, and must use a kernel that can enforce constraints particular to clustering. We might require that 1) each point is assigned to one and only one cluster, and possibly, 2) each cluster must have at least one member. For instance, given a function of the form (5.2), it could be stipulated that the the kernel K and weights $\mathbf{c}$ collectively satisfy

$$\int_{\hat{X}} h_{row}(\hat{x}, x)\, d\hat{x} = \int_{\hat{X}} \sum_{\hat{x}',x'} c_{\hat{x}',x'} K([\hat{x}'\ x']^T, [\hat{x}\ x]^T)\, d\hat{x} = 1,\ \forall x,$$

with a similar constraint on $h_{col}(\hat{y}, y)$. In order to cluster within a regularization framework, we would need to choose a kernel that can sufficiently bias the solution towards enforcing clustering-specific constraints and allow for recovery of cluster members from the solution itself. If the solution which minimizes the objective is meaningless in terms of mappings from points to clusters, then that solution is clearly useless for clustering purposes. It is therefore of paramount importance that we choose a kernel that can enforce the constraints above to give sufficient recoverability while maintaining acceptable performance as defined by the clustering objective function. In the event that such a kernel cannot be found for a given problem, then optimization over an RKHS may not be appropriate.

Bibliography

[1] C. Aggarwal. Towards Systematic Design of Distance Functions for Datamining Applications. *Proc. ACM SIGKDD Conf.*, 9-18, 2003.

[2] N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 686:337-404, 1950.

[3] M. Bazaraa, H. Sherali, and C. Shetty. *Nonlinear Programming: Theory and Algorithms, 2nd. Ed.*, Wiley & Sons, New York, 1993.

[4] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-Training. *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp.92-100, 1998.

[5] O. Bousquet, and E. Elisseeff. Stability and Generalization. *Journal of Machine Learning Research*, 2(Mar):499-526, 2002.

[6] V. Castelli, and T. Cover. The Relative Value of Labeled and Unlabeled Samples in Pattern Recognition with an Unknown Mixing Parameter *IEEE Trans. Info. Theory*, 42(6):2102-2117, 1996.

[7] V. Cherkassky, and F. Mulier. *Learning from Data: Concepts, Theory, and Methods.* Wiley & Sons, New York, 1998.

[8] T. Cover and J. Thomas. *Elements of Information Theory*, Wiley & Sons, New York, 1991.

[9] F. Cucker, and S. Smale. On the Mathematical Foundations of Learning. *Bulletin of the American Mathematical Society*, 2002.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle. A Multilinear Singular Value Decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253-1278, 2000.

[11] I. Dhillon, S. Mallela, and D. Modha. Information-Theoretic Co-clustering. *Proc. ACM SIGKDD '03*, August 24-27, Washington, DC, USA, 2003.

[12] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley, New York, 2001.

[13] N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate Information Bottleneck. *Proc. UAI 2001*, pp.152-161.

[14] B. Heisele, P. Ho and T. Poggio. Face Recognition with Support Vector Machines: Global Versus Component-based Approach. *ICCV'01, Vancouver, Canada*, Vol. 2, pp. 688–694, 2001.

[15] T. Hofmann, and J. Buhmann. Pairwise Data Clustering by Deterministic Annealing. *IEEE Tr. on Pat. Anal. and Mach. Int.*, 19(1):1–4, 1997.

[16] Y. Ivanov, B. Blumberg and A. Pentland. Expectation-Maximization for Weakly Labeled Data. *18th International Conference on Machine Learning*, Williamstown, MA, June 2001.

[17] T. Jaakkola, M. Meila, and T. Jebara. Maximum Entropy Discrimination, *NIPS 12*, MIT Press, 1999.

[18] A.K. Jain, and R.C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Upper Saddle River, USA, 1988.

[19] B. Kim. Multi-Source Human Identification. *M.Eng. Thesis, EECS, MIT,* June 2003.

[20] S. Kirpatrick, C. Getall, and M. Vecchi. Optimization by simulated annealing, *Science*, 220:671-680, 1983.

[21] G. Kimeldorf and G. Wahba. A correspondence between Bayesian estimaton on stochastic processes and smoothing by splines. *Ann. Math. Stat.*, 2:495-502, 1971.

[22] C. Nakajima, M. Pontil, B. Heisele and T. Poggio. Full-body person recognition system. *Pattern Recognition*, Volume 36, Issue 9, September 2003, pp.1997-2006.

[23] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. In T. Dietterich et. al. eds., *Advances in Neural Information Signal Processing 14*, MIT Press, Cambridge, MA, 2002.

[24] K. Nigam and R. Ghani. Analyzing the Effectiveness and Applicability of Co-training. *Ninth International Conference on Information and Knowledge Management (CIKM-2000)*, pp. 86-93. 2000.

[25] D. Pierce and C. Cardie. Limitations of Co-Training for Natural Language Learning from Large Datasets. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-2001).*

[26] K. Rose. Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems. *Proc. of the IEEE*, 86(11), Nov., 1998.

[27] B. Scholkopf, R. Herbrich, A. Smola, and R. Williamson. A Generalized Representer Theorem. *Proc. of the Annual Conference on Computational Learning Theory*, 416-426, 2001.

[28] N. Slonim, N. Friedman, and N. Tishby. Unsupervised Document Classification using sequential information maximization. *ACM SIGIR*, 2002.

[29] N. Tishby, F. Pereira, and W. Bialek. The Information Bottleneck Method. In *Proc. 27th Allerton Conference on Communication and Computation*, 1999.

[30] L. Trefethen, and D. Bau. *Numerical Linear Algebra*, SIAM Press, Philadelphia, USA, 1997.

[31] V. Vapnik. *Statistical Learning Theory*, Wiley, New York, USA, 1998.

[32] P. Viola, and W. Wells. Alignment by Maximization of Mutual Information. *Intl. J. of Comp. Vis.*, 24(2):137-154, 1997. 1999.

[33] T. Zhang, and F. Oles. A Probability Analysis on the Value of Unlabeled Data for Classification Problems. In *Proc. ICML 2000*, pp.1191–1198, 2000.