UNIVERSITY OF CALIFORNIA,
IRVINE

Exploring Entity Resolution for Multimedia Person Identification

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Liyan Zhang

Dissertation Committee:
Professor Sharad Mehrotra, Chair
Professor Ramesh Jain
Professor Dmitri V. Kalashnikov
Professor Deva Ramanan
Professor Nalini Venkatasubramanian

2014

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

Foremost, I would like to express my deepest gratitude to my PhD advisor, Professor Sharad Mehrotra, for his advice and guidance during my Ph.D. program at University of California, Irvine. I would like to thank him for his continuous support, patience, motivation, enthusiasm, and immense knowledge in the past five years. He has been always very supportive not just to my research and living in U.S., but also to my professional development. I deeply appreciated his encouragement for me on discovering novel ideas, discussing with other researchers, and pursuing academic career. I could not have imagined having a better advisor and mentor for my PhD study.

I am also very grateful to the professors in my dissertation committee, Professor Ramesh Jain, Professor Dmitri V. Kalashnikov, Professor Deva Ramanan, and Professor Nalini Venkatasubramanian. They have provided plenty of insightful comments and advice to my research work. I want to thank them for their encouragement and support to my research during my PhD study.

In addition, I appreciate the help and support from all of my labmates and colleagues, Zhijing Qin, Ronen Vaisenberg, Jie Xu, Hotham Altwaijry, Yasser Altowim, Ye Zhao and Yingyi Bu. Besides, I would like to thank my friends Zhi Chen, Wei Zhang, Jingwen Zhang, Yongxue Li, Xiujuan Yi, and all the other friends here who have offered me help, support, and joy.

Finally, I would like to especially thank my family for giving me continuous love and care. I could not have done it without them and their supports.

The materials of some chapters in this dissertation have appeared in “Context-based Person Identification Framework for Smart Video Surveillance” ⓒ 2013 Springer by L. Zhang et al. and “Context Assisted Face Clustering Framework with Human-in-the-Loop” ⓒ 2014 Springer by L. Zhang et al. The co-author listed in this publication directed and supervised research which forms the basis for the thesis/dissertation.

CURRICULUM VITAE

Liyan Zhang

EDUCATION

Doctor of Philosophy in Computer Science **2009–2014**
University of California, Irvine *Irvine, CA*

Master of Engineering in Software Engineering **2006–2009**
Tsinghua University *Beijing, China*

Bachelor of Science in Computer Science **2002–2006**
Hebei University of Technology *Tianjin, China*

WORKING EXPERIENCE

Graduate Research Assistant **2009–2014**
University of California, Irvine *Irvine, California*

Senior Research Intern **2013–2013, 2012–2012, 2011–2011**
Canon U.S.A., Inc *Irvine, California*

Research Intern **2008–2009**
School of Computing, National University of Singapore *Singapore*

Engineering Intern **2007–2008**
SUN China Engineering and Research Institute *Beijing, China*

English Teacher **2006–2007**
New Oriental School *Tianjin, China*

REFERRED PUBLICATIONS

Query-driven Approach to Face Clustering and Tagging **2014**
Liyan Zhang, Dmitri V. Kalashnikov, Sharad Mehrotra, Deva Ramanan

In submission

Context Assisted Face Clustering Framework with Human-in-the-Loop **2014**
Liyan Zhang, Dmitri V. Kalashnikov, Sharad Mehrotra

International Journal of Multimedia Information Retrieval (IJMIR)

A Collaborative Approach for Face Verification and Attributes Refinement **2014**
Liyan Zhang, Bradley Denney, Juwei Lu

Information Sciences

Context-based Person Identification Framework for Smart Video Surveillance **2013**
Liyan Zhang, Dmitri V. Kalashnikov, Sharad Mehrotra

Machine Vision and Applications (MVA)

A Unified Framework for Context Assisted Face Clustering **2013**
Liyan Zhang, Dmitri V. Kalashnikov, Sharad Mehrotra

ACM International Conference on Multimedia Retrieval (ICMR)

Cross-Space Affinity Learning with Its Application to Movie Recommendation **2013**
Jinhui Tang, Guo-Jun Qi, Liyan Zhang, Changsheng Xu

IEEE Transactions on Knowledge and Data Engineering(TKDE)

Video Entity Resolution: Applying ER Techniques for Smart Video Surveillance **2011**
Liyan Zhang, Ronen Vaisenberg, Sharad Mehrotra, Dmitri V. Kalashnikov

IQ2S in conjunction with IEEE PERCOM 2011

A Topical PageRank based Algorithm for Recommender Systems **2008**
Liyan Zhang, Kai Zhang, Chunping Li

ACM SIGIR

ABSTRACT OF THE DISSERTATION

Exploring Entity Resolution for Multimedia Person Identification

By

Liyan Zhang

Doctor of Philosophy in Computer Science

University of California, Irvine, 2014

Professor Sharad Mehrotra, Chair

The explosion of massive media data induced by the proliferation of digital cameras, mobile devices as well as the emergence of online media websites, has led us into the era of big data. Accurate and effective analyses of the big multimedia data to support semantically enriched representation in terms of events, activities, and entities can bring transformative improvements to a variety of application domains. The basic form of multimedia analysis for more sophisticated interpretation is characterized by questions such as "who, what, where, when" that identify subjects, activities, locations, and time associated with images/video segments. In this thesis, we primarily focus on the "who" question, which is referred as the person identification problem in multimedia data.

While advances in image processing and computer vision has resulted in powerful techniques for person identification, such techniques based on the facial appearance representations, are usually prone to errors due to a variety of factors including noise, poor signal quality, occlusion, etc. It is widely recognized in the multimedia research community that additional contextual features can be leveraged to bring significant improvements to such tasks. Nevertheless, how to systematically utilize the heterogeneous contextual information still poses a big challenge. Besides, the person identification procedure is conventionally processed in an "offline" setting where the typical goal is to achieve complete annotations of the whole

collection before further applications. Such an “offline ” process is not tenable when dealing with big multimedia data, since the limitation of computational resources as well as restriction of manpower does not allow us to process every image with each possible tag and clean up every potentially noisy result.

We note that similar challenges also arise in the database domain, especially for the entity resolution task. To address these challenges, recent entity resolution research has explored a series of powerful methods including techniques to exploit relationships, contextual information, domain semantics in the form of constraints and ontologies, etc. for the purpose of resolving references in structured/semi-structured and unstructured textual data. Additionally, query-driven data cleaning techniques have also been proposed and explored to resolve the challenges of big data.

In this thesis, we aim to explore how such advanced entity resolution techniques can be exploited to improve semantic interpretation of multimedia data, specifically for the person identification problem. We first explore how to leverage the automatic data cleaning techniques to exploit relationships, contextual information, domain semantics, constraints, etc., to enhance the performance of face clustering and recognition. Then, we propose the new paradigm for face clustering/tagging suited for big data where image enrichment is seamlessly integrated into the image retrieval/analysis process – we refer to this new paradigm as “query-driven image enrichment”.

Particularly, we first study the person identification problem in the context of surveillance videos and propose a context-based framework for low-quality data, which integrates multiple contextual information leveraging the entity resolution framework called RelDC to improve the performance of person identification. Inspired by the significant results improvement, we investigate the face clustering problem and propose a unified framework that employs bootstrapping to automatically learn adaptive rules to integrate heterogeneous context information together. Furthermore, we exploit the human-in-the-loop mechanism to leverage

human interaction to achieve high quality clustering results. Later, to address the challenges of big multimedia data, we propose the query-driven approach to face clustering/tagging which investigates the query-driven active learning strategies in order to achieve the accurate query answers with minimum user participation.

Chapter 1

Introduction

The prevalence of digital cameras, surveillance cameras as well as the emergence of online media websites makes the creation, storage and sharing of multimedia content much easier than before, leading to the explosion of massive media data. Recent estimates suggest that by 2018, more than 80% of total traffic on the internet will consist of video transmissions, and more than 50% of total traffic will originate from non-PCs (e.g., mobile) devices [2]. These statistics suggest that big multimedia data from heterogeneous sources will play a vital role in the future, supporting a variety of end applications.

In the era of big multimedia data, accurate and effective analyses of raw multimedia data to support semantically enriched representation in terms of events, activities, and entities can bring transformative improvements to variety of application domains (e.g., surveillance), as well as, enable a large number of new applications to be built. We observe that much of the semantic interpretation of multimedia data is driven by and built on top of simpler analysis of images/video shots characterized by questions such as *who, where, what*, and *when* [70, 67]. In the above, *who* refers to the person(s) or subject(s) of interest in the image/video shot, *what* to the activity they are involved in, *where* to the location and *when* to the time.

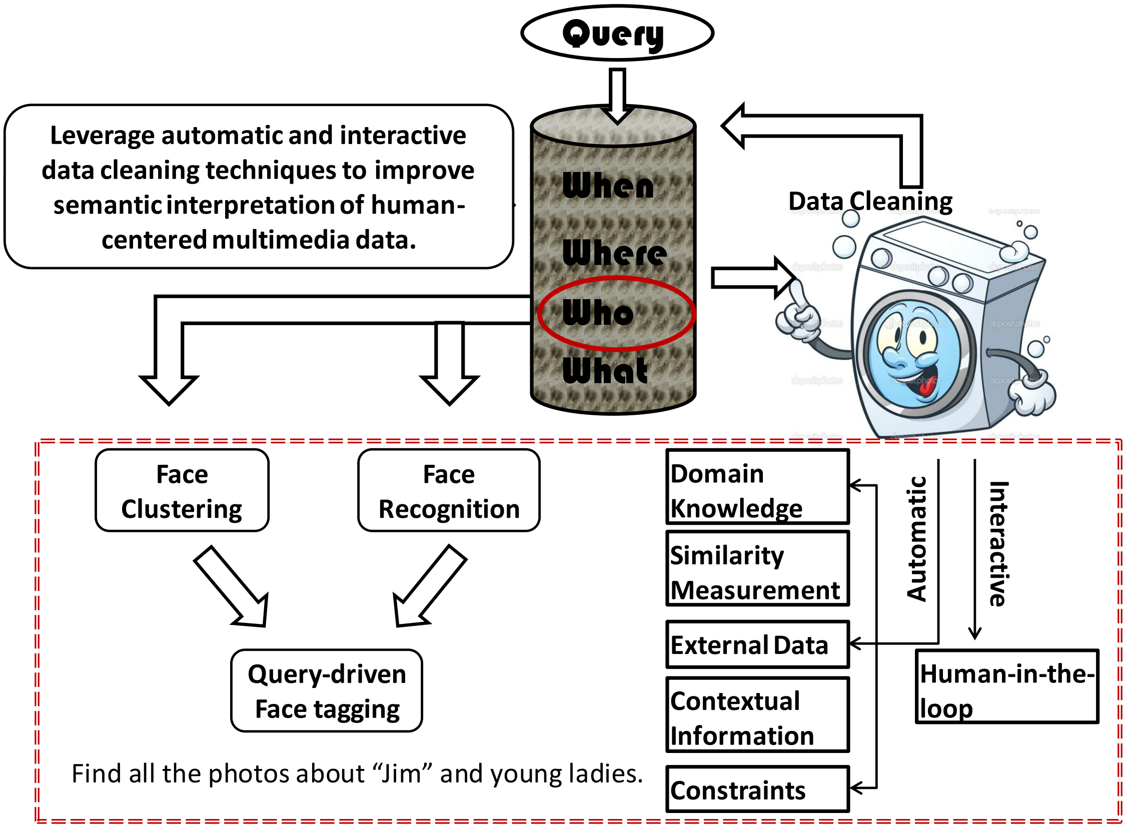

Query
Leverage automatic and interactive data cleaning techniques to improve semantic interpretation of human-centered multimedia data.
When
Where
Who
What
Data Cleaning
Face Clustering
Face Recognition
Query-driven Face tagging
Find all the photos about "Jim" and young ladies.
Domain Knowledge
Similarity Measurement
External Data
Contextual Information
Constraints
Automatic
Interactive
Human-in-the-loop

Figure 1.1: The goal of this thesis is to explore the automatic and interactive data cleaning techniques for semantic interpretation of big multimedia data. Specifically, we mainly focus on "who" problem, referred as person identification problem. We first exploit the automatic techniques which can integrate heterogenous contextual information as well as domain semantics to enhance the performance of face clustering and recognition. Then, to address the challenges of big data, we propose the query-driven paradigm for face tagging which investigates the interactive approaches to help users achieve accurate query answers.

Building on top of such a representation of images/video shots, one can apply sophisticated spatial/spatio-temporal analyses to built higher-level semantic representations captured in the visual media. Of the four questions above, often when (i.e., time) and where (i.e., location) are relatively simple to determine – time could be deciphered using the clock associated with the camera, and location can be ascertained through the identity of the camera (in case of fixed in-situ cameras) or through GPS and/or other localization technologies such as WiFi localization readily available in emerging mobile devices. In contrast, activity (what) and people identification (who) are significantly more complex. In this thesis, we aim to explore the "who" problem, referred as person identification in multimedia data.

In the remaining of this chapter, we first present motivation behind our research in Sec. 1.1. We then state our thesis problem, research challenges and lay out the scope of work within the thesis in Sec. 1.2. Finally, we highlight our contributions in Sec. 1.3.

1.1 Motivation

While advances in image processing and computer vision has resulted in powerful techniques for person identification, such image analysis techniques are prone to errors due to a variety of factors including noise, poor signal quality, occlusion, etc. It is widely recognized in the multimedia research community that additional contextual features can be leveraged to bring significant improvements to such tasks. Nevertheless, how to systematically utilize the heterogeneous contextual information still poses a big challenge. Additionally, the person identification procedure is conventionally processed in an “offline” setting where the typical goal is to achieve complete annotations of the whole collection before further applications. Such an “offline ” process is not tenable when dealing with big multimedia data, since the limitation of computational resources as well as restriction of manpower does not allow us to process every image with each possible tag and clean up every potentially noisy result. Therefore, facing the challenges of big multimedia data, the conventional “offline” setting of person identification which mainly relies on the facial features is not tenable. In the following, we enumerate concrete examples to describe the limitations of conventional person identification approaches.

Fail to detect faces with poor quality data. In the domain of computer vision, the most straightforward way to identify a person is to perform face detection followed by face recognition. However, in the resource constrained environments, where transmission delay, bandwidth restriction, and packet loss may prevent the capture of high quality data, face detection and recognition becomes more complex. We have experimented with Picasa’s face

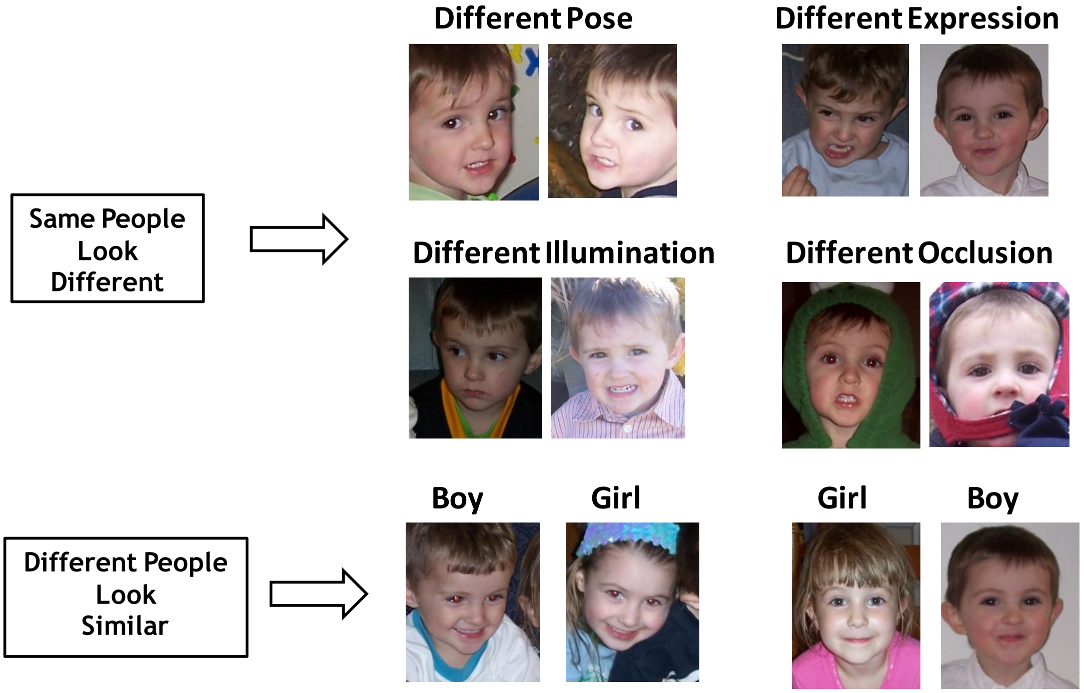


Figure 1.2: Examples to illustrate the limitations of facial appearance based features. Faces from the same person may look different due to the variation of pose, expression, illumination, occlusion,etc.; Faces from different persons may look alike if appearing with similar poses and expression.

detector on the smart surveillance video dataset, and found that it can detect faces in only 7% of the cases and then among them it can recognize only 4% of faces. Several reasons account for the low detection rate: (1) faces can not be captured if people walk with their back to the cameras; (2) faces are too small to be detected when people are far away from cameras; (3) the large variation of people's pose and expression brings more challenges to face detection. Thus the traditional face detection and recognition techniques are not sufficient to handle the poor quality surveillance video data.

Facial appearance features are not sufficient. The most conventional approach for face recognition is facial appearance based methods, which has been extensively explored in the prior literature. Although significant progress has been achieved, where these standard approaches can achieve good performance under controlled conditions, they usually tend to suffer dealing with the uncontrolled situations where faces are captured with a large variation of pose, expression, illumination, etc. These nuisance factors might cause the differences between faces describing the same person captured in distinct conditions to be larger than those between two different people in similar conditions, as illustrated in Figure 1.2.

Challenges of utilizing contextual information. Realizing the limitations of facial appearance based person identification approaches, research in multimedia domain has shifted to context assisted methods. Although prior research efforts have extensively explored using contextual features to improve the quality of person identification, such techniques often aim at exploring just one (or a few) contextual feature types, with the decision often made at the image level only. How to develop a unified framework that integrates heterogeneous context information together to improve the performance of disambiguation is still a challenge.

Limitations of "offline" setting for face tagging. Face tagging which aims to guide users to annotate the identity of faces is an important component in image analysis systems. While face tagging has been extensively studied in the literature, it has largely been studied as an "offline" process where the typical goal is to achieve accurate and complete annotation of the entire collection with least amount of human input. Such an "offline" process for tagging, however, is unsuited if data collections are very large or highly dynamic as is getting to be the case in the era of big data. We simply do not have enough computational resources to process every image with each possible tags, and we will not have adequate manpower to clean the potentially noisy data.

1.1.1 Our Approach

Entity Resolution. We note that similar challenges also arise in the database domain, especially for the entity resolution task. Entity resolution, developed primarily in the context of structured/semi-structured and unstructured textual data, aims to disambiguate entity/-concept references even when the same entity may be represented differently in different contexts/data sources. Such reference disambiguation tasks are typically modeled as either classification (often referred to as lookup in data cleaning literature) or as clustering (often referred to as grouping) tasks. Recent entity resolution research has explored a series of powerful methods including techniques to exploit relationships, contextual information, domain semantics in the form of constraints and ontologies, etc. for the purpose of resolving references in structured/semi-structured and unstructured textual data. Additionally, query-driven data cleaning techniques have also been proposed and explored to resolve the challenges of big data.

Multimedia Person Identification. Person identification aims to identify the same real-world person appearing in different images/videos, which task are usually modeled as face recognition (classification problem) or face clustering (clustering problem). We also observe that domain semantics as well as contextual information can provide extra clues for person identification. For instance, in the scenario of surveillance video, a person entering the office space first in a given day is quite likely the owner associated with the space. Thus, multimedia person identification can be connected with entity resolution since they share the similar goal, tasks, and approaches.

Therefore, with the connection between entity resolution and person identification, the goal of this thesis is to explore how such advanced entity resolution techniques can be exploited to improve semantic interpretation of multimedia data, specifically for the person identification problem. We first explore how to leverage the automatic data cleaning techniques to exploit relationships, contextual information, domain semantics, constraints, etc., to enhance the performance of face clustering and recognition. Then, we propose the new paradigm for face clustering/tagging suited for big data where image enrichment is seamlessly integrated into the image retrieval/analysis process – we refer to this new paradigm as “query-driven image enrichment”.

1.2 Thesis Problem, Challenges, and Scope

1.2.1 Thesis Problem

In this thesis, we aim to explore the entity resolution framework specifically suited for analyzing and improving semantic interpretation of visual data, with the person identification problem. Such framework can be further leveraged to develop a new multimedia database technology that can support SQL style retrieval/analyses on the big multimedia data. A quick example in the scenario of surveillance videos to illustrate usefulness of such retrieval technology can be "retrieve images of all visitors to Bren Hall who met with the database faculty". With the growing importance of visual data, the importance of both better tools to interpret multimedia data and technology to support accurate query processing/analyses on the interpreted data will substantially increase.

Since the task of person identification can be modeled as either face recognition or face clustering, we plan to first explore the automatic techniques which can leverage contextual information, domain semantics, constraints, etc., to improve the performance of such tasks. To further support the SQL style analyses of big multimedia data, we will explore the query-driven approaches which can leverage interactive data cleaning capable of exploiting human input during query processing to reduce uncertainty in the results that arises naturally due to uncertainty in the underlying semantic representation of data.

1.2.2 Challenges

Developing data cleaning methods for improving multimedia semantically interpretation and supporting SQL queries on big multimedia data poses certain challenges.

Contextual Information Utilization. The first challenge arises from the observation that rich domain semantics might be available to help interpret data. Since visual data captures information about the real-world, many of the real world constraints and semantics can be encoded and exploited for the purpose of disambiguation. Such domain semantics may be in the form of patterns – e.g., a person entering the office space first in a given day is quite likely the owner associated with the space, or constraints – e.g., a person cannot be at two locations at the same time, a person cannot show up in the same image/frame more than once, etc. Such patterns/constraints may either be straightforward to specify by an analyst or might alternatively be learnt from the data. In either case, such constraints and patterns can then be leveraged for the purpose of semantic interpretation of multimedia data. While data cleaning literature, specifically, data repair [88, 13, 23, 25, 57] research has considered exploiting domain semantics, the work is limited to specific types of constraints (e.g., cardinality constraints in [8, 10, 18], and various forms of functional dependencies [25, 30]). How to appropriately extract and leverage the heterogenous domain semantics and contexture information to improve the multimedia semantic interpretation is a big challenge. Meanwhile, the rich set of semantic constraints and patterns in multimedia data also provides a new avenue of research.

Query-driven approaches. Another challenge we will explore is how to design the query-driven strategy to support SQL queries on the big multimedia data with uncertainty. Particularly, we need to exploit "human-in-the-loop" techniques to reduce uncertainty in the semantic representation of multimedia data during query answering. Such techniques represent, in our view, a new direction of research for not just multimedia data cleaning but data cleaning in general. While the query-driven approach is attractive, it opens a whole set of new challenges that one must address in developing such an approach.

- We need to design a proper framework to perform queries on the probabilistic multimedia database with uncertainties. Due to the big volume of multimedia data, it is infeasible to process every image with every possible tags and clean all the potential noise data before hand. Therefore, we need to process queries on the multimedia data with plenty of noises and uncertainties.

- We must decide on the type of questions that the system can ask of a user during retrieval/analysis session. Not each question is equally easy for the user to answer – so one needs to develop a cost model for questions based on the burden answering the question poses to the user.

- Once we have fixed the type of questions, we need to decide on which set of such questions will benefit the query / analysis task at hand. The goal might be to choose a set of questions such that the quality of the answer to the original query progressively becomes better at the highest possible rate. Note that the system may not ask the user a question one at a time, but in a batch.

Recent work in data cleaning have begun to explore how human input obtained through crowd-sourcing [82] could be used to improve data cleaning models. Our goal, in contrast is, however, to reduce uncertainty in query results by exploiting feedback from the end-user directly by asking the user a (small number of) relevant questions that can help minimize the uncertainty. Of course, a side effect of the user-feedback could be reduction in uncertainty in data as well as improvements to models used by the system for disambiguation thereby improving performance of future queries. The query context for feedback brings a new dimension to the problem. Note that the problem being addressed is significantly different compared to work in the multimedia community on relevance feedback [65, 15, 66, 14]. The goal there was to seek user-feedback to better understand/represent the user's intent. The goal here is to use feedback to overcome the underlying uncertainty in data (and not the query).

1.2.3 Thesis Scope

To address the aforementioned challenges concretely, in this thesis we concentrate on three real-world applications. The first problem focuses on the person identification problem in the context of surveillance videos – we aim to improve the performance of person identification leveraging heterogeneous contextual information. In the second problem, we concentrate on exploring a context assisted framework for face clustering task. Finally, facing the challenges of big multimedia data, we explore a new paradigm for face clustering/tagging, called query-driven image enrichment, which aims to leverage interactive techniques to achieve accurate query answers with minimum user participation.

1.3 Thesis Contributions and Organization

In this dissertation, we explore the automatic as well as interactive data cleaning techniques to improve the semantic interpretation of big multimedia data. We specially concentrate on the aforementioned three concrete tasks to address the challenges. The major contributions of our work are listed below:

- We propose a context-based person identification framework for Smart Video Surveillance [98, 97]. Smart video surveillance (SVS) applications enhance situational awareness by allowing domain analysts to focus on the events of higher priority. SVS approaches operate by trying to extract and interpret higher "semantic" level events that occur in video. One of the key challenges of SVS is that of person identification where the task is for each subject that occurs in a video shot to identify the person it corresponds to. The problem of person identification is especially challenging in resource-constrained environments where transmission delay, bandwidth restriction, and packet loss may prevent the capture of high-quality data. Conventional person identification approaches which primarily are based on analyzing facial features are often not sufficient to deal with poor-quality data. To address this challenge, we propose a framework that leverages heterogeneous contextual information together with facial features to handle the problem of person identification for low-quality data. We first investigate the appropriate methods to utilize heterogeneous context features including clothing, activity, human attributes, gait, people co-occurrence, and so on. We then propose a unified approach for person identification that builds on top of our generic entity resolution framework called RelDC, which can integrate all these context features to improve the quality of person identification. This work thus links one well-known problem of person identification from the computer vision research area (that deals with video/images) with another well-recognized challenge known as entity resolution from the database and AI/ML areas (that deals with textual data). We apply the proposed solution to a real-world dataset consisting of several weeks of surveillance videos. The results demonstrate the effectiveness and efficiency of our approach even on low-quality video data.

- we present a unified framework for context assisted face clustering [95]. Automatic face clustering, which aims to group faces referring to the same people together, is a key component for face tagging and image management. Standard face clustering approaches that are based on analyzing facial features can already achieve high-precision results. However, they often suffer from low recall due to the large variation of faces in pose, expression, illumination, occlusion, etc. To improve the clustering recall without reducing the high precision, we leverage the heterogeneous context information to iteratively merge the clusters referring to same entities. We first investigate the appropriate methods to utilize the context information at the cluster level, including using of "common scene", people co-occurrence, human attributes, and clothing. We then propose a unified framework that employs bootstrapping to automatically learn adaptive rules to integrate this heterogeneous contextual information, along with facial features, together. Experimental results on two personal photo collections and one real-world surveillance dataset demonstrate the effectiveness of the proposed approach in improving recall while maintaining very high precision of face clustering.

- We discuss a novel methodology for integrating human-in-the-loop feedback mechanisms that leverage human interaction to achieve the high-quality clustering and face tagging results [96]. Automatic cleaning techniques can help us to improve the semantic interpretation of media data; however, it is difficult to get the very high quality data merely using automatic techniques. Therefore, we turn to the help from user feedback, that is, by using human-in-the-loop techniques. The high-level goal of such techniques is to be able to get very high quality face clustering results while minimizing user participation. We first present an appropriate user interface, and then design the question-asking strategy which considered both relevance and impact to rank questions. The experiments demonstrate that the proposed techniques outperform various baselines and reach higher quality results.

- We propose a query-driven approach to face clustering/tagging. In the era of big data, a traditional "offline" setting to processing image data is simply not tenable. We simply do not have the computational power to process every image with every possible tag, and moreover, we will not have the manpower to clean up the potentially noisy results. We introduce a query-driven approach to visual tagging, focusing on the application of face tagging and clustering. We integrate active learning with query-driven probabilistic databases. Rather than asking a user to provide manual labels so as to minimize the uncertainty of labels (face tags) across the entire dataset, we ask the user to provide labels that minimize the uncertainty of his/her query result (e.g., "How many times did Bob and Jim appear together?").We use a data-driven Gaussian process model of facial appearance to write probabilistic estimates of facial identity into a probabilistic database, which can then support inference through query answering.

Importantly, the database is augmented with contextual constraints (faces in the same image cannot be the same identity while faces in the same track must be identical). Experiments on the real-world photo collections demonstrate the effectiveness of the proposed method.

The dissertation is organized as follows. In Chapter 2, we present the related work of person identification task and entity resolution techniques in database domain. Chapter 3 is dedicated to present our research of context-based person identification framework for smart video surveillance. In Chapter 4, we first study the unified context assisted framework for face clustering, then we explore integrating the "human-in-the-loop" strategy to further improve the performance of face clustering. Chapter 5 presents query-driven approaches to face clustering/tagging to address the challenges of big data. Finally, in Chapter 6, we conclude the thesis and discuss some interesting future research directions.

Chapter 2

Related Work

Since the goal of this thesis is to explore the entity resolution techniques to improve the semantic interpretation for big multimedia data, especially for the person identification problem, in this chapter, we will first discuss the existing techniques for entity resolution. Then, we will proceed to describe the current approaches for person identification tasks in the context of surveillance videos as well as personal photos.

2.1 Entity Resolution

High quality of data is a fundamental requirement for effective data analysis which is used by many scientific and decision-support applications to learn about the real-world and its phenomena [40, 49, 39]. However, many Information Quality (IQ) problems such as errors, duplicates, incompleteness, etc., exist in most real-world datasets. Among these IQ problems, *Entity Resolution* (also known as deduplication or record linkage) is among the most challenging and well studied problem. It arises especially when dealing with raw textual data, or integrating multiple data sources to create a single unified database. The essence of ER problem is that the same real-world entities are usually referred to in different ways in multiple data sources, leading to ambiguity. For instances, the real-world person name 'John Smith' might be represented as 'J. Smith', or misspelled as 'John Smitx'. Besides, two distinct individuals may be referred as the same representation, e.g., both 'John Smith' and 'Jane Smith' referred as 'J. Smith'. Therefore, the goal of ER is to resolve these entities by identifying the records representing the same entity.

There are two main instances for ER problem: *Lookup*[46, 16] and *Grouping* [58, 16]. *Lookup* is a classification problem, with the goal of identifying the object that each reference refers to. *Grouping* is a clustering problem, which goal is to correctly group the representations that refer to the same object. We primarily will be interested in an instance of the lookup problem. Our research group at the University of California, Irvine has also contributed significantly to the area of ER in the context of Project Sherlock@UCI, e.g., [44, 95, 63, 62, 64, 21, 45]. The most related work of our group is summarized next.

2.1.1 Relationship-Based Data Cleaning (RelDC)

To address the entity resolution problem, we have developed a powerful disambiguation engine called the *Relationship-based Data Cleaning* (RelDC) [46, 17, 47, 61, 19, 48]. RelD-C is based on the observation that many real-world datasets are relational[1] in nature, as they contain information not only about entities and their attributes, but also *relationships* among them as well as *attributes* associated with the relationships. RelDC provides a principled domain-independent methodology to exploit these relationships for disambiguation, significantly improving data quality.

RelDC works by representing and analyzing each dataset in the form of entity-relationship graph. In this graph, entities are represented as nodes and edges correspond to relation-

[1] We use the standard definition of relational datasets as used in the database literature.

ships among entities. The graph is augmented further to represent ambiguity in data. The augmented graph is then analyzed to discover interconnections, including indirect and long connections, between entities which are then used to make disambiguation decisions to distinguish between same/similar representations of different entities as well as to learn different representations of the same entity. RelDC is based on a simple principle that entities tend to cluster and form multiple relationships among themselves.

After the construction of entity-relationship graphs, the algorithm computes the *connection strengths* between each uncertain reference and each of the reference's potential "options" – entities it could refer to. For instance, reference 'J. Smith' might have two options: 'John Smith' and 'Jane Smith'. The reference will be resolved to the option that has the strongest combination of the connection strength and the traditional feature-based similarity. Logically, the computation of the connection strength can be divided into two parts: first finding the connections which correspond to paths in the graph, and then measuring the strength in the discovered connections. In general, many connections between a pair of nodes may exist. For efficiency, only the important paths are considered, e.g., L-short simple paths. The strength of the discovered connections is measured by employing one of the connection strength models [46]. For instance, one model computes the connection strength of a path as the probability of following the path in the graph via a random walk.

After the connection strength is computed, this problem is transformed into an optimization problem of determining the weights between each reference and each of reference's option nodes. Once the weights are computed by solving the optimization problem, RelDC resolves the ambiguous reference to the option with the largest weight. Finally, the outcome of the disambiguation is used to create a regular (cleaned) database.

2.2 Person Identification

In the prior literature, lots of prior work has explored the issues of face recognition/clustering. The most conventional approaches are merely based on facial features, which are very sensitive to the variations of image captured conditions. To deal with this problem, researchers shift their research interests to context assisted methods. In the following, we will summarize the prior related works about these two types of approaches, and then briefly introduce the face tagging problem with active learning.

2.2.1 Video Based Person Identification

The conventional approaches for *person identification* are to first use face detection followed by face recognition. Figure 2.1 illustrates the basic schema for person identification. Given a face frame, after locating faces via a face detector, the extracted faces are passed to a matcher which leverages the face recognition techniques to measure the similarities between the extracted faces and "gallery faces" (where true identities of people are known) in order to determine the identities of the extracted faces.

In general, face detection is the first and essential component in person identification. In the prior literature, hundreds of approaches to face detection have been proposed. According to the survey paper [91, 94], early works (before year 2000) can be grouped into four categories: knowledge-based methods [89], feature invariant approaches [93], template matching methods [53], and appearance-based methods [60, 74]. Significant progress has been made in the past decade. Particularly, the work by Viola and Jones [81], with opensource implementation in the OpenCV library, has made face detection practically feasible in real-world applications, such as digital cameras and photo organization software. Recently, Zhu and Ramanan [103] have proposed a unified model based on a mixtures of trees for face detection,

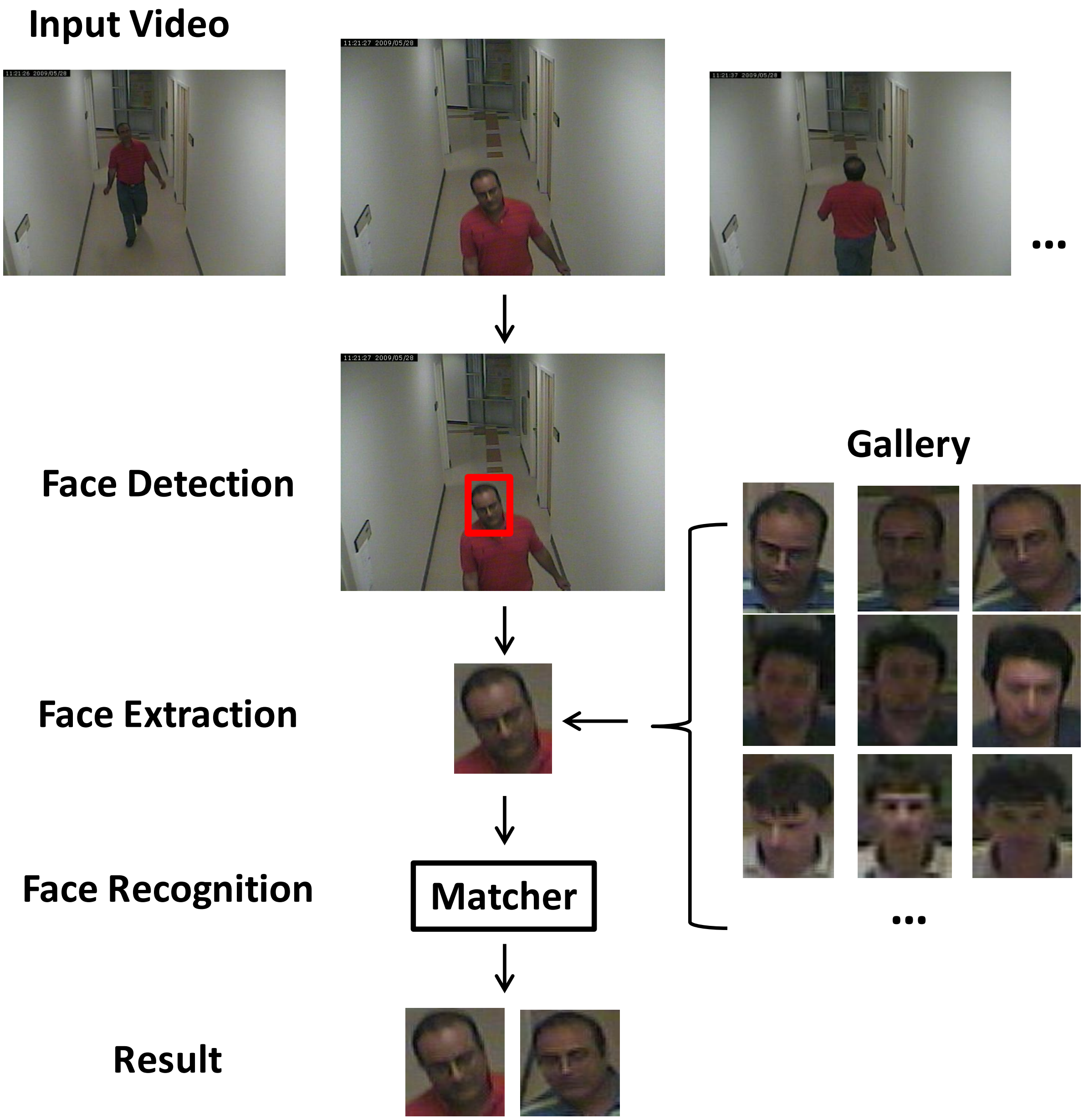


Figure 2.1: Example of Basic Person Identification Process

pose estimation, and landmark estimation in real-world, cluttered images.

Face recognition is another active topic of research that has attracted significant attention in the last two decades. Most of the research efforts have focused on techniques for still images, especially face representation methods. Recently, descriptor-based face representation approaches have been proposed and proven to be effective. They include Local Binary Pattern (LBP) [4] describing the micro-structure of faces, SIFT and Histogram of Oriented Gradients (HOG) [28], and so on. These face recognition techniques are able to achieve good performance in controlled situations, but tend to suffer when dealing with uncontrolled conditions where faces are captured with a large variation in pose, illumination, expression, scale, motion blur, occlusion, etc. Thus leveraging context features could bring significant improvement on top of techniques that rely on low-level visual features only, especially in the context of surveillance videos.

Compared to still images, videos often have more useful features and additional context information that can aid in face recognition. For example, a video sequence would often contain several images of the same entity, which potentially shows the entity's appearance under different conditions. Surveillance videos usually have temporal and spatial information available, which still images do not always have. In addition, video frames are capable of storing the objects in different angles, which contain 3-D geometric information. To better leverage these properties, some face recognition algorithms have been proposed to operate on video data. They include using temporal voting to improve the identification rates, extracting 2-D or 3-D face structures from video sequences [35][36][37]. However, these methods do not fully exploit the context information and very few of them address the problem of integration of heterogeneous context features.

Therefore, we propose to leverage heterogeneous contextual information to improve the performance of video-based face recognition. To integrate the heterogeneous contextual features together, we connect the problem of *person identification* with the well studied *entity resolu-*

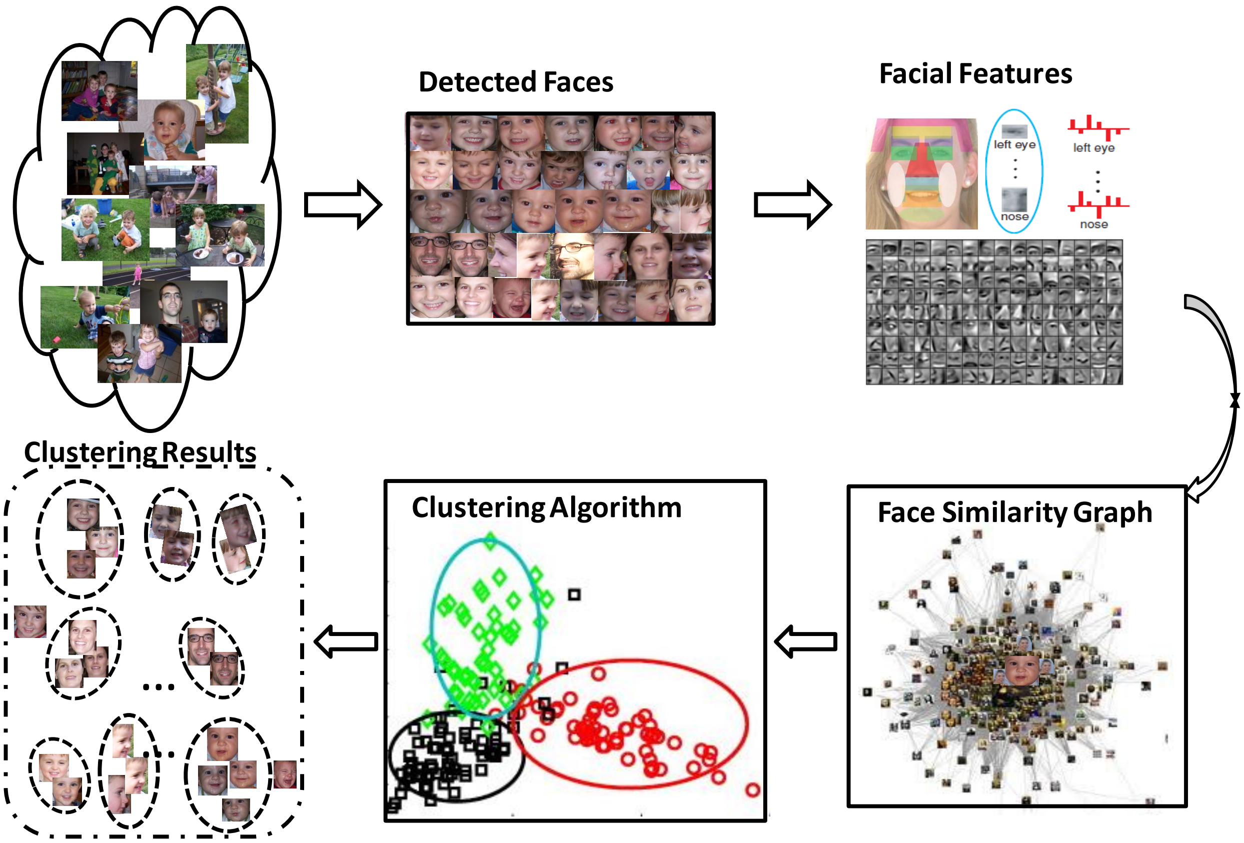


Figure 2.2: Conventional Framework for Face Clustering

tion problem, and apply our entity resolution RelDC framework to construct a relationship graph to resolve the corresponding person identification problem.

2.2.2 Facial Appearance based Face Clustering Approaches

As illustrated in Figure 2.2, facial appearance based methods are the most conventional strategies to handle face clustering/recognition. Given a photo collection, after the process of face detection and alignment, different types of low-level facial features can be extracted to represent the detected faces, including Local Binary Pattern (LBP) [4], Histogram of Oriented Gradients (HOG) [28], and Local Phase Quantization (LPQ), etc. Following that, the extracted high-dimensional features will be projected into low-dimensional spaces leveraging dimensionality reduction techniques such as Principle Component Analysis (PCA) and Lin-

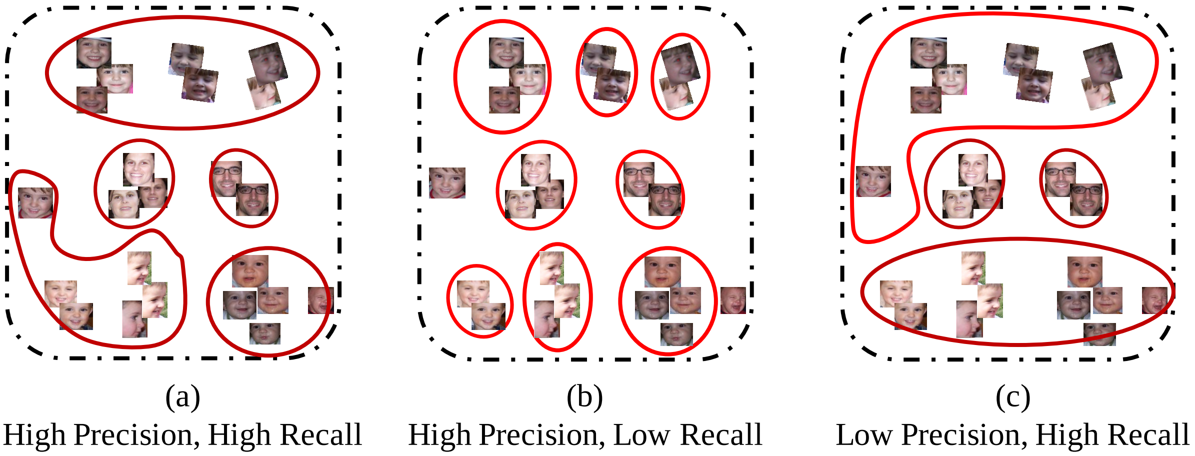
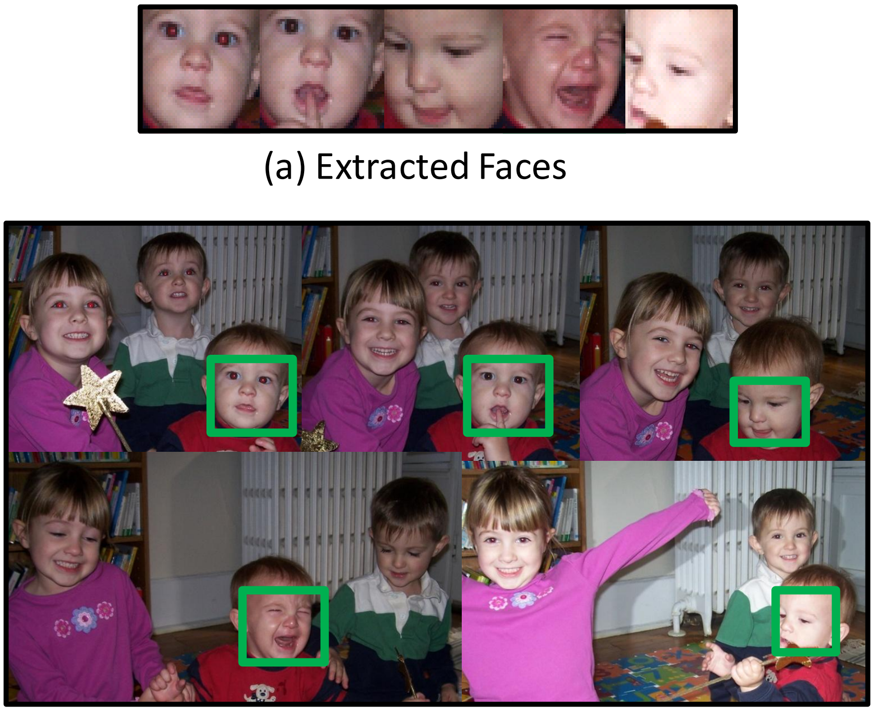

(a)
High Precision, High Recall
(b)
High Precision, Low Recall
(c)
Low Precision, High Recall

Figure 2.3: Clustering Results by selecting different thresholds. (a)The ideal clustering results. (b) Tight threshold leads to high precision but low recall results. (c) Loose threshold leads to low precision and high recall results.

ear Discriminant Analysis (LDA). Then the low-dimensional discriminant features can be used to compute face similarities, and further construct face groups by performing clustering algorithms such as K-Means, Affinity Propagation [31], etc.

These traditional approaches are able to achieve good performance under controlled conditions, but they tend to suffer when dealing with uncontrolled situations where faces are captured with a large of variation in pose, illumination, motion blur, occlusion, expression, scale, etc. These nuisance factors may cause the visual differences between faces captured in distinct conditions referring to the same entity to be greater than those between two different entities under similar conditions. Therefore, the techniques that rely on low level facial features only are not sufficient to handle the issues and achieve the very high-quality clustering performance. Just as demonstrated in Figure 2.3, the tight threshold will lead to results with high precision but low recall, while the loose threshold will lead to high recall but low precision results. In consequence, researchers shift their attentions to context assisted methods.



(a) Extracted Faces

(b) The Whole Images

Figure 2.4: Example of Face Clusters by Picasa

2.2.3 Context Assisted Approaches

The utilization of context information could bring significant improvement on top of the techniques that rely on facial features only. As illustrated in Figure 2.4, it is even challenging for a human to determine whether the five faces (in Figure 2.4(a)) describe the same entity. However, considering the entire collection of whole images which contain various of context information, it becomes obvious that the five faces refer to the same baby.

In the prior literature, many types of context information have been investigated as additional clues to facial features for face recognition/clustering, such as Geo-location and image captured time[101], people co-occurrence[73][86][54], social norm and conventional positioning observed[34], human attributes [52], text or other linked information[87][12], clothing [33][100], etc. In [52], Kumar et al. proposed that human attributes can be employed as an additional information to improve face verification performance, however, they did not consider the different roles that each attribute type plays in identifying different people. Social context such as people co-occurrence, has been investigated in [73][86][54] , but none of these works propose an appropriate way to leverage people co-occurrence information in cluster level. Clothing information has attracted much attention in face clustering [33][100]. However, these works did not introduce time decay factor in leveraging clothing information.

2.2.4 Face Tagging with Active Learning

Prior research has extensively explored the topic of face tagging and significant progress has been achieved in recent years [27, 51, 79]. Besides the design of user interface [27], one of the key problems is choosing which faces to tag in order to maximize the performance of the entire data. This problem can be resolved by the active learning framework. For example, Kapoor et al. [51] propose a gaussian process based active learning paradigm which incorporates constraints as a prior to guide users to tag faces. Tian et al. [79] perform partial clustering and assume each cluster contains a single entity to facilitate face tagging. In the field of computer vision, active learning has been widely employed in a large variaty of applications, including object categorization [50], video annotation, and face tagging [51]. Generally, the conventional approaches for face tagging and active learning aim to choose the unlabeled samples which can maximally reduce the uncertainty of the whole data set. However, this strategy can not be applied to our problem since our target is to answer users queries rather than reduce uncertainties of the whole data set.

2.2.5 Limitations of Existing Methods

Although advances in computer vision and multimedia domain has resulted in a series of powerful techniques for face detection, face recognition/clustering, and face tagging, such techniques are not sufficient to address the challenges in the real-world applications. For instance, it is very difficult to detect faces from low-quality images/videos. Besides, the large variation of face poses, expression, illumination, occlusions, makes the task of face recognition/clustering more complex and challenging. Although context information has been recognized to be useful, most of existing context-assist techniques only explore one (or a few) contextual feature types, with decision made at the image level. How to systematically integrate heterogeneous contextual information to improve the disambiguation performance is still a challenge.

Chapter 3

Context-based Person Identification Framework for Video Surveillance

In this chapter, we study *the person identification problem in the context of smart video surveillance.*

Advances in sensing, networking, and computational technologies has allowed the possibility of creating sentient pervasive spaces wherein sensors embedded in physical environments are used to monitor its evolving state to improve the quality of our lives. There are numerous physical world domains in which sensors are used to enable new functionalities and/or bring new efficiencies including intelligent transportation systems, reconnaissance, surveillance systems, smart buildings, smart grid, and so on.

In this chapter, we focus on Smart Video Surveillance (SVS) systems wherein video cameras are installed within buildings to monitor human activities [41, 42, 78]. Surveillance system could support variety of tasks: from building security to new applications such as locating/-tracking people, inventory, or tasks like analysis of human activity in shared spaces (such as offices) to bring improvements on how the building is used. One of the key challenges in building smart surveillance systems is that of automatically extracting semantic information from the video streams [75, 76, 83, 84]. This semantic information may correspond to human activities, events of interest, and so on that can then be used to create a representation of the state of the physical world, e.g., a building. This semantic representation, when stored inside a sufficiently powerful spatio-temporal database, can be used to build variety of monitoring and/or analysis applications. Most of the current work in this direction focuses on computer vision techniques. Automatic detection of events from surveillance videos is a difficult challenge and the performance of current techniques, often leaves a room for improvement. While event detection consists of multiple challenges, (e.g., activity detection, location determination, and so on), in this chapter we focus on a particularly challenging task of *person identification*.

The challenge of *person identification* (PI) consists of associating each subject that occurs in the video with a real-world person it corresponds to. In the domain of computer vision, the most direct way to identify a person is to perform *face detection* followed by *face recognition*, the accuracy of which is limited even when video data is of high quality, due to the large variation of illumination, pose, expression and occlusion, etc. Thus, in the resource constrained environments, where transmission delay, bandwidth restriction, and packet loss may prevent the capture of high quality data, face detection and recognition becomes more complex. We have experimented with Picasa's face detector on our video dataset[1], and found that it can detect faces in only 7% of the cases and then among them it can recognize only 4% of faces.

Figure 3.1 illustrates the example of frames in our video dataset, where only one face is successfully detected (solid-line rectangle) utilizing the current face detection techniques. Several reasons account for the low detection rate: (1) faces can not be captured if people walk with their back to the cameras; (2) faces are too small to be detected when people are far away from cameras; (3) the large variation of people's pose and expression brings more

[1] 704×480 resolution per frame

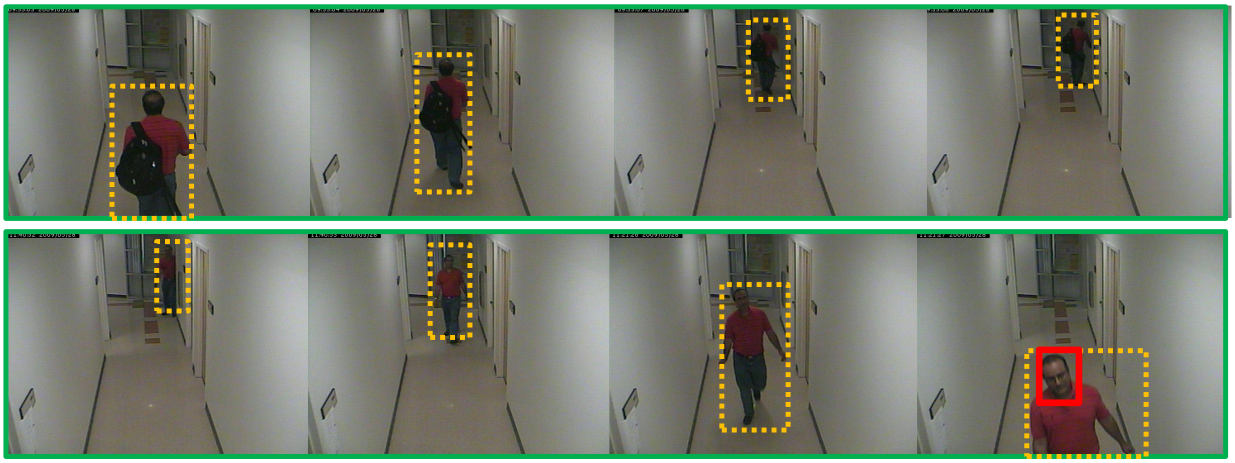


Figure 3.1: Example of Surveillance Video Frames

challenges to face detection. Thus the traditional face detection and recognition techniques are not sufficient to handle the poor quality surveillance video data.

To deal with the poor quality video data and overcome the limitation of current face detection techniques, we shift our research focus to context- based approaches. Contextual data such as time, space, clothing, people co-occurrence, gait and activities, is able to provide the additional cues for person identification. Consider the frames illustrated in Figure 3.1 for example. Although only one face is detected and no recognition results are provided, the identities of all the subjects can be estimated by analyzing the contextual information. First, time and foreground color continuities split the eight frames into two sequences or shots. The first four frames construct the first shot, and the following four frames form the second shot, where subjects within each shot describe the same entities. Furthermore, some other contextual features reveal the high possibility that the two subjects in these shots are the same entity. For instance, they both share the similar clothing (red T-shirt and gray pants), they perform similar activities (walking in front of the same camera, though in opposite directions), and they have the same gaits (walking speed). Thus the context features help to reveal that the subjects in the eight frames very likely refer to the same entity. To identify this person, face recognition process is usually inevitable. However, the activity information can also provide extra cues to recognize people's identity. In the above example, suppose that the first shot in Figure 3.1 is the first shot of that day where a person enters the corner office which belongs to "Bob". Then most probably this person is "Bob" because in most cases, the first person entering the office should have the key. Therefore, by analyzing contextual information even without face recognition results, we can predict that the very likely identity of subject in all the eight frames is "Bob". The example demonstrates the essential role that contextual data plays in the *person identification* issue for the low quality video data. Another significant advantage of context information is its weaker sensitivity to video data quality as compared that of face recognition. That makes context-driven approaches more robust and reliable when dealing with poor quality data.

In this chapter, based on our previous work[98], we explore a novel approach to leverage contextual information, including time, space, clothing, people co-occurrence, gait and activities to improve the performance of *person identification*. To exploit contextual information, we connect the problem of person identification with a well-studied problem of *entity resolution* [46, 16, 58], which typically deals with textual data. *Entity resolution* is a very active research area where many powerful and generic approaches have been proposed, some of which could potentially be applied to the person identification problem. In this chapter, we first investigate methods for extracting and processing several different types of context features. We then demonstrate how to apply a relationship-based approach for entity resolution, called RelDC [48], to the person identification problem. RelDC is an algorithmic framework for analyzing object features as well as inter-object relationships, to improve the quality of entity resolution. In this chapter, we will demonstrate how RelDC framework for *entity resolution* could be leveraged to solve a *person identification* problem that arises when analyzing video streams produced by cameras installed in the CS Department at UC Irvine. Our empirical evaluation demonstrates the advantage of the context-based solution over the traditional techniques, as well as its effectiveness and robustness. The proposed approach shows clear improvements over approaches that only exploit facial features. The

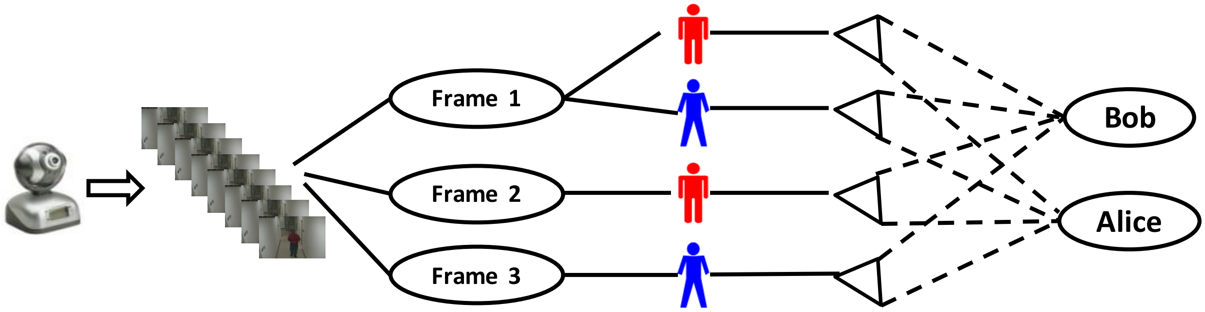


Figure 3.2: Example of Person Identification for Surveillance Videos

improvement is even more pronounced for low quality data, as it relies on contextual features that are less sensitive to deterioration of data quality.

The rest of this chapter is organized as follows. We first formally define the problem in Section 3.1. In Section 3.2, we present the proposed approach for context based person identification. Section 3.3 demonstrates experiments and results. Finally, we conclude in Section 3.4 by highlighting key points of our work.

3.1 Problem Definition

Let $\mathcal{D}$ be the surveillance video dataset. The dataset contains K video frames $F = \{f_1, f_2, \ldots, f_K\}$ wherein motion has been detected. Let t_i denote the time stamp of each frame f_i. When a frame f_i contains just one subject, we will refer to the subject as x_i, or as x^{f_i}. Let $P = \{p_1, p_2, \ldots, p_{|P|}\}$ be the set of (known) people of interest that appear in our dataset. Then the goal of person identification is for each subject x_i to compute w_{ij} which denotes the probability that x_i is person p_j, and correctly identify the person $p_k \in P$ that subject x_i corresponds to. If the subject is not in P, then the algorithm should output $x_i = \texttt{other}$. Table 3.1 summarizes some of the notations throughout this chapter.

Figure 3.2 illustrates an example of the person identification problem, where the goal is

Table 3.1: Notation and Description

Notation	Meaning
$\mathcal{D}$	The surveillance video dataset being processed
$F = \{f_1, f_2, \dots, f_K\}$	The set of detected motioned video frames
$S = \{s_1, s_2, \dots, s_{\lvert S \rvert}\}$	The set of shots after video segmentation
$X = \{x_1, x_2, \dots, x_{\lvert X \rvert}\}$	The set of subjects appearing in the video
$P = \{p_1, p_2, \dots, p_{\lvert P \rvert}\}$	The set of real-world people of interest
w_{ij}	The probability that subject x_i is person p_j
$S^C(x_i, x_j)$	The cloth similarity between subjects x_i and x_j
$S_{ij}^{act}(act_i^m, act_j^n)$	The similarity between activity act_i^m and act_j^n
$\mathbb{P}(x_i = p_j \mid act_i, t_k)$	The probability that subject x_i is person p_j based on activity and time
A_i	The attribute vector for subject x_i
$FR(x_i, p_j)$	The probability that subject x_i is person p_j based on facial features
$G = (V, E)$	The entity relationship graph
$cs(x_i, p_j)$	The connection strength measure between subject x_i and person p_j

to determine whom the subject in each video frame refers to: "Bob" or "Alice". We can observe that the entity resolution problem has a very similar goal, that is, to associate each uncertain reference to an object in the database with the real-world object. Hence in this chapter we demonstrate how to apply one entity-resolution framework called RelDC to the problem of person identification. The framework will exploit the relationships between contextual features of subjects in the video surveillance to improve the quality of the person identification task.

3.2 Context based Framework for Person Identification

Figure 3.3 illustrates the general framework of context-based person identification for surveillance videos. Given the stored videos from surveillance cameras, the framework first segments

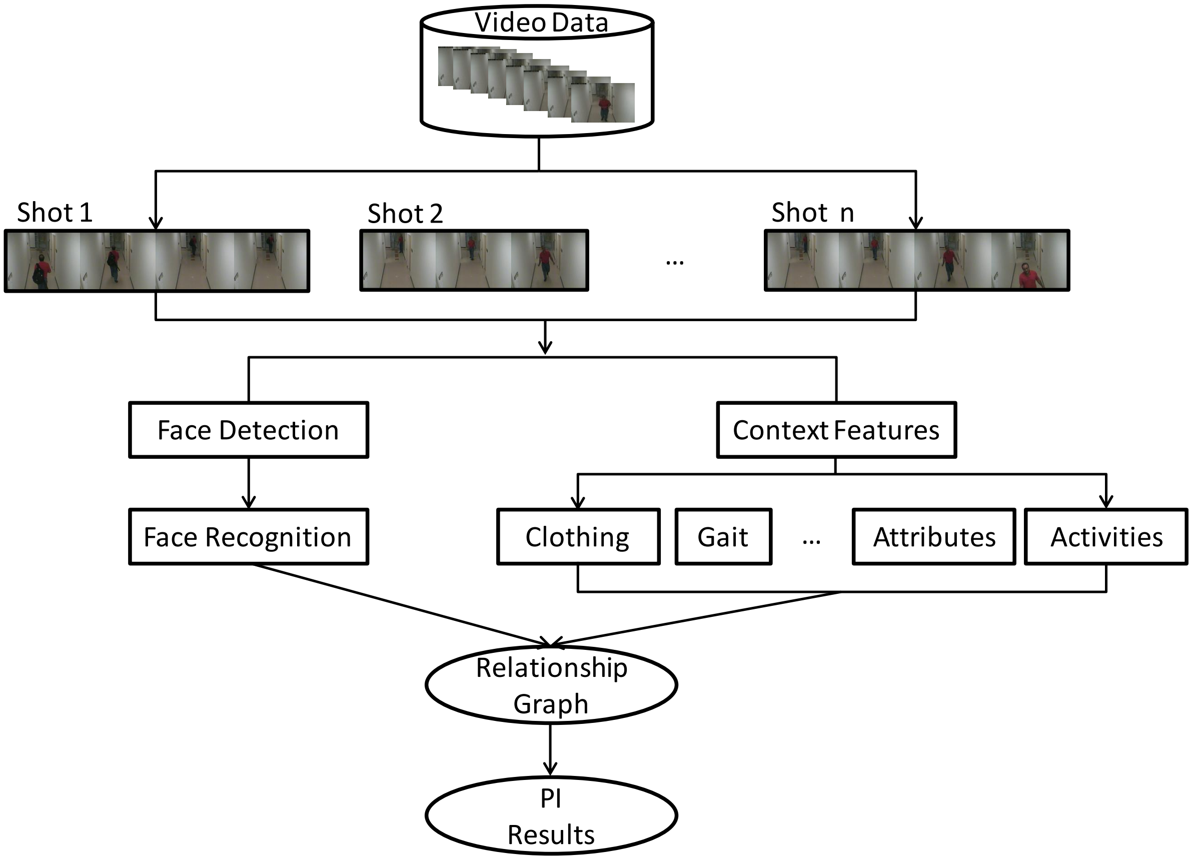


Figure 3.3: General Framework for Context-based Approach

the frames with motion into shots based on temporal information. To facilitate person i-dentification based on person faces the framework performs several preliminary steps such as face detection, extraction, facial representation and recognition. It then extracts the contextual features including people's clothing, attributes, gait, activities, etc. After the extraction of face and contextual features, the framework constructs the entity-relationship graph and then applies the entity resolution algorithm RelDC on the graph to perform the corresponding person identification task.

In the following we discuss how to extract contextual features from surveillance videos, and then leverage RelDC framework to integrate these features together to resolve the person identification problem.

3.2.1 Contextual Feature Extraction

Contextual features can provide additional cues to facilitate video-based person identification, especially for poor quality video data. In the following, we describe how to extract and leverage contextual features, such as people's clothing, attribute, gait, activities, co-occurrence, and so on, to improve the performance of person identification.

Temporal Segmentation

We first describe temporal segmentation which is an essential part in video processing. We segment videos into *shots*. Intuitively, subjects appearing in consecutive frames are likely to be the same person. Hence, we initially group frames into shots just based on the time continuity. But time continuity alone can not guarantee person continuity. If the subjects' color histograms of two consecutive frames are significantly different indicating potentially different people, the shot is split further at such break points.

Suppose that we obtain a set of shots $S = \{s_1, s_2, \ldots, s_{|S|}\}$ after the video segmentation. Most of the time the frames that belong to the same shot describe the same entities. Thus the person identification task reduces from identifying the subjects in an image to identifying the subjects in a shot. We next describe how to extract contextual features for a shot.

Clothing

People's clothing can be a good discriminative feature for distinguishing among people [33][90]. Although people change their clothes across different days, they do not change it too often within shorter period of time, and hence the same clothing in such cases is often strong evidence that two images contain the same person. To accurately capture the clothing information of an individual in an image, we separate the person from the background

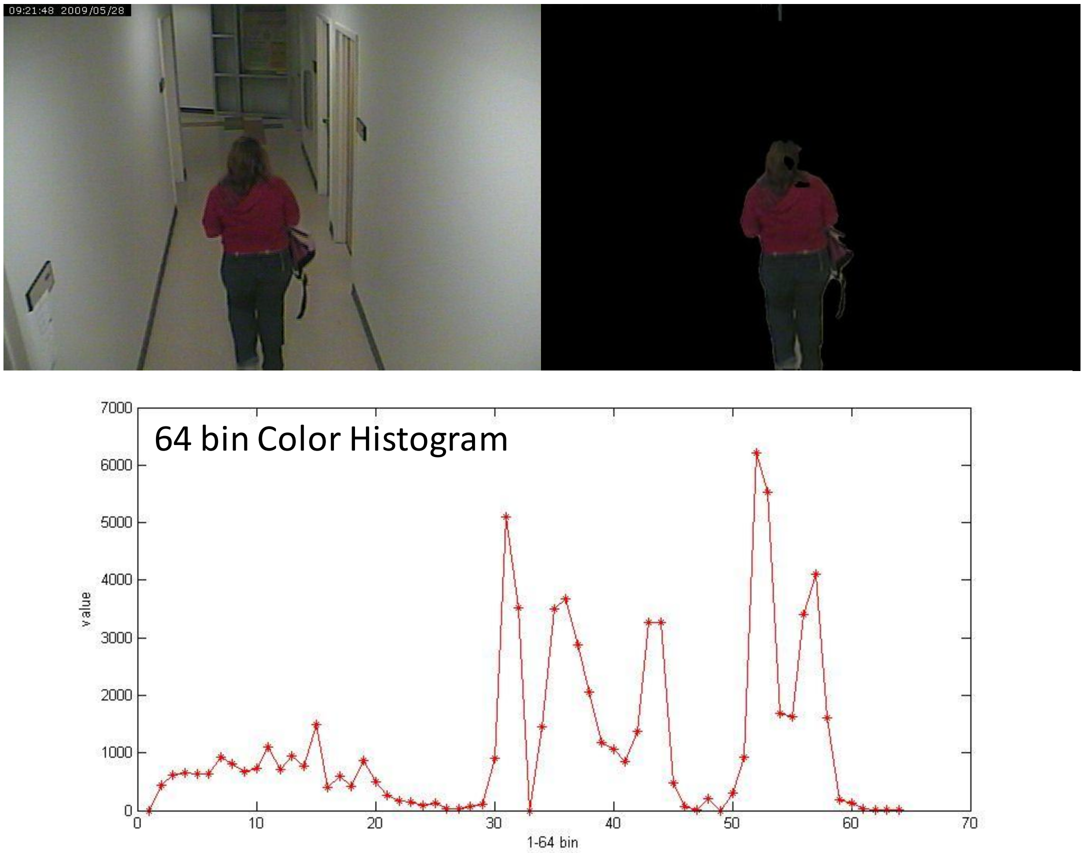


Figure 3.4: Example of Foreground Extraction

by applying a background subtraction algorithm[11]. After color extraction processing, the foreground area is represented by a 64-dimensional vector, which consists of a 32-bin hue histogram, a 16-bin saturation histogram, and a 16-bin brightness histogram. Figure 3.4 shows an example of the extracted foreground image and corresponding color histogram.

The extracted clothing features can be used to compute the clothing-based similarity among subjects. For each pair of subjects x_i and x_j, let C_i and C_j be their clothing histograms and t_i and t_j by the timestamps when x_i and x_j have been captured in video. We can choose an appropriate similarity measure to compute the similarities between them, such as the cosine similarity. For instance, if we assume that people keep the same clothing during the same day, we can define

$$S^C(x_i, x_j) = \begin{cases} \frac{C_i \cdot C_j}{|C_i||C_j|} & \text{if } day(t_i) = day(t_j) \\ 0 & \text{otherwise} \end{cases}$$

To compute the similarity of subjects from two shots, the algorithm selects a subject from a certain frame in a shot to represent the shot. Usually the algorithm chooses a frame towards the middle, which tends to capture the profile of the person better.

Activity

Activities and events associated with subjects prove to be very relevant to the problem of person identification [26][102]. The trajectory and walking direction can serve as a cue indicating the identity of the individual. For example, the activity of entering an office can provide strong evidence about the identity of the subject entering the office: it is likely to be either (one of the) person(s) who works in this office, or their collaborators and friends. Furthermore, considering the time of the activity in addition to the activity itself can often provide even better disambiguation power. For example, on any given weekday, the person who enters an office first on that day is likely to be the owner of the office. In addition, by analyzing past video data, the behavior routines for different people can be extracted, which later can provide clues about the identify of subjects in video. For instance, if we discover that "Bob" is accustomed to entering the coffee room to drink his coffee at about 10 a.m. each weekday, then the subject who enters the coffee room at around 10 a.m. is possibly "Bob". Therefore, subject activities can often provide additional evidence to recognize people. We now discuss how to extract and analyze certain people's activities.

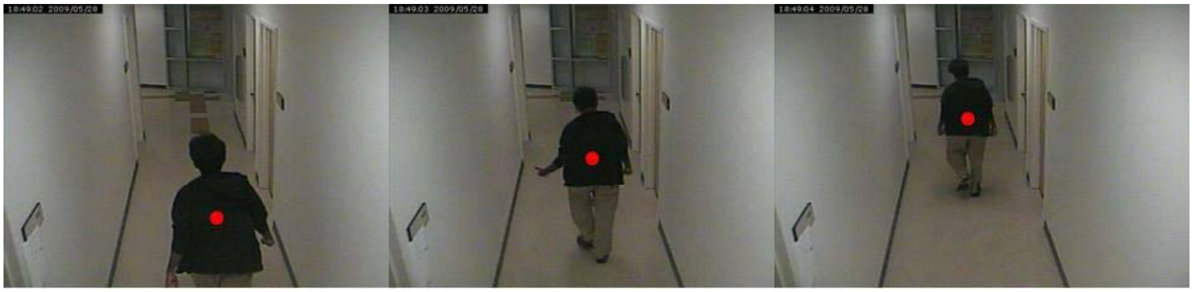


Figure 3.5: Example of Walking Direction.

Bounding Box and Centroid Extraction. To track the trajectory of a subject and obtain his activity information, we need to extract bounding box and centroid of the subject. To do that we consider three consecutive frames with the same object. We first compute the differences of the first two frames by subtraction, and then compute the differences of the last two frames. By combining the two different parts, we get the location of objects. After obtaining the bounding box, we determine the centroid of subjects by averaging the points of x-axes and y-axes.

Walking Direction. The most common activity in surveillance dataset is walking. The walking direction (towards or away from the camera) is an important factor to predict the subsequent behavior of a person. The walking direction can be obtained automatically by analyzing the changes of the centroid between two consecutive frames in a shot. For example, as illustrated in Figure 3.5, by determining that the centroid of the subject is moving from the bottom to the top in the camera view, we can determine that this person is walking away from the camera.

Activity Detection. We focus on detecting simple regular type of behavior of people, including entering and exiting a room, walking through the corridor, standing still, and so on. These types of behavior can be determined by analyzing the bounding box of a person. For instance, for walking the algorithm focuses on the first and last frame in a shot, which

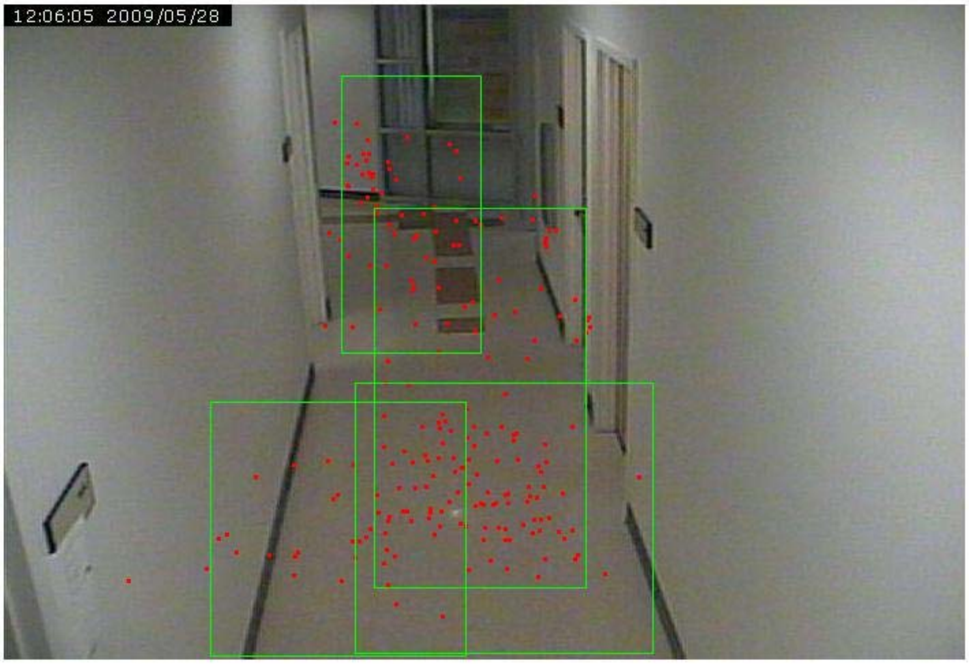


Figure 3.6: Example for Location Clustering.

we are called *entrance* and *exit* frames. By analyzing the bounding box (BB) of a subject in the entrance frame, we could predict where the subject has come from. Similarly, the exit frame could tell us where this person is headed to.

If we consider all the BBs in entrance and exit frames, we can find several locations in the camera view, where people are most likely to appear or disappear. These locations, denoted as $L = \{l_1, l_2, \ldots, l_{|L|}\}$, can be automatically computed in an unsupervised way by clustering the centroid of entrance/exit BBs. Based on this analysis, we automatically obtain the entrance and exit point in an image. Figure 3.6 demonstrates an example of the clustering result of the entrance and exit locations.

After computing the set of entrance and exit locations $L = \{l_1, l_2, \ldots, l_{|L|}\}$, we compute the distance between them and determine the entrance and exit points in each shot. Suppose that in a shot s_m the subject x_i walks from location l_p to l_q. Then we can denote the activity as $act_i^m : \{l_p \rightarrow l_q\}$.

Activity Similarity. For each shot the algorithm extracts the activity information by performing the aforementioned process. We assume that two subjects with similar activities have a certain possibility to describe the same person. Thus based on this assumption, we connect the potentially same subjects through the similar activities. Suppose that for two subject x_i and x_j from shot s_m and shot s_n respectively, the algorithm extract activity information $act_i^m : \{l_a \rightarrow l_b\}$ and $act_j^n : \{l_c \rightarrow l_d\}$. We can define the activity similarity as follows.

$$S_{ij}^{act}(act_i^m, act_j^n) = \begin{cases} 1 & \text{if } l_a = l_c(l_d) \text{ and } l_b = l_d(l_c) \\ 0.5 & \text{if } l_a = l_c(l_d) \ \text{ or } \ l_b = l_d(l_c) \end{cases}$$

In this equation, activities with the exact opposite entrance/exit points are defined to be equal, for example, the subject x_i with activity $act_i : \{l_a \rightarrow l_b\}$ and the subject x_j with activity $act_j : \{l_b \rightarrow l_a\}$ are considered to share the same activity. Thus the activity similarities can be leveraged to connect the subjects which share the same/similar activities.

Person Estimation Based on Activity. The intuition is that the identity of a person can be estimated by analyzing his activities. In general, given labeled past data we can compute priors such as $\mathbb{P}(x_i = p_m | act_i)$, which corresponds to the probability that the observed subject x_i is the real-world person p_m, given that the subject participates in activity act_i, such as entering/exiting a certain location. Similarly, we can compute $\mathbb{P}(x_i = p_m | act_i, t_k)$ which also considers time.

Person Gait

Gait is also a good feature to identify a particular person, because different people's gaits are often different. For example, somebody might walk very fast or slow, somebody might walk with swinging arms or head. Thus by analyzing the characteristics of people's gaits, we might be able to better predict the identity of one subject or the sameness of two subjects. For example, if the walking speed of two subjects differs significantly, then they might not refer to the same entity.

Face-derived Human Attributes

Face-derived human attributes that could be estimated by analyzing people faces, such as gender, age, ethnicity, facial traits, and so on, are important evidence to identify a person. By considering these attributes, many uncertainties and errors for person identification can be avoided, such as confusing a "men" with a "women", an "adult" with a "child", and so on. To obtain attribute values from a given face, we use the attribute system[52]. It contains 73 types of attributes classifiers, such as "black hair", "big nose", or "wearing eyeglasses". Thus for each subject x_i, the algorithm computes 73-D attribute vector, denoted as A_i. The attribute similarity of two subjects x_i and x_j can be measured as the cosine similarity between A_i and A_j. In addition, if the extracted attribute for x_i is significantly different from that of the real-world person p_m, then x_i is not likely to be p_m.

However, the extraction of reliable attribute values depends on the quality of video data. This limitation usually leads to the failure of attribute extraction on lower quality data.

People Co-occurrence

To recognize the identity of a person, people that frequently co-occur/present with that person in the same frames can provide vital evidence. For example, suppose that “Bob” and “Alice” are good friends and usually walk together, then the identity of one person might imply that of the other. Thus given the labeled past video data, we can statistically analyze the people co-occurrence information, and compute the prior probability of one person in the presence of the other. Furthermore, from the co-occurrence/presence of two people in one frame we can derive that the two subjects are different people. This observation can help to differentiate subjects.

3.2.2 Face Detection and Recognition

Face detection and recognition is a direct way to identify a person. However, it does not perform well in our dataset due to several reasons. First, the surveillance cameras used are of low quality and also the resolution of each frame is not very high: 704×480. Second, people may actually walk away from cameras, in which case the cameras only capture their backs and not faces. Because of that, the best face detection algorithms we have tried could only detect faces in about 7% of frames, and recognize 1 or 2 faces for a frequently appearing person out of all of his/her images in the dataset. Although the result is not ideal, we could still leverage it for further processing. We define a function $FR(x_i, p_j)$ which reflects the result obtained by the face recognition. If x_i and p_j are the same according to face recognition, we set $FR(x_i, p_j) = 1$, and otherwise $FR(x_i, p_j) = 0$.

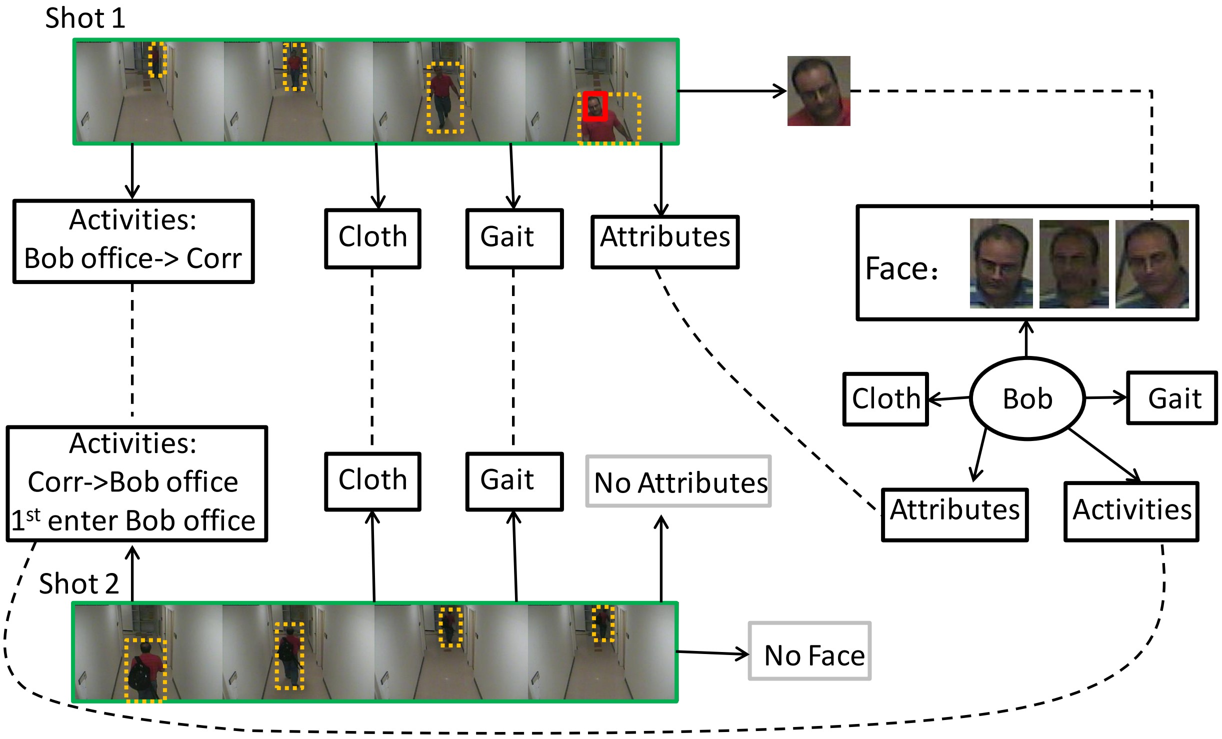


Figure 3.7: Example of Context-based Person Identificantion.

3.2.3 Solving the Person Identification Problem with RelDC

In the previous sections we have described how to extract contextual features including the people's clothing, face-derived attributes, gait, activities, co-occurrence, etc, and obtain the face recognition results. In this section we show how to represent the person identification problem as an entity resolution problem to be solved by our graph-based RelDC entity resolution framework.

RelDC performs entity resolution by analyzing object features as well as inter-object relationships to improve the data quality. To analyze relationships, RelDC leverages the entity-relationship graph of the dataset. The proposed framework will utilize inherent and contextual features, as well as the relationships, to improve the quality of person identification.

Figure 3.7 shows an example of the person identification process that employs both the inherent and contextual features. The simple person identification task in the example is to discover whether the subject in the given frames is “Bob” or someone else. The example shows that, by using face recognition, only one face (marked in the red rectangle) can be detected and recognized to be “Bob”, whereas the remaining subjects cannot be identified. On the other hand, by leveraging the context information, the identity of all the subjects can be recognized. Context information such as activity, clothing, gait, face-derived attributes can be extracted from the both the probe frames (the ones to be disambiguated) and gallery frames (the references frames where the labels/indentities are known). First, based on the time continuity, the frames are segmented into two shots, where in each shot the frames describe the same person. Thus, the four subjects in Shot 1 all refer to “Bob”. For Shot 2, although no face-based features can is computed (since the person is walking with his back towards the camera) , the subjects in Shot 2 can also be connected to “Bob” through contextual features. One such connection is the similar contextual features between Shot 2 and Shot 1 that we now know refers to “Bob”. Another connection is the special activity of Shot 2 which illustrates that the subject is the first person entering “Bob” offices on that day. Therefore, by constructing an entity-relationship graph which considers both inherent and contextual feature, the identity of subjects in all the probe frames can be resolved.

Entity-Relationship Graph

In order to apply RelDC, the algorithm first constructs an entity-relationship graph $G = (V, E)$ to represent the given person identification task, where V is the set of nodes and E is the set of edges. Each node corresponds to an entity and each edge to a relationship. The graph will contain several different *types* of nodes: *shot*, *subject*, *person*, *clothing*, *attribute*, *gait*, and *activity*. The edges linking these nodes correspond to the relationships. For instance, the edge between a shot node and a subject node correspond to the “appears in”

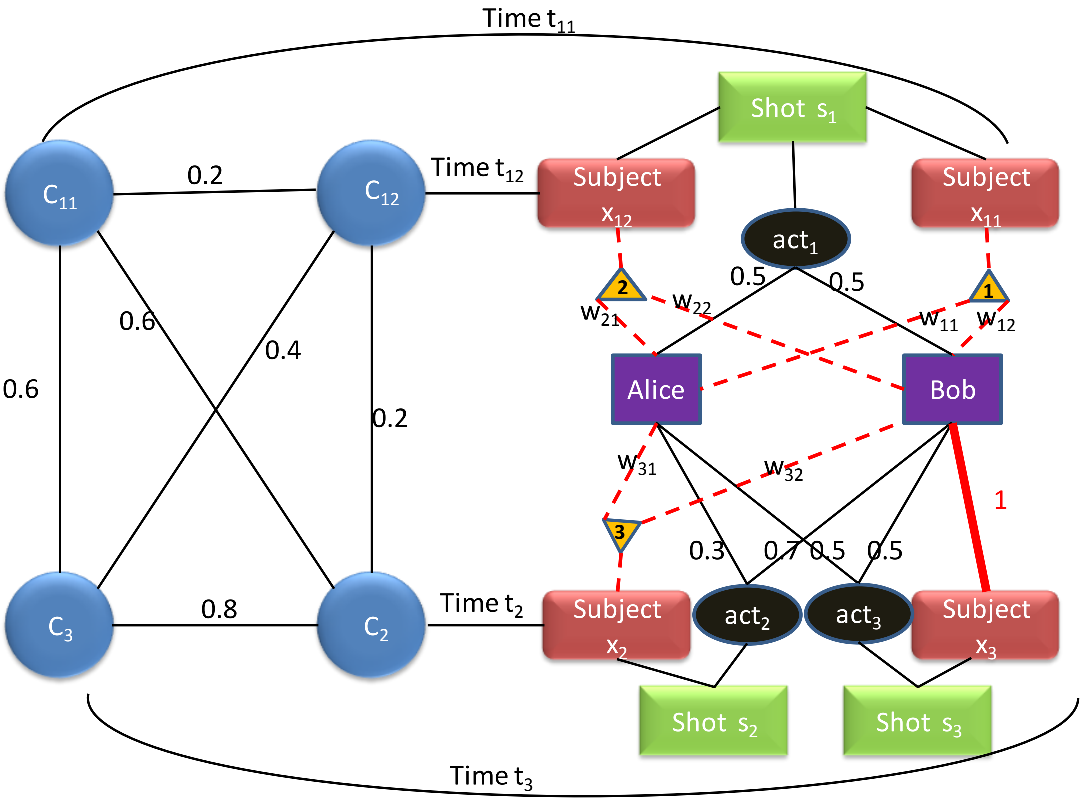


Figure 3.8: Example of Entity-Relationship Graph

relationship.

In graph G, edges have weights where a weight is a real number in [0,1] that reflects the degree of confidence in the relationship. For example, if there is an edge with weight 0.8 between a subject node and a person node, this implies the algorithm has 80% confidence that this subject and person are the same. The edge weight between two color histogram nodes denotes their similarity.

Figure 3.8 illustrates an example of an entity-relationship graph. It shows a case where the set of people of interest consists of just two persons: Alice and Bob. It considers three shots s_1, s_2, s_3, where s_1 captures two subjects x_{11} and x_{12}, shot s_2 captures x_2 and s_3 has x_3. The graph only shows the clothing and activity contextual features, the other contextual features are not shown for clarity. The goal is to match people with shots.

Subject x_{11}, x_{12}, x_2, x_3 in the graph are connected with their corresponding clothing color histograms C_{11}, C_{12}, C_2, C_3. An edge between two color histogram nodes represents the similarity between them. For instance, the similarity of C_2 and C_3 is 0.8. In addition, subjects are connected to the corresponding activities, which could be indicative of who these subjects are. For example, if the past labeled data is available, from the fact that subject s_3 is connected to activity act_3, we can get the prior probability of 0.7 that s_3 is Bob. The graph also shows that according to face recognition subject x_2 in shot s_2 is Bob.

The main goal is to analyze the relationships between the subject nodes and person nodes, and compute the weight w_{ij} that each subject x_i associating with person p_j. Notice, weights w_{ij} are the only variables in the graph whereas all other edge-weights are fixed constants. After constructing the graph, RelDC will compute the value of those w_{ij} weights based on the notion connection strength discussed next. After computing the weights, RelDC will use them to resolve each subject to the person that has the highest weight.

Connection Strength Computation

The constructed entity-relationship graph G illustrates the connections and linkages between subjects appearing in the video shots and real-world people. Intuitively, the more paths exist between two entities, the stronger the two entities are related. Thus we introduce the definition of connection strength $cs(x_l, p_j)$ between each subject node x_l and person node p_j, to reflect how strongly subject x_l and person p_j are related. The value of $cs(x_l, p_j)$ can be computed according to some connection strength model. The computation process logically consists of two parts: finding the connections (paths) between the two nodes and then measuring the strength in of the discovered connections.

Generally, many different paths can exist between two nodes, and considering very long paths could be inefficient. Therefore, in our approach, only important connection paths are taken into account, for instance, L-short simple paths (e.g., $L \leq 4$). For example, in Figure 3.8 one 4-short simple path between subject x_2 and person "Bob" is "x_2-C_2-C_3-x_3-Bob". We will use $P_L(x_l, p_j)$ to denote the set of all the L-short simple paths between subject node x_l and person node p_j.

To measure the strength of the discovered connections, some connection strength models [46] can be leveraged. For instance, we can compute the connection strength of a path p^a as the probability of following path p^a in graph G via random walks. The connection strength $cs(x_l, p_j)$ can be computed as the sum of the connection strengths of paths in $P_L(x_l, p_j)$.

$$cs(x_l, p_j) = \sum_{p^a \in P_L(x_l, p_j)} c(p^a). \tag{3.1}$$

Weight Computation

After computing the connection strength measures $cs(x_l, p_j)$ for each unresolved subject x_l and real-world person p_j, the next task is to determine the desired weight w_{lj} which should represent the confidence that subject x_l matches person p_j. RelDC computes these weights based on the the Context Attraction Principle (CAP) [46] that states that if $c_{r\ell} \geq c_{rj}$ then $w_{r\ell} \geq w_{rj}$, where $c_{r\ell} = c(x_r, p_\ell)$ and $c_{rj} = c(x_r, p_j)$. In other words, the higher weight should be assigned to the better connected person. Therefore, the weights are computed based on the connection strength. In particular, RelDC sets the weight proportional to the corresponding connection strengths: $w_{rj}c_{r\ell} = w_{r\ell}c_{rj}$. Using this strategy and given that $\sum_{j=1}^{N} w_{rj} = 1$ (if each possible "option node", that is, each possible person, are listed), the weight w_{rj}, for $j = 1, 2, \ldots, N$, can be computed as follows.

$$w_{rj} = \begin{cases} \frac{c_{rj}}{\sum_{j=1}^{N} c_{rj}} & \text{if } \sum_{j=1}^{N} c_{rj} > 0; \\ \frac{1}{N} & \text{if } \sum_{j=1}^{N} c_{rj} = 0. \end{cases} \tag{3.2}$$

Thus, since some paths can go through edges labeled with w_{ij} weight, the desired weight w_{rj} can be defined as a function of other option weights $\mathbf{w}$: $w_{rj} = f_{rj}(\mathbf{w})$.

$$\begin{cases} w_{rj} = f_{rj}(\mathbf{w}) & \text{(for all } r, j) \\ 0 \leq w_{rj} \leq 1 & \text{(for all } r, j) \end{cases} \tag{3.3}$$

The goal is to solve System (3.3). System (3.3) might not have a solution as it can be over-constrained. Thus, a slack is added to it by transforming each equation $w_{rj} = f_{rj}(\mathbf{w})$ into $f_{rj}(\mathbf{w}) - \xi_{rj} \leq w_{rj} \leq f_{rj}(\mathbf{w}) + \xi_{rj}$. Here, ξ_{rj} is a slack variable that can take on any real nonnegative value. The problem transforms into solving the optimization problem, where the objective is to minimize the sum of all ξ_{rj}:

$$\begin{cases} \text{Constraints:} & \\ f_{rj}(\mathbf{w}) - \xi_{rj} \leq w_{rj} \leq f_{rj}(\mathbf{w}) + \xi_{rj} & \text{(for all } r, j) \\ 0 \leq w_{rj} \leq 1 & \text{(for all } r, j) \\ 0 \leq \xi_{rj} & \text{(for all } r, j) \\ & \\ \text{Objective: Minimize} \sum_{r,j} \xi_{rj} & \end{cases} \tag{3.4}$$

System (3.4) always has a solution and it can be solved by a solver or iteratively. In our scenario, we solve this system in an iterative way [46]. The solution of this system are the values for all w_{rj} weights.

Interpretation Procedure

The computed weight w_{rj} reflects the algorithm's confidence that subject x_r is person p_j. The next task is to decide which person to assign to x_r given the weights. The original RelDC chooses the person p_j who has the largest weight w_{rj} among $w_{r1}, w_{r2}, \ldots, w_{r|P|}$, when resolving the references of subject x_r.

The original strategy is meant for the case where each possible person p_j that x_r can refer to is known beforehand. However, in the setting of the person identification problem, this is not the case, as the algorithm is trying to decide if x_r refers to one of the known people p_j of interest or to some "other" person. To handle this new "other" category, we modify the the original RelDC algorithm to also checks if all of the computed weights are above a certain predefined threshold t. If they are below the threshold, this means the algorithm does not have enough evidence to resolve subject x_r, in which case it assigns x_r to "other". Otherwise, it will pick the person with the largest weight – the same way as the original algorithm.

3.3 Experiments and Results

3.3.1 Experimental Datasets

Our experimental dataset consists of two weeks' surveillance videos from two adjacent cameras located in the second floor of CS Department building at UC Irvine [80]. These cameras are distributed in the corners of a corridor, near the offices of the Information System Group

(ISG) members. Activities of graduate students and faculty, such as entering and exiting offices, hallway conversations, walking, and so on, are captured by these cameras. Frames are collected continuously when motion is detected with the frame rate of 1 frame a second for each camera. The resulting video shots are relatively simple, with one (or, rarely, a few) person(s) performing simple activities. The task is to map the unknown subjects into known people.

To test the performance of the proposed algorithm, we manually labeled 4 people from the video dataset to assign the ground truth labels. The video collected over 2 weeks contains several (over 50) individuals of which we manually labeled 4. We then have divided the dataset into 2 parts. The first week has been used as training data and the second week as test data. From the training data, we get the faces of the chosen 4 people, and train a face recognizer. We also extract activities of people, and compute priors based on activities.

3.3.2 Evaluation Metrics

We have applied RelDC (in a limited form with a simplified connection strength model) to identify the four people from the testing dataset. After obtaining the weight w_{rj} for each subject x_r to person p_j, we decide which person that each subject should be assigned to using our strategy. The subject x_r can be assigned to p_j only if two requirements are satisfied: (1) w_{rj} is the largest among $w_{r1}, w_{r2}, \ldots, w_{r|P|}$, (2) $w_{rj} \geq threshold$. If the weights of a subject for each optional person are almost equal, and none of them is larger than the threshold, then this subject will be considered as "others". By setting different thresholds, we can get different recognition results.

To evaluate the performance of the proposed method, we choose precision and recall as the evaluation metrics. By selecting a particular threshold value, each subject x_r can be assigned a label denoted as $L(x_r)$. The ground truth of identity for each subject x_r is referred as $T(x_r)$.

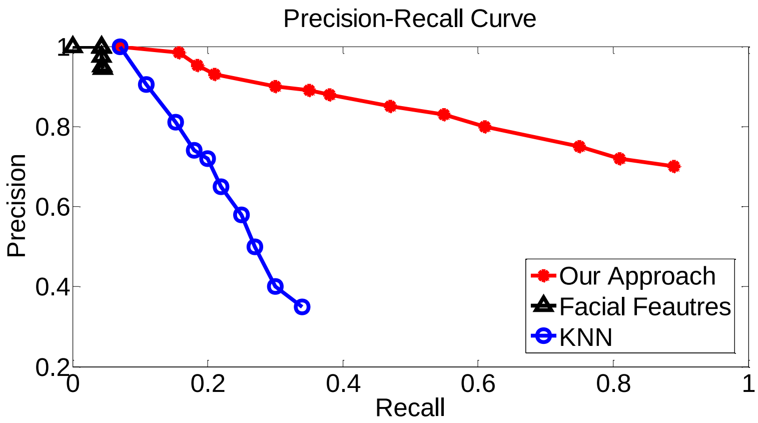


Figure 3.9: Precision-Recall Curve

Then as to each person p_j in the person set P, we can compute the corresponding precision and recall based on $L(x_r)$ and $T(x_r)$. Thus the total precision and recall can be obtained by averaging the precision and recall for each targeted person p_j.

$$Precision = \frac{1}{|P|}\sum_{j=1}^{|P|}\frac{|\{x_r|L(x_r) = p_j \wedge T(x_r) = p_j\}|}{|\{x_r|L(x_r) = p_j\}|} \tag{3.5}$$

$$Recall = \frac{1}{|P|}\sum_{j=1}^{|P|}\frac{|\{x_r|L(x_r) = p_j \wedge T(x_r) = p_j\}|}{|\{x_r|T(x_r) = p_j\}|} \tag{3.6}$$

3.3.3 Results

Figure 3.9 illustrates the precision-recall curve achieved by selecting different threshold values. We compare our approach with two conventional approaches.

- Facial features based method. As shown in Figure 3.9, if merely leveraging facial visual features, the performance of person identification is very poor. The recall is pretty low

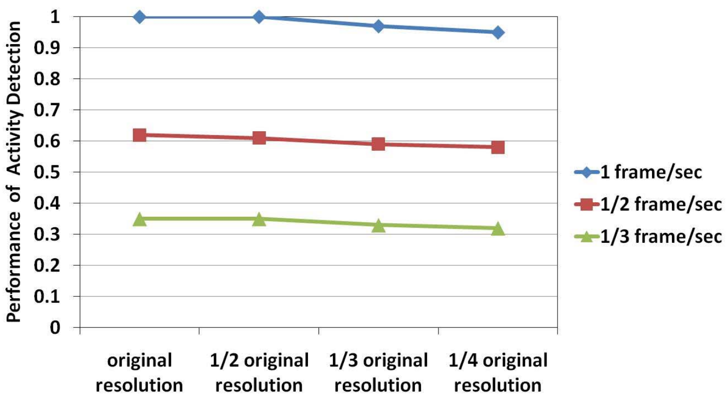


Figure 3.10: Activity Detection with Decreasing of Resolution and Sampling Rate

because most faces in the dataset can not be detected due to the low quality of data, and thus the following recognition process is not able to be performed.

- K nearest neighbors method (KNN). To perform KNN, we just simply aggregate all the heterogeneous context features to obtain the overall subject similarities, and then label the K nearest neighbors of the resolved subject with the same identity. By introducing context features, this method can achieve better performance than the facial features based method. However, in this method, the underlying relationships between different context features are not considered.

The comparison with the above two approaches demonstrates the superiority of our approach. The advantages of our approach lie in that we not only leverage heterogeneous context features, but also explore the underlying relationships to integrate heterogeneous context features together to improve the recognition performance.

To test the robustness of our approach, we degrade the resolution and sampling rate of frames in our dataset respectively, and run a series of experiments on such dataset. Our algorithm mainly relies on context features such as activities, which are less sensitive to the deterioration of video quality. Figure 3.10 indicates that the decrease of frame resolution does not affect the performance of activity detection since the contextual information (such

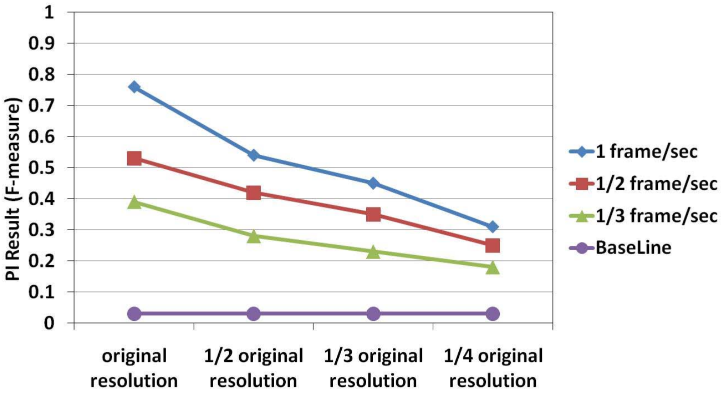


Figure 3.11: PI Result with Decreasing of Resolution and Sampling Rate

as time and location) is less sensitive to the frame resolution. But the performance of activity detection (suppose the performance with the original resolution and sampling rate is 100%) drops when sampling rate reduces from 1 frame/sec to 1/2 and 1/3 frame/sec, because many important frames are lost with the decrease of sampling rate. Figure 3.11 illustrates that person identification result drops with the reduction of resolution and sampling rate, due to the loss of activity and color information. However, person identification result of our algorithm even with the lowest resolution and sampling rate is much better than the baseline results of Naive Approach (which predicts results just based on the occurrence probability in the training dataset). Consequently, Figure 3.11 demonstrates the robustness of our approach with low quality video data, because our approach leverages contextual data rather than merely relying on the quality of video data.

3.4 Chapter Conclusion and Future Work

In this chater we considered the task of person identification in the context of Smart Video Surveillance. We have demonstrated how an instance of indoor person identification problem (for video data) can be converted into the problem of entity resolution (which typically deals with textual data). The area of entity resolution has become very active as of recently, with many research groups proposing powerful generic algorithms and frameworks. Thus, establishing a connection between the two problems has the potential to benefit the person identification problem, which could be viewed as a specific instance of ER problem. Our experiments of using a simplified version of RelDC framework for entity resolution have demonstrated the effectiveness of our approach.

This work is, however, only a first step in exploiting ER techniques for video data cleaning tasks. Our current approach has numerous assumptions and limitations: (1)The approach assumes that color of clothing is a strong identifier for a person on a given day; if several people wear similar color clothes and have similar activities, it is hard to distinguish them using the current approach. (2)If several people appear together, it is sometimes hard for the algorithm to correctly separate these subjects, and this negatively affects the result. Our future work will explore how additional features derived from video, as well as additional semantics in the form of context and metadata (e.g., knowledge of building layout, offices, meeting times, etc.) can be used to further improve person identification.

Chapter 4

Context Assisted Face Clustering Framework with Human-in-the-Loop

In this chapter, we study *the problem of context-assisted face clustering with human-in-the-loop.*

With the explosion of massive media data, the problem of image organization, management and retrieval has become an important challenge [43][75][77][101]. Naturally, the focus in many image collections is people. To better understand and manage the human-centered photos, face tagging that aims to help users associate people names with faces becomes an essential task. The fundamental problem towards face tagging and management is face clustering, which aims to group faces that refer to the same people together.

Clustering faces by utilizing facial appearance features is the most conventional approach. It has been extensively studied and significant progress has been achieved in the last two decades [4][6][7][28][29]. These standard techniques have already been employed in several commercial systems such as Google Picasa, Apple iPhoto, and Microsoft EasyAlbum. These systems usually produce face clusters that have high precision (faces in each cluster refer

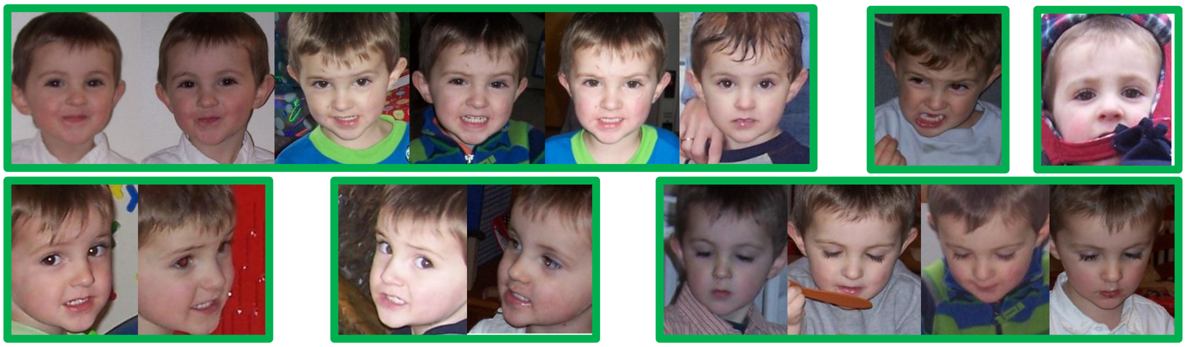


Figure 4.1: Example of Face Clusters by Picasa

to the same person), but low recall (faces of a single person fall into different clusters). In addition, a large number of small/singleton face clusters are often returned, which bring heavy burden on the users to label all the faces in the album. Fig. 4.1 illustrates the example of face clustering result, where faces of a single person fall into six different (pure) clusters, instead of one. One reason for low recall is due to the large variation of faces in pose, expression, illumination, occlusion, etc. That makes it challenging to group faces correctly by using the standard techniques that focus primarily on facial features and largely ignore the context. Another reason is that when systems like Picasa ask for manual feedback from the user, users most often prefer to merge pure (high-precision) clusters rather than manually clean contaminated (low-recall) ones. Consequently, such systems are often tuned to strongly prefer the precision over recall. The goal of our work is to leverage heterogeneous context information to improve the recall of cluster results without reducing the high precision.

Prior research efforts have extensively explored using contextual features to improve the quality of face clustering [73][86][97][98][100]. In general, in contrast to our work, such techniques often aim at exploring just one (or a few) contextual feature types, with the merging decision often made at the image level only. We, however, develop a unified framework that integrates heterogeneous context information together to improve the performance of face clustering. The framework learns the roles and importance of different feature types from data. It can take into account time decay of features and makes the merging decision at both image and cluster levels. Examples of types of contextual cues that have been used in the past include geo-location and image capture time [101], people co-occurrence [54][73][86] , social norm and conventional positioning observed[34], human attributes [52], text or other linked information[12][87], clothing [33][100], etc. For instance, [52] proposes to employ human attributes as an additional features. However, the authors do not explore the different roles that each attribute type plays in identifying different people. Social context, such as people co-occurrence, has been investigated in [54][73][86]. But these approaches do not deal with cluster-level co-occurrence information. Clothing information has been used extensively in face clustering [33][100]. However, these techniques do not employ the important time decay factor in leveraging clothing information.

This chapter is an extension of our previous work[95], above on which we integrate human-in-the-loop to the unified context assisted framework. The unified framework is illustrated in Figure 5.4. We start with the initial set of clusters generated by the standard approach for the given photo collection. The initial clusters have high precision but low recall. We iteratively merge the clusters that are likely to refer to the same entities to get higher recall. We use contextual and facial features in two regards: for computing similarities (how similar are two clusters) and for defining constraints (which clusters cannot refer to the same person). The framework then uses bootstrapping to learn the importance of different heterogeneous feature types directly from data. To achieve higher quality, this learning is done adaptively per cluster in a photo collection, because the importance of different features can change from person to person and in different photo collections. For example, clothing is a good distinguishing feature in a photo album where people's clothes are distinct, but a weak feature in a photo collection where people are wearing uniform. We employ the ideas of bootstrapping to partially label any given dataset in automated fashion without any human input. These labels then allow us to learn the importance of various features directly from the given photo collection. Clusters are then merged iteratively, based on the importance of the learned features and computed similarity, to produce a higher quality clustering. Finally,

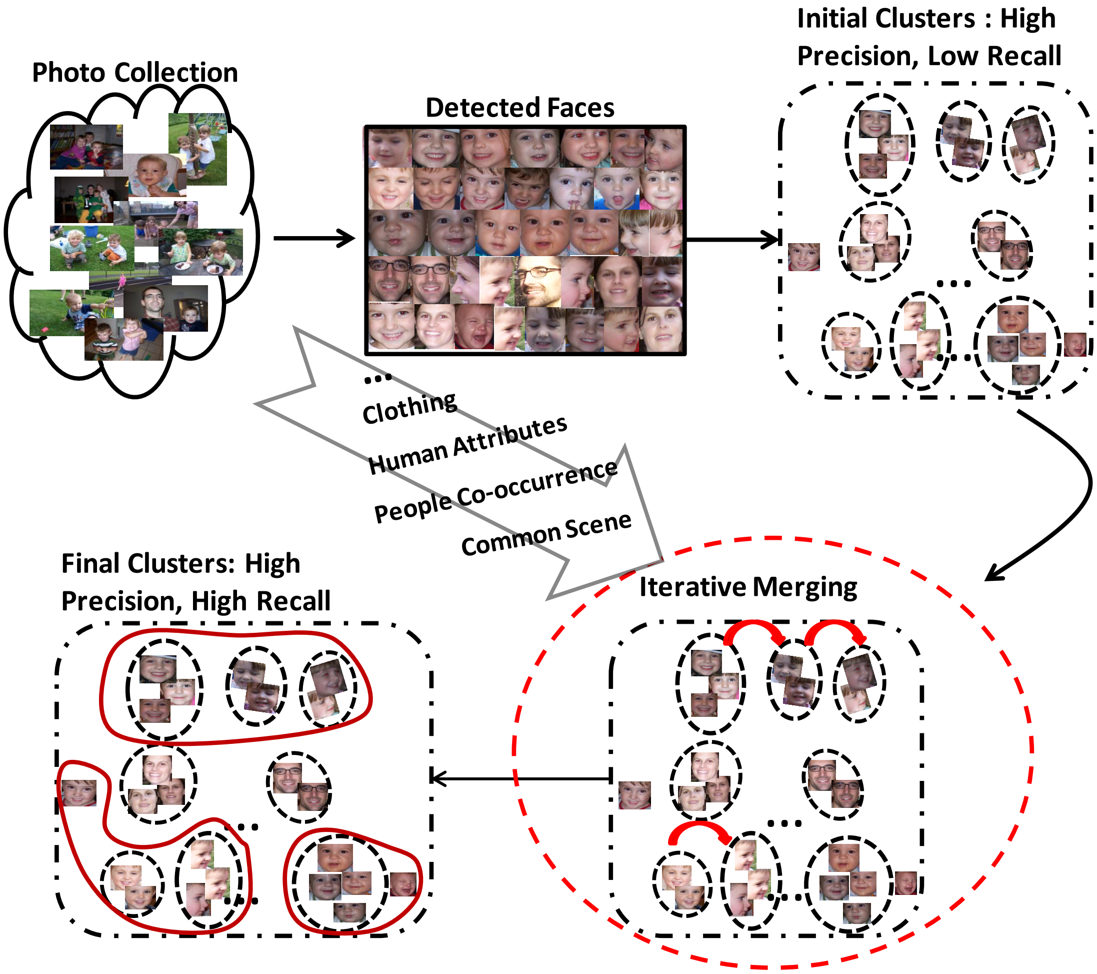


Figure 4.2: The General Framework

we discuss the proper method to integrate human-in-the-loop in order to leverage human interaction to achieve the very high-quality clustering results.

The rest of this chapter is organized as follows. We first formally define the problem in Section 4.1. In Section 4.2, we describe how to leverage the context information at the cluster level, including common scene, people co-occurrence, human attributes, and clothing. In Section 4.3, we propose the unified framework which automatically learns rules to integrate heterogeneous context information together to iteratively merge clusters. In Section 4.4, we discuss the proper method to integrate human-in-the-loop process. The proposed approach is empirically evaluated in Section 4.5. Finally, we conclude in Section 4.6 by highlighting key points of our work.

4.1 Problem Definition

Suppose that a human-centered photo album P_h contains K images denoted as $\{I_1, I_2, \ldots, I_K\}$, see Figure 5.4. Assume that n faces are detected in P_h, with each face denoted as f_i for $i = 1, 2, \ldots, n$, or $f_i^{I_k}$ (that is, f_i is extracted from image I_k). Suppose that by applying the standard algorithm which is based on facial features, we obtain N clusters $\{C_1, C_2, \ldots, C_N\}$, where each cluster is assumed to be pure, but multiple clusters could refer to the same entity. Our goal is to leverage heterogeneous context information to merge clusters such that we still get very high precision clusters but also improve the recall.

There have been many studies that analyze behaviors of different metrics for measuring quality of clustering. A recent prominent study by Artiles et al. suggests that B-cubed precision, recall and F-measure is one of the best combination of metrics to use according to many criteria [5]. Let $C(f_i)$ be the cluster that f_i is put into by a clustering algorithm. Let $L(f_i)$ be to the real category/label (person) f_i refers to in the ground truth. Given two faces f_i and f_j, the correctness $Correct(f_i, f_j)$ is defined as:

$$Correct(f_i, f_j) = \begin{cases} 1 & \text{if } L(f_i) = L(f_j) \wedge C(f_i) = C(f_j) \\ 0 & \text{otherwise} \end{cases}$$

B-cubed precision of an item f_i is computed as the proportion of correctly related items in its cluster (including itself): $Pre(f_i) = \frac{\sum_{f_j : C(f_i) = C(f_j)} Correct(f_i, f_j)}{\|\{f_j | C(f_i) = C(f_j)\}\|}$. The overall B-cubed precision is the averaged precision of all items: $Pre = \frac{1}{n}\sum_{i=1}^{n} Pre(f_i)$. Similarly, B-cubed recall of f_i is the proportion of correctly related items in its category: $Rec(f_i) = \frac{\sum_{f_j : L(f_i) = L(f_j)} Correct(f_i, f_j)}{\|\{f_j | L(f_i) = L(f_j)\}\|}$.

The overall recall is then: $Rec = \frac{1}{n}\sum_{i=1}^{n} Rec(f_i)$. The F-measure is then defined as the harmonic mean of the precision and recall.

4.2 Context Feature Extraction

Most prior research effort focus on leveraging context features directly at the face level[33][52][54]. That is, the similarity is computed between two faces and not two clusters. In this section, we will describe how to utilize context features at the cluster level. Context features are not only able to provide additional contextual *similarity* information to link clusters that co-refer (refer to the same entity), but also generate *constraints* that identify clusters that cannot co-refer (cannot refer to the same entity).

4.2.1 Context Similarities

Common Scene

It is common for a photographer to take multiple photos of the same "scene" in a relatively short period of time. This phenomenon happens for example when the photographer wants to ensure that at least some of the pictures taken will be of acceptable quality, or when people pose for photos and change their poses somewhat in the sequence of common scene photos. Common scene photos are often taken within small intervals of time from each other and they contain almost the same background and almost the same group of people in each photo. Surprisingly, we are not aware of much existing work that would use common scene detection to improve face-clustering performance. However common scene detection can provide additional evidence to link clusters describing the same entity, since images in a common scene often contain the same people.

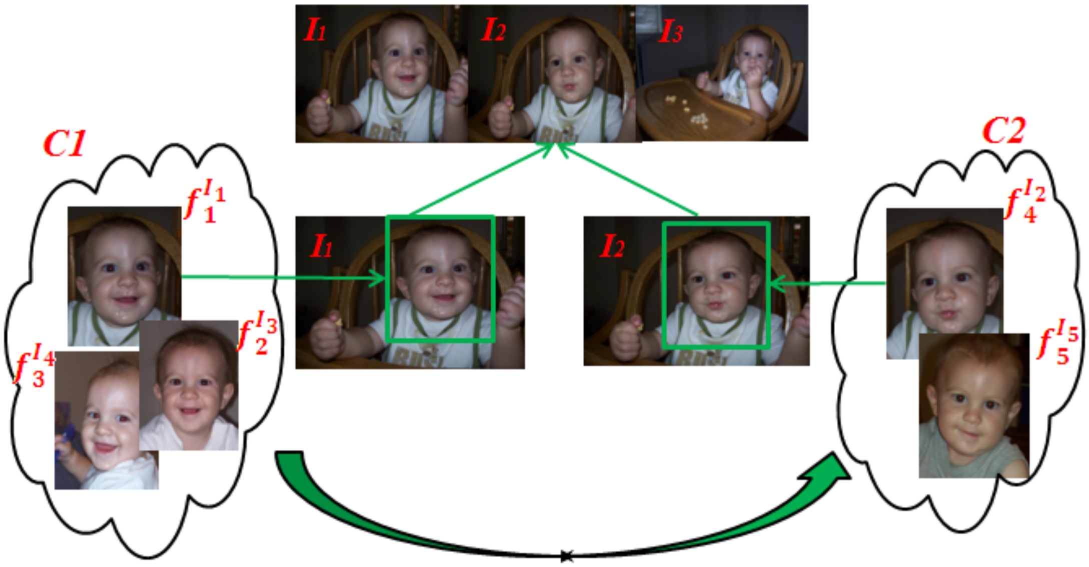


Figure 4.3: Example of Common Scene

To divide images into common scene clusters, some EXIF information (such as image captured time, geo-location, camera model, etc.), and image visual features (color, texture, shape) and image file name can be leveraged. Suppose that in a photo album P_h containing K images $\{I_1, I_2, ..., I_K\}$, the algorithm finds M common scene clusters. Let $CS(I_k)$ denotes the common scene of image I_k. Based on the assumption that two images forming the common scene might describe the same entities, two entities even with dissimilar facial appearances might be linked by the common scene.

For example, as shown in Figure 4.3, C_1 and C_2 are two initial face clusters based on face appearance. Face $f_1^{I_1}$, extracted from image I_1, belongs to cluster C_1, and face $f_4^{I_2}$ extracted from image I_2 is put into C_2. Since images I_1 and I_2 share the common scene $CS(I_1) = CS(I_2)$, it is possible they describe the same entities. Thus faces $f_1^{I_1}$ and $f_4^{I_2}$ have some possibility to be the same, and the two face clusters C_1 and C_2 are linked to each other via the common scene.

Thus the context similarity $S^{cs}(C_m, C_n)$ of two face clusters C_m and C_n based on common scene is defined as the number of distinct common scenes between the pairs of images from each cluster:

$$\mu_{mn}^{cs} = \{CS(I_k)|CS(I_k) = CS(I_l)) \wedge (f_i^{I_k} \in C_m) \wedge (f_j^{I_l} \in C_n)\} \tag{4.1}$$

$$S^{cs}(C_m, C_n) = \parallel \mu_{mn}^{cs} \parallel \tag{4.2}$$

Thus μ_{mn}^{cs} is the set of common scenes across two face clusters C_m and C_n. $S^{cs}(C_m, C_n)$ is the cardinality of set μ_{mn}^{cs}. The larger value $S^{cs}(C_m, C_n)$ is, the higher the likelihood that C_m and C_n refer to the same entity.

People Co-occurrence

The surrounding faces can provide vital evidence in recognizing the identity of a given face in an image. Suppose that "Rose" and "John" are good friends and often take photos together, then the identity of one person will probably imply the other. In [86], Wu et al. investigated people co-occurrence feature and proposed a social context similarity measurement by counting the common co-occurred single clusters between two clusters. However, this measurement could be greatly improved because single cluster linkage alone is not strong evidence. In this section, we propose a new social context similarity measurement, which use the common cluster-group as evidence to link clusters. Experiments reveal that the linkage of cluster-groups is more reliable than the linkage of single cluster.

Cluster co-occurrence. First, let us define the co-occurrence relationship between two clusters. We will say that clusters C_m and C_n *co-occur* in/via image I_k, if I_k contains at least two faces such that one is from C_m and the other one is from C_n. In general, the co-occurrence measure $Co(C_m, C_n)$ returns the number of distinct images in which C_m and C_n co-occur:

$$Co(C_m, C_n) = \parallel \{I_k | \exists f_i^{I_k}, f_j^{I_k} \text{ s.t. } (f_i^{I_k} \in C_m) \wedge (f_j^{I_k} \in C_n)\} \parallel$$

The co-occurrence relationship between three and more face clusters has a similar definition. Consider the faces in Figure 4.4 as an example. There, C_1, C_2, C_3, C_4 are four initial face clusters. Since there exists an image I_1 that contain three faces f_1, f_4 and f_6 such that $f_1 \in C_1$, $f_4 \in C_2$, $f_6 \in C_3$, thus $Co(C_1, C_2, C_3) = 1$. Similarly, for the clusters C_1, C_2, C_4 it holds $Co(C_1, C_2, C_4) = 1$. Based on common sense, we know that a person cannot co-occur with himself in an image unless the image is doctored or contains a reflection, e.g., in a mirror. Consequently, clusters connected via a non-zero co-occurrence relationship should refer to different entities. This property will be used later on by the framework to generate context *constraints*.

Co-occurrence graph. The co-occurrence of two face clusters reveals the social relationship between them and between the people they correspond to. We now will describe how to construct cluster co-occurrence graph. Observe that if two face clusters have similar co-occurrence relationships, then the two face clusters might refer to the same entity. This is since people tend to appear with the same group of people in photos, e.g., the same friends. In the example in Figure 4.4, both C_3 and C_4 co-occur with C_1 and C_2. Such co-occurrence can serve as extra evidence that C_3 and C_4 possibly refer to the same entity. Notice, to demonstrate this graphically, we can represent C_3 and C_4 as nodes in a graph both of which are linked together via a different node that corresponds to C_1 and C_2 as a single cluster-group.

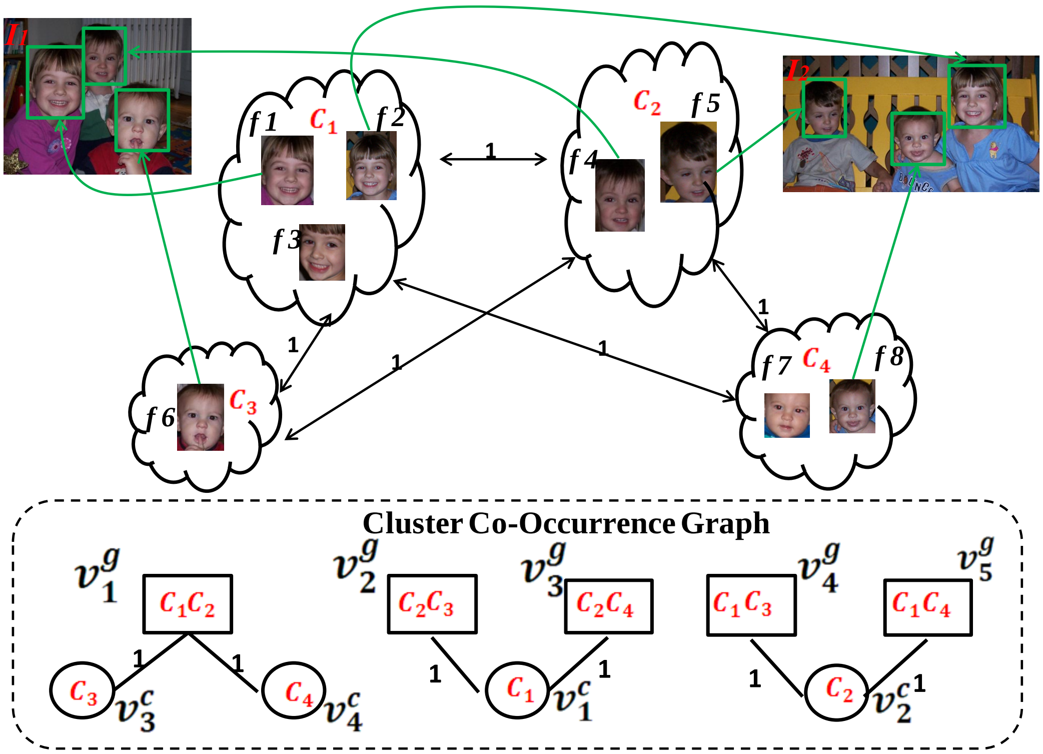


Figure 4.4: Example of People Co-occurrence

To analyze the various co-occurrences among clusters, we construct the cluster co-occurrence graph $G = (V, E)$. G is a labeled undirectional graph. The set of nodes V in the graph consists of two types of nodes: $V = V^c \cap V^g$. Node $v_i^c \in V^c$ corresponds to each single face cluster C_i. Node $v_j^g \in V^g$ corresponds to each face cluster-group found in an image. The group nodes are constructed as follows. For each image I_k that contains at least two faces, let Φ^{I_k} denote the set of all the clusters that contain faces present in I_k. We construct $\parallel \Phi^{I_k} \parallel$ cluster-groups, where each group is a set of clusters $\Phi^{I_k} \setminus \{C_j\}$ for each $C_j \in \Phi^{I_k}$. For example, if image I_1 has faces for three clusters $\Phi^{I_1} = \{C_1, C_2, C_3\}$, then the groups are going to be $\{C_1, C_2\}$, $\{C_1, C_3\}$, and $\{C_2, C_3\}$. A node v_j^g is created once per each distinct group. Edge $e_{ij} \in E$ is created between nodes v_i^c and v_j^g only when v_i^c occurs in the context of group v_j^g at least once, that is when exists at least one image I_k such that $v_i^c \cap v_j^g = \Phi^{I_k}$. Edge e_{ij} is labeled with the number of such images, i.e., edge weight $w_{ij} = \parallel \{I_k | v_i^c \cap v_j^g = \Phi^{I_k}\} \parallel$.

Consider Figure 4.4 as an example. For images I_1 and I_2 we have $\Phi^{I_1} = \{C_1, C_2, C_3\}$,

$\Phi^{I_2} = \{C_1, C_2, C_4\}$. Thus we construct four V^c nodes for C_1, C_2, C_3, C_4, and five V^g nodes for $\{C_1, C_2\}$, $\{C_1, C_3\}$, $\{C_2, C_3\}$, $\{C_1, C_4\}$, $\{C_2, C_4\}$. Edges are created accordingly.

From the cluster co-occurrence graph, we observe that if two V^c nodes v^c_m and v^c_n connects to the same V^g node v^g_k, then v^c_m and v^c_n possibly refer to the same entity. For instance, in Figure 4.4, both C_3 and C_4 connects with $\{C_1, C_2\}$, so C_3 and C_4 are possibly the same. The context similarity from cluster co-occurrence $S^{co}(C_m, C_n)$ for C_m and C_n can be then defined as the flow between these two clusters,

$$S^{co}(C_m, C_n) = \sum_{{V^g}_k \leftrightarrow {V^c}_m, {V^g}_k \leftrightarrow {V^c}_n} \min(w_{mk}, w_{kn}) \tag{4.3}$$

In general, the co-occurrence similarity between two clusters can be measured as the sum of weights of paths which link them through V^g nodes. The larger the number/weight of paths that link C_m and C_n, the higher the likelihood that C_m and C_n refer to the same entity.[1]

Human Attributes

Human attributes, such as gender, age, ethnicity, facial traits, etc., are important evidence to identify a person. By considering attributes, many uncertainties and errors for face clustering can be avoided, such as confusing “men” with “women”, “adults” with “children”, etc. To get attribute values for a given face, we use the attribute system[52]. It returns values for the 73 types of attributes, such as “black hair”, “big nose”, or “wearing eyeglasses”. Thus, with each face f_i we associate a 73-D attribute vector denoted as A^{f_i}.

[1] Notice, in general, there could be different models for assigning weights to paths in addition to the flow model considered in the chaper. For example, paths that go through larger group nodes could be assigned higher weight since larger groups of people tend to be better context than smaller ones.

In [52], Kumar et al. suggests that attributes can be used to help face verification by choosing some measurement (e.g., cosine similarity) to compute attribute similarities. However, the importance of each type of attribute usually differs when identifying different entities. For example, in a photo album containing just one baby, age is an important factor for identifying this baby; while if several babies exist in an album, then age is not a strongly discriminative feature. Thus, it is essential to determine the importance of attributes for identifying a given entity in the photo collections.

To achieve this, we learn the importance of attributes from the face cluster itself, by leveraging *bootstrapping*. Here, bootstrapping refers to the process of being able to automatically label part of the data, without any human input, and then use these labels to train a classifier. The learned classifier is then used to label the remaining data. One of the main challenges in applying bootstrapping is to be able to provide these partial labels. The general idea of our solution is that faces that belong to one face cluster are very likely to refer to the same entity due to the purity of the initial clusters, hence they can form the positive samples. In turn, faces from two clusters that co-occur in the same image most likely refer to the different people (since a person cannot co-occur with himself in a photo), which can be used to construct the negative samples.

Based on the above discussion, the training dataset can be constructed for each cluster. Figure 4.5 illustrates the attribute training dataset for identifying C_1 from the example in Figure 4.4. Three faces f_1, f_2, f_3 fall into C_1, so the attributes of these three faces A^{f_1},A^{f_2},A^{f_3} are labeled as C_1. Since the other three clusters C_2, C_3, C_4 have the co-occurrence relationship with C_1, they are considered to describe different entities. Thus the attributes of faces from the other three clusters can be treated as the negative samples. In this way, the attribute training dataset can be constructed automatically for each cluster.

After the attribute training dataset is constructed, a classifier, such as SVM, can be learned for each cluster C_m. Given a 73-D attribute feature A^{f_i} for any face f_i, the task of the

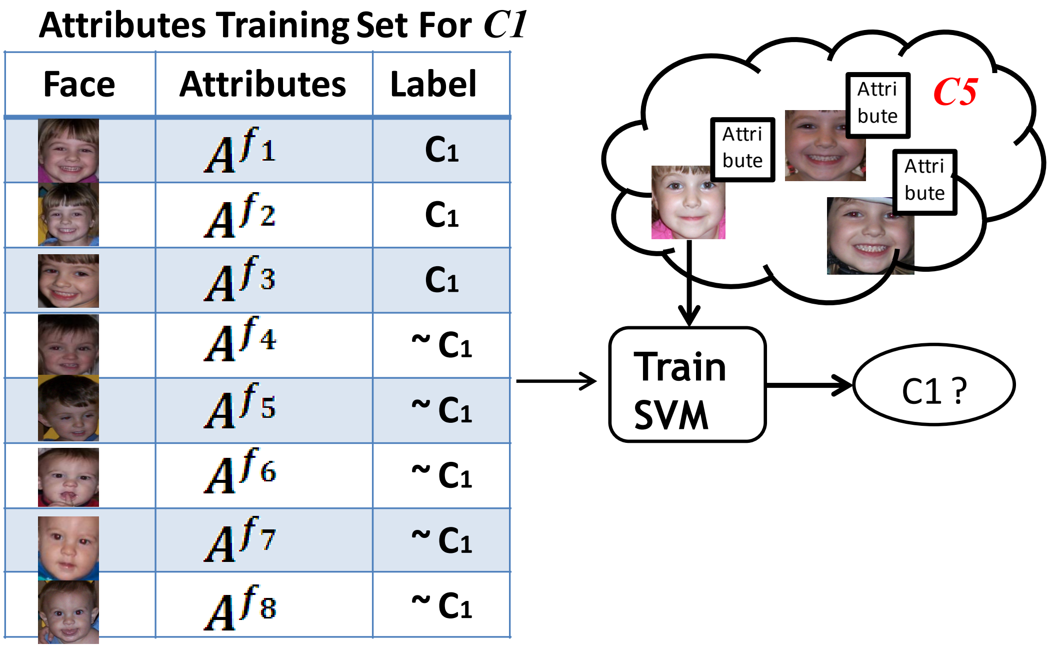


Figure 4.5: Example of Human Attributes

classifier is to output whether this face f_i belongs to C_m. In addition to outputting a binary yes/no decision, modern classifiers can also output the probability that f_i belongs to C_m, denoted as $P^A(f_i \in C_m)$. Thus, by applying classifier learned for C_m to each face in an unknown face cluster C_n, we can compute the average probability that C_n belongs to C_m, denoted as $S^A(C_n \rightsquigarrow C_m)$:

$$S^A(C_n \rightsquigarrow C_m) = \frac{1}{\parallel C_n \parallel} \sum_{f_i \in C_n} P^A(f_i \in C_m) \tag{4.4}$$

Attribute similarity between C_m and C_n is defined as,

$$S^{attr}(C_m, C_n) = \frac{S^A(C_n \rightsquigarrow C_m) + S^A(C_m \rightsquigarrow C_n)}{2} \tag{4.5}$$

That is, the attribute based similarity $S^{attr}(C_m, C_n)$ between two clusters is the average of the average probability of one cluster to belong to the other.

Clothing Information

Clothing information could be a strong feature for determining the identity of a person. However, clothing is a time-sensitive feature since people can change their clothes. Clothing has been considered in the previous work for face clustering, e.g. in [100], but not as a time-sensitive feature described next.

In this section, we introduce time decay factor to control the effect of clothing in identifying people. We propose that the similarity between f_i and f_j should be a function of time:

$$S^c(f_i, f_j) = sim(ch_{f_i}, ch_{f_j}) \times e^{-\triangle t/2s^2} \tag{4.6}$$

In the above formula, $sim(ch_{f_i}, ch_{f_j})$ refers to the clothing similarity computed only on visual features. Notation $\triangle t$ refers to the capture time difference between 2 faces. By construction, the above time-decay function incorporates the relationship between $\triangle t$ and the effectiveness of clothing features. The smaller $\triangle t$ is, the more effective clothing feature is. With the time difference value $\triangle t$ growing, the effectiveness of clothing feature is decreasing. When the time difference $\triangle t$ is much larger than the time slot threshold s, the clothing feature becomes ineffective.

To compute the clothing similarity, the first step is to detect the location of clothing for the given face, which can be implemented by leveraging the techniques from [33] or simply using a bounding box below detected faces. After that, some low level image features (color, texture) can be extracted to represent the clothing information, and then similarities can be computed.

To obtain the cluster similarity from clothing information, we can compute the clothing similarity between each pair of faces and then choose the maximum value:

$$S^{cloth}(C_m, C_n) = \max_{f_i \in C_m, f_j \in C_n} S^c(f_i, f_j) \tag{4.7}$$

Thus the clothing similarity between C_m and C_n is computed by selecting the maximum clothing similarity between each pair of faces respectively falling in the 2 face clusters.

4.2.2 Context Constraints

In the previous section we have explained how context features can be used as extra positive evidence for computing similarity between clusters. Context features, such as people co-occurrence and human attributes, can also provide *constraints* or negative evidence, which can be used to identify clusters that should refer to different entities.

From cluster co-occurrence relationship, we can derive that two face clusters with $Co(C_m, C_n) > 0$ should refer to definitely different entities, because a person cannot co-occur with himself (in normal cases). Thus we can define that if $Co(C_m, C_n) > 0$, the context dissimilarity from co-occurrence feature is 1, denoted as $D^{co}(C_m, C_n) = 1$.

From human attributes, we can derive that two clusters with vastly different attributes values, such as age, gender, ethnicity information should refer to different entities. Thus we can define that if two clusters C_m and C_n have distinct age, gender, ethnicity attribute values, then context dissimilarity from human attributes feature is 1, referred as $D^{attr}(C_m, C_n) = 1$. Then we can define the context dissimilarity measurement between two clusters as follows:

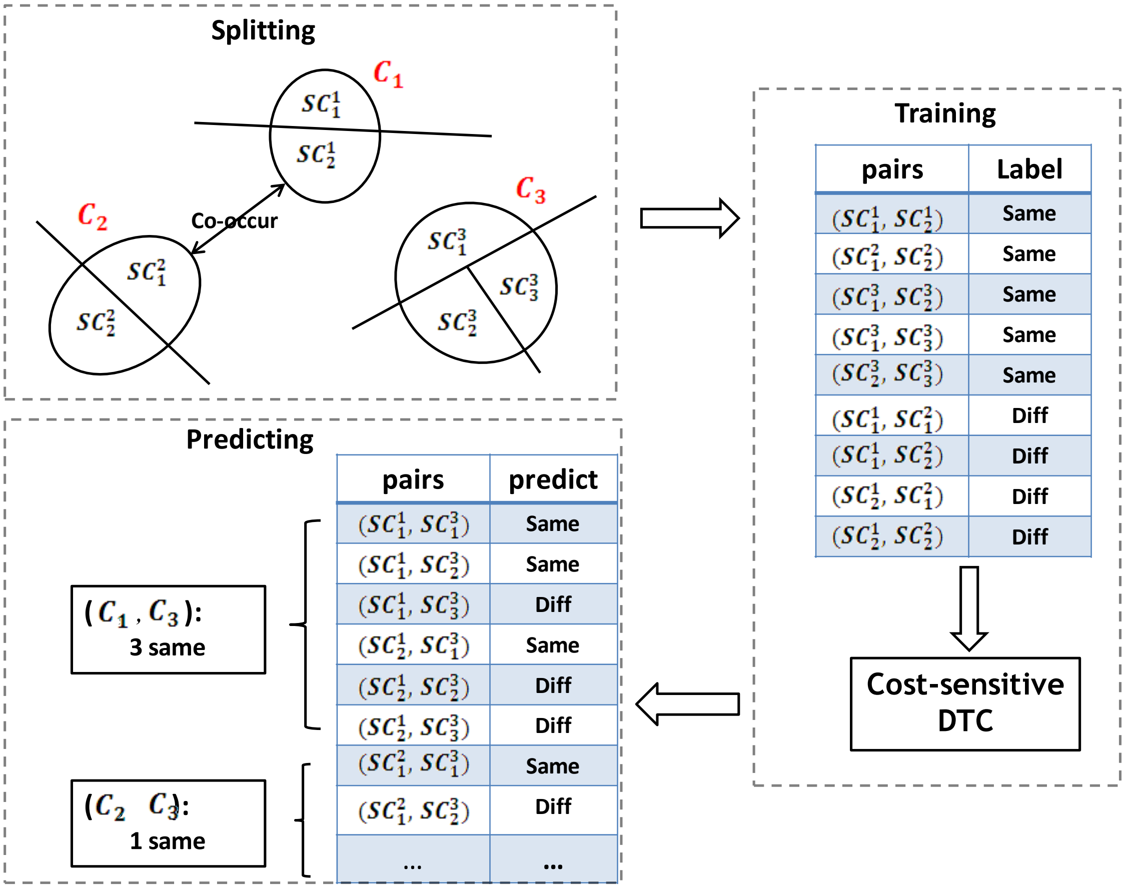
$$D(C_m, C_n) = \begin{cases} 1 & \text{if } D^{co}(C_m, C_n) = 1 \text{ or } D^{attr}(C_m, C_n) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Thus $D(C_m, C_n) = 1$ means C_m and C_n are most likely different, $D(C_m, C_n) = 0$ means that the dissimilarity measure between C_m and C_n cannot tell if they are different or not. The context constraints will be leveraged to implement the bootstrapping ideas explained in the following section.

4.3 The Unified Framework

In the previous section we have discussed how to leverage the context information from two aspects: computing context similarities (S^{cs}, S^{co}, S^{attr}, S^{cloth}) and context constraints (D^{co}, D^{attr}). In this section, we will develop an approach for integrating these heterogeneous context features together to facilitate face clustering.

One possible solution for aggregating these context features is to compute the overall similarity as weighted linear sum of the context similarities. The overall similarity can then be used to merge clusters that do not violate the context constraints. However, this basic solution has several limitations: it is too coarse-grained and it could be difficult to set the weights that would work best for all possible photo collections. Alternatively, the other option is to automatically learn some rules to combine these context features together to make a merging decision. If the rules are satisfied, the two face clusters can be merged. For example, a rule could be if $S^{cs}(C_m, C_n) > 3$ and $S^{co}(C_m, C_n) > 4$, then merge C_m and C_n. The experiments reveal that if the rules are defined appropriately, significantly better merging results can be achieved compared to the basic solution.



pairs	Label
(SC_1^1, SC_2^1)	Same
(SC_1^2, SC_2^2)	Same
(SC_1^3, SC_2^3)	Same
(SC_1^3, SC_3^3)	Same
(SC_2^3, SC_3^3)	Same
(SC_1^1, SC_1^2)	Diff
(SC_1^1, SC_2^2)	Diff
(SC_2^1, SC_1^2)	Diff
(SC_2^1, SC_2^2)	Diff

pairs	predict
(SC_1^1, SC_1^3)	Same
(SC_1^1, SC_2^3)	Same
(SC_1^1, SC_3^3)	Diff
(SC_2^1, SC_1^3)	Same
(SC_2^1, SC_2^3)	Diff
(SC_2^1, SC_3^3)	Diff
(SC_1^2, SC_1^3)	Same
(SC_1^2, SC_2^3)	Diff
...	...

Figure 4.6: Example of Bootstrapping Process

Nevertheless, it is hard to define and fix rules that would work well for all possible photo albums. Instead, rules that are automatically tuned to each photo collection would naturally perform better. This is since the importance of each type of context feature usually varies due to the diversity of image datasets. For example, clothing might be important evidence in a photo album where people's clothing is distinct, but it will lose the effect in a photo collection where people wearing uniform. Thus, inspired by [20][45][62][99], we propose a unified framework that can automatically learn and adapt the rules to get high quality of face clustering.

4.3.1 Construction of Training Dataset

To automatically learn the rules, training dataset is often required. However, since we are trying to automatically learn and tune the rules per each photo collection, it is unlikely that training data will be available, as it will not accompany each given collection. Nevertheless, such rules could be learned by leveraging bootstrapping and semi-supervised learning techniques. To apply those techniques, we need to automatically partially label the dataset. The constructed training dataset should contain positive samples (same face cluster pairs) and negative samples (different face cluster pairs). The key challenge is to be able to automatically, without any human input, label the positive and negative samples for part of the data.

In the above section, we discuss that the context information can provide constraints to distinguish clusters referring to different entities. For example, two face clusters with co-occurrence relationship, or distinct attribute values (age, gender, ethnicity), are most likely different. Based on this observation, the negative samples can be constructed.

Then the next issue becomes how to obtain the positive pairs. Due to the purity of initial face clusters, faces that are part of one face cluster refer to the same entity. If we split an initial face cluster into smaller clusters, then these split smaller clusters should refer to the same entity. Thus the split smaller clusters will form the positive sample pairs.

Strategy for Splitting Clusters

Many splitting strategies can be adopted for splitting existing pure clusters into subclusters. For example, one equi-part strategy is to split each initial face cluster into two (or other fixed number of) roughly equally-sized subclusters. An alternative equi-size strategy is to predefine the subcluster size (e.g., $sz = 10$ faces) and then split each cluster into subclusters

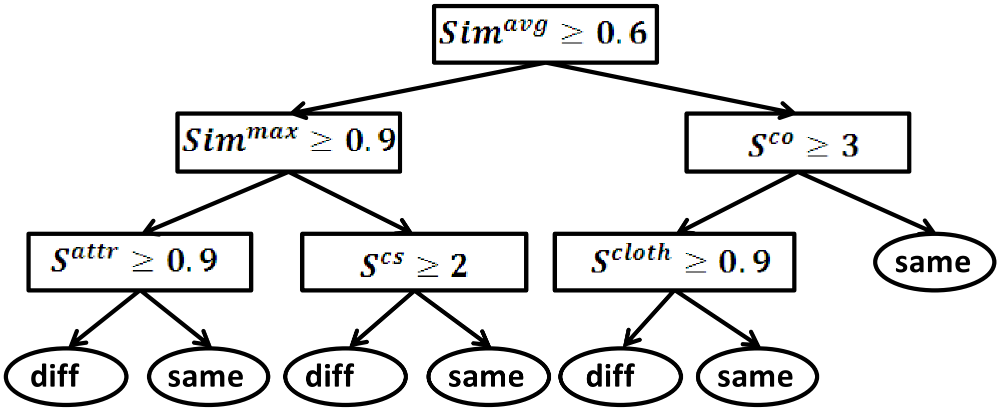


Figure 4.7: Example of Decision Tree Classifier

of that size. The equi-size strategy has demonstrated a consistent advantage over other tested options since some of the context features similarities depend on cluster sizes. For example, the context similarity between two large clusters is usually stronger than the similarity between two small clusters. Thus, by considering split clusters of roughly the same size, the effect of cluster size is reduced.

Consider N initial pure face clusters $C_1, C_2, \ldots, C_N$, and the predefined subcluster size is sz. Then each cluster C_m with $\| C_m \| > sz$, can be randomly divided into $\left\lceil \frac{\|C_m\|}{sz} \right\rceil$ subclusters, denoted as $\{SC_1^m, SC_2^m, \ldots\}$. Figure 4.6 illustrates an example of splitting clusters.

Automatic Labeling

After splitting clusters into subclusters, the next task is to automatically label the positive and negative training samples. Due to the purity of the initial face clusters, if two subclusters come from the same initial cluster, they form the positive sample, labeled as the "same" pair. If two subclusters come from two different clusters that have co-occurrence relation or distinct attribute values, then the two subclusters form the negative sample, labeled as "diff" pair. Thus, given two subclusters SC_i^m and SC_j^n, the label $La(SC_i^m, SC_j^n)$ can be generated as follows:

$$La(SC_i^m, SC_j^n) = \begin{cases} same & \text{if } m = n, \\ diff & \text{if } D(C_m, C_n) = 1, \\ unknown & \text{otherwise.} \end{cases}$$

Figure 4.6 illustrates how to construct the training dataset. As shown in Figure 4.6, subcluster pairs coming from the same initial cluster are labeled as "same" pairs, e.g., (SC_1^1, SC_2^1), (SC_1^2, SC_2^2), etc. Since C_1 and C_2 have the co-occurrence relationship, each subcluster pair respectively deriving from C_1 and C_2 will compose the "diff" pairs, e.g., (SC_1^1, SC_1^2), (SC_1^1, SC_2^2), etc.

Feature Construction

After splitting clusters into subclusters, the algorithm will try to determine which subclusters refer to the same entity. To do that, it first needs to associate a feature vector with each subcluster pair. After that, it will use a classifier to predict whether or not the pair co-refers.

Specifically, for each pair of subclusters SC_i^m and SC_j^n the algorithm associates four features that correspond to the cluster level context similarities S^{cs}, S^{co}, S^{attr}, S^{cloth}, as described in the above section. In addition, the face appearance similarities between two subclusters are also important, which are measured in three ways: (1) the maximum similarity between face pairs, denoted as Sim^{max}; (2) the minimum similarity between face pairs Sim^{min}; (3) the average similarity of face pairs, referred as Sim^{avg}. Therefore, the algorithm associates 4 types of context features and 3 types of face-based features with each subcluster pairs. Other types of features can also be integrated to this unified framework.

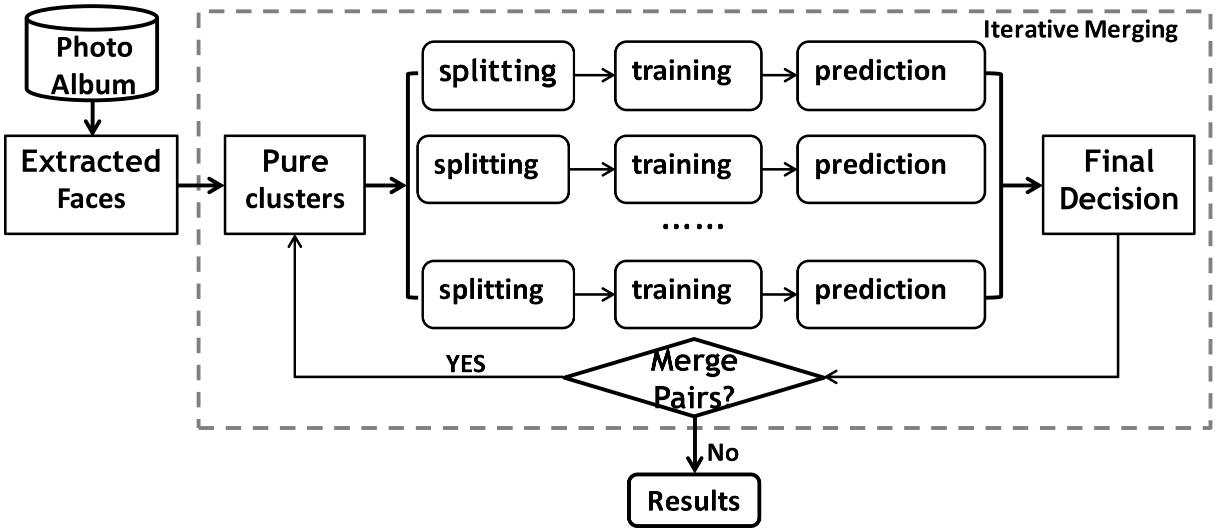


Figure 4.8: Iterative Merging Framework

4.3.2 Classifier Training and Predicting

After the automatic construction of the partially labeled training dataset, the next goal is to learn the merging rules from this training data. Then the learned rules can be applied to predict "same/different" labels for the pairs of subclusters that have been labeled "unknown" before. In this scenario, we choose to use *cost-sensitive* variant of the Decision Tree Classifier (DTC) as the classifier to learn the rules, though other classifiers might also be applied. The reason for using cost-sensitive and not regular DTC is that a single incorrect merge decision can very negatively affect the precision of clusters. That would defeat the purpose of our goal of improving the recall while maintaining the same high precision of the initial clustering. The cost-sensitive version of DTC allows to set the cost of false-positive errors to be much higher than that of false-negative errors. Therefore, we train a very conservative classifier which will try to avoid the false-positive errors thus ensuring high precision of the resulting clusters. To avoid over-fitting problem, we prune the over-fitted branches from the DTC. Figure 4.7 illustrated an example of the learned DTC.

As shown in Figure 4.6, the learned DTC can be applied to relabel previously "unknown" pairs by assigning "same" or "diff" labels. For example, in Figure 4.6, pair (SC_1^1, SC_1^3) is predicted to be “same”, and pair (SC_1^1, SC_3^3) to be “diff”.

To make the overall merging decision for the face clusters, we need to combine the decisions of the corresponding subclusters. For example, in Figure 4.6, analyzing the predictions for face cluster pair (C_1, C_3), we discover that 3 subcluster pairs are labeled “same”, and 3 pairs are labeled “diff”. Similarly, for cluster pair (C_2, C_3), 1 subcluster pair is labeled “same”, and 5 subcluster pairs are labeled “diff”. Hence the issue is how to make the final merging decision.

4.3.3 Final Merging Decision

Due to the randomness of the splitting strategy, the prediction results might differ with differently split clusters. To reduce the uncertainty introduced by the random splitting strategy, we propose to repeat the “splitting-training-predicting” process multiple times. If two face clusters are predicted to be “same” every time, then the face cluster pair should have higher probability to refer to the same entity.

Multiple Splitting-Training-Predicting

Based on the above discussion, the algorithm repeats the “splitting-training-predicting” process multiple times. Each time, the algorithm splits the initial face clusters into subclusters randomly, constructs the training dataset, trains the classifiers, predicts the “unknown” pairs, and then map the subcluster pairs predictions into the merge decisions. Let $T^{same}(C_m, C_n)$ be the number of times that face cluster pair C_m and C_n are predicted to be “same”. Similarly, let $T^{diff}(C_m, C_n)$ be the number of times they are predicted to be different. Naturally, the larger T^{same} is, the higher the probability is that this cluster pair refer to the same entity.

Final Decision

After perform the "splitting-training-predicting" process t times (e.g., $t = 5$), we can compute T^{same} and T^{diff} values for each pair of clusters, based on which the final merging decision can be made. For example, merge a pair when its $\frac{T^{same}+1}{T^{diff}+1}$ ratio exceed a certain threshold. To avoid early propagation of the incorrect merges, a higher threshold can be selected in the first several iterations, which can be decreased gradually in the subsequent iterations.

4.3.4 Iterative Merging Strategy

Figure 4.8 demonstrates the overall iterative merging framework. As shown in Figure 4.8, after the faces are extracted from the photo album, facial visual features are used to group the faces into initial clusters, which are very pure (high precision, low recall). Then our goal is to merge the pure cluster pairs in order to improve the recall without reducing the high precision. Leveraging multiple context information, and applying bootstrapping ideas, we perform the "splitting-training-predicting" process several times, and then make the combined merging decision. Based on the final decision, some face cluster pairs will be merged and updated. Before the actual merging operation, we need to guarantee that the merging pairs are not conflict based on constraints. And then the next iteration will be repeated until no merging pairs are obtained. Then the final automatic clustering results are achieved.

4.4 Improving Results by Leveraging Human-in-the-Loop Techniques

Thus far we have considered the construction of face clusters by only utilizing fully automated techniques that do not require any human involvement. While the proposed automated context-assisted framework reaches very high quality results, naturally it can make mistakes as well. This is especially the case after certain point in the execution: if the algorithm is forced to continue performing merges, then the resulting precision will drop, as the algorithm will start to base its decisions on weaker evidence.

This problem can be mitigated with the help of user feedback, that is, by using human-in-the-loop techniques. In general, human-in-the-loop mechanisms have been actively employed for similar purposes by many commercial applications, such as Google Picasa. The high-level goal of such techniques is to be able to get very high quality face clustering results while minimizing user participation. That is, the task is to provide an appropriate user interface as well as to design the right question-asking strategy, as naive solutions can easily overburden the user with too many unnecessary questions. We next describe the user interface and the algorithm for choosing questions to ask that are utilized in our approach. In the experiment section we will demonstrate that the proposed techniques outperform various baselines and reach higher quality results.

4.4.1 User Feedback Interface

Before asking the user for feedback, the algorithm first applies the fully-automated techniques described in the previous sections to merge clusters. It stops the merging process at the tipping point where the recall is significantly improved, precision is still very high, but the precision is about to start to drop. At that stage, the algorithm stops the fully automated

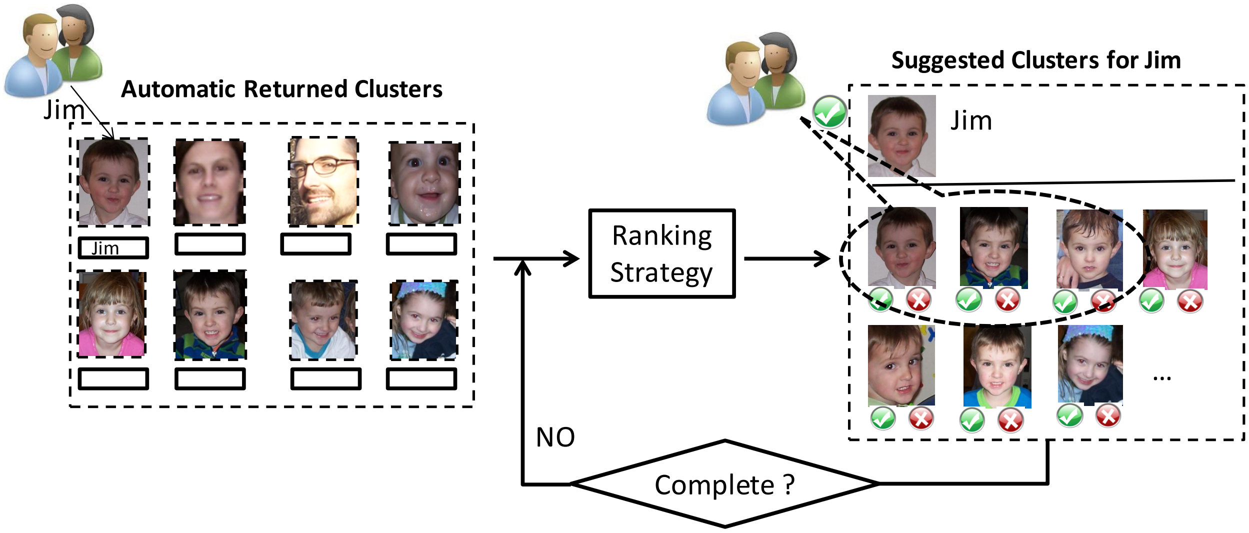


Figure 4.9: An Example of Human Computer Interaction Interface

process and starts leveraging the user feedback by asking the user to label the yet-unlabeled clusters manually.

Figure 4.9 illustrates the interface for the user feedback. It is based on the observation that often users prefer to disambiguate one person at a time instead of disambiguating all people at once. For instance, the user might want to first find and merge all clusters of say herself only, and then of her friends and relatives, and so on.

To accomplish this task, the interface initially shows all clusters to the user. To minimize the clutter, the interface represent each cluster by a single image selected to stand for this entire cluster. The user then chooses and labels one cluster as the “starting” or “target” cluster, signaling the algorithm that she now wants to focus on disambiguating this specific person. In Figure 4.9 (left part), this step is exemplified by the user choosing one cluster and labeling it as “Jim”. At that stage, the algorithm tries to help the user to disambiguate the chosen person by ranking the yet-unlabeled clusters and then presenting K of them in the ranked order to the user, see Figure 4.9 (right part). We will explain different ranking strategies later in this section.

The user then has the opportunity to provide feedback to the algorithm by clicking *yes* or *no* on all or some of the K clusters, indicating whether these clusters represent the *same* or *different* person as the starting cluster. After that, the algorithm factors in the newly provided user labeling, reranks the remaining yet-unlabeled clusters, and the process continues until the disambiguation is complete or when the user wishes to move on to the next person to disambiguate. We will next describe the ranking algorithm used in our solution. The algorithm leverages the fully automated approach itself to generate candidates for labeling and is capable of incorporating newly added user constraints to effectively filter out unnecessary questions.

4.4.2 Ranking Algorithm

The task of the ranking algorithm is to choose a set of K yet-unlabeled clusters to show for the user for the yes/no feedback. Intuitively, to generate this set the ranking algorithm will need to balance two criteria. First, the chosen clusters should represent the same person to the one being currently disambiguated. Second, the chosen clusters should have the most impact on reducing the overall uncertainty in the system and on increasing the quality (e.g. recall) of disambiguating the chosen person.

Figure 4.10 illustrates the proposed ranking algorithm. It consists of two steps: (1) the selection of candidate set and (2) the ranking of candidate nodes selected on Step 1. Based on the results of these two steps, K clusters are chosen to display to the user on the next iteration. This 2-step process is repeated iteratively after each time the user provides the feedback. In the first step, the algorithm chooses a pool of (more than K) potential candidate clusters. The choice is made based on the likelihood for these candidate clusters to be the same as the person being disambiguated. In the second step, the algorithm examines the chosen pool of clusters to pick K of them, such that asking feedback for them would result

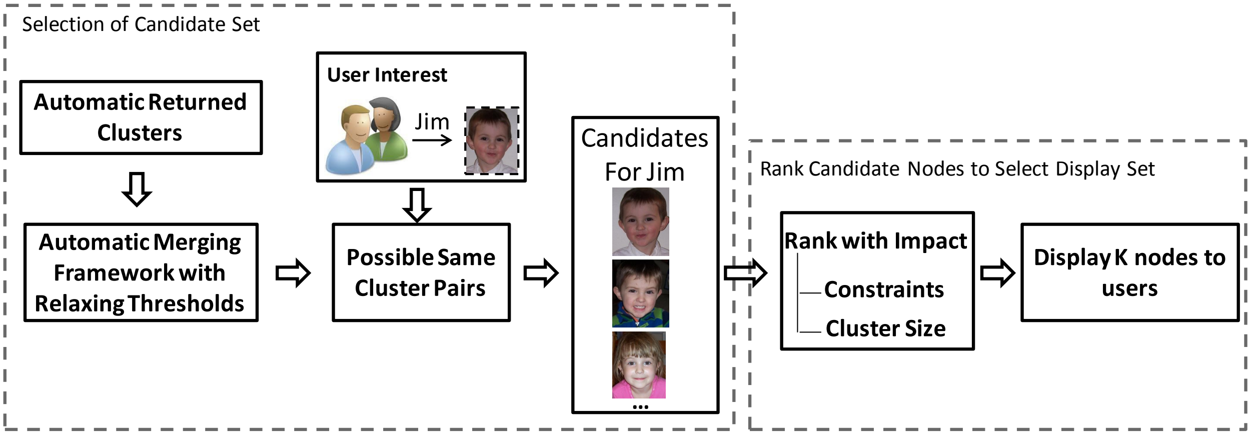

Selection of Candidate Set
Automatic Returned Clusters
Automatic Merging Framework with Relaxing Thresholds
User Interest
Jim
Possible Same Cluster Pairs
Candidates For Jim
...
Rank Candidate Nodes to Select Display Set
Rank with Impact
Constraints
Cluster Size
Display K nodes to users

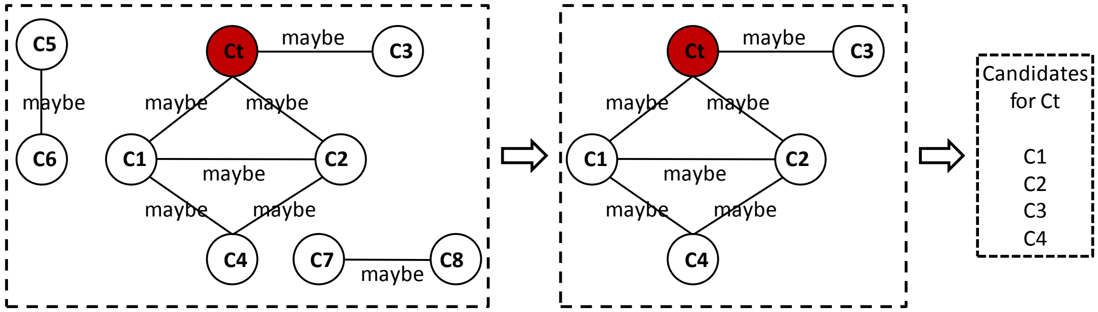
Figure 4.10: 2-Step Framework for Choosing K User Questions.

in the most overall impact. We next describe these two steps in detail.

Step 1: Choosing the Candidate Set of Clusters

Recall that the overall goal of the ranking algorithm is to help the user disambiguate the "target" person quickly with minimal effort. In turn, the task of the first step is to find clusters that are most similar to the target person being disambiguated. For the latter goal, the algorithm leverages the automatic merging process (that have been described in the previous sections) to generate the set of candidate clusters that are possibly the same as the target cluster. Although this pool might contain erroneous clusters, the overall automated framework is known to produce high-quality result and thus it generates the candidate set of high-quality as well.

To generate the candidate set, the algorithm relaxes the threshold and observes which additional clusters would merge with the target cluster (possibly through transitive closure) – these clusters are added to the candidate set. In this process the algorithm also factors in all the constraints available in the system. They include the constraints in the form of "no" answers provided for the clusters by the user in the previous feedback iterations, indicating that certain clusters are not the same.



(a) Possible Co-referred Cluster Pairs **(b) Clusters connected with Ct.**

Figure 4.11: An Example of Candidate Set Construction

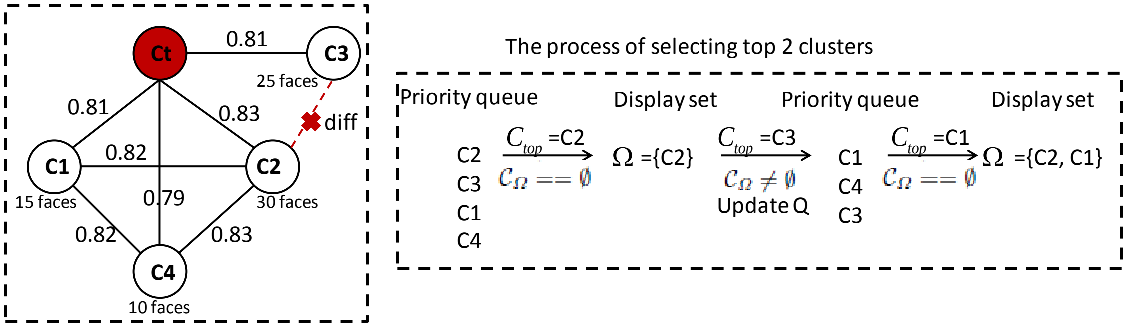


Figure 4.12: An Example of Selecting Top K Clusters from Candidate Set

Figure 4.11 illustrates an example of the process of candidate set construction for the target cluster C_t. The algorithm first uses the automated framework to construct a cluster relationship graph. In this graph, each node corresponds to a cluster C_i, and each edge (C_i, C_j) corresponds to the fact that clusters C_i and C_j will merge if the threshold is relaxed. The candidate pool of clusters are then all the clusters that have either direct or indirect connection to the target cluster C_t, as demonstrated in Figure 4.11(b). Therefore, the candidate set $\mathbb{S}_{cand}$ for C_t is $\mathbb{S}_{cand} = \{C_1, C_2, C_3, C_4\}$.

Then, the goal of Step 2 of the algorithm is to select the set Ω of K nodes from the candidate set $\mathbb{S}_{cand}$ to show to the user for feedback.

Step 2: Ranking Candidate Nodes Based on Impact

Following the above procedure, we have obtained the set of candidate clusters $\mathbb{S}_{cand}$ which potentially co-refer with the target cluster C_t. The next task is to rank them and in order to choose a subset $\Omega \subseteq \mathbb{S}_{cand}$ of K clusters to display to users for feedback. Intuitively, the most similar to C_t clusters should be ranked higher in order for the user to quickly merge clusters that co-refer. This factor has already been taken into consideration in Step 1. Since the goal is to minimize the burden on the user, Step 2 of the algorithm takes into account another factor: the impact of resolving clusters, that is, the amount of the reduction of the uncertainty and the ability to maximize recall.

Augmenting the Graph with Similarity Values and Constraints. To better illustrate the concepts related to the impact metric, we will use an augmented version of the above-defined graph to also encode the similarities and constraints computed by the system. Like in the above-defined graph, in the augmented graph each node also corresponds to a cluster C_i and a presence of an edge (C_i, C_j) corresponds to the fact that (C_i, C_j) would merge by the fully-automated part of the framework if the thresholds are relaxed accordingly. The label of the edge (C_i, C_j) corresponds to the similarity S_{ij} computed by the automated part of the algorithm and reflects the probability that C_i and C_j co-refer. For instance, the similarity between C_t and C_2 in Figure 4.12 is 0.83. Similarly, there is a special label that encode the constraint that two clusters are different, e.g. as for clusters C_2 an C_3 in Figure 4.12. Notice, such a constraint could arise as part of the automated process (see Section 4.2) or it could reflect the "no" answer provided by the user in one of the previous feedback iterations.

Definition of Impact. We will refer to the process of determining whether the given node (cluster) C_i is same/different from the target cluster C_t as *resolving* the node (cluster) C_i.

A node (cluster) C_i can be directly resolved by asking a question to the user on whether C_i and C_t co-refer. Later on we will also see that, in some cases, a node (cluster) C_i could be indirectly resolved without asking the user a question about C_i but rather by factoring in constraints associated with C_i.

Generally, the impact of a node refers to the uncertainty reduction that results from resolving the node. In our case, we can define the impact of one node as the number of uncertain faces that will be resolved (that is, determined whether or not they co-refer with C_t) by providing positive or negative feedback for the node. For example, for two clusters $C_i, C_j \in \mathbb{S}_{cand}$, if C_i and C_j are comparable in their similarity to C_t, but $|C_i| = 100$ (size) whereas $|C_j| = 1$, then the algorithm might prefer to put C_i into Ω over C_j, as disambiguating C_i has the potential to resolve many more faces thus improving the recall quicker.

The size of a node, while important, is not the only factor to consider for computing the impact of resolving the node. The framework should also take into account which faces will be resolved as the result of enforcing the existing constraints. For example, Figure 4.12 encodes the fact that clusters C_2 and C_3 do not co-refer based on the existing constraints, denoted as $D(C_2, C_3) = 1$. Thus, if we resolve C_2 to be, say, "yes" (co-refer with C_t), then C_3 should automatically be "no" (does not co-refer with C_t) due to the constraint and vice versa. Therefore, resolving C_2 or C_3 might lead to more faces to be resolved than just their respective sizes.

Overview of the Naive Exponential Solution. Having defined the impact as the overall number of faces that get resolved as the outcome of the user feedback, we now can outline a naive exponential solution that would choose the optimal subset Ω of K clusters from $\mathbb{S}_{cand}$ to maximize impact. Notice, while we do not know whether the user will answer yes/no for a given cluster $C_i \in \Omega$, we can compute the impact in the expected sense, by assuming the user will answer *yes* with probability S_{it} which can be generated from the automatic procedure.

The algorithm will simply need to enumerate over all different combinations of K clusters from $\mathbb{S}_{cand}$. For each such combination Ω, the algorithm in turn can enumerate all yes/no responses that the user can generates, which should be valid with respect to the constraints. For each possible response to the K questions, the algorithm can compute the impact by computing how many faces will get resolved. The expectation then can be computed by factoring in the probability of yes/no answer for each given cluster in Ω. After that, the algorithm can choose one (optimal) combination Ω of K clusters from $\mathbb{S}_{cand}$ that leads to the maximal expected impact.

Fast Heuristic Solution. While the above naive algorithm will choose the optimal solution, it is clearly exponential and thus impractical. We now will consider a heuristic solution that is not only fast, but also reaches very high quality results.

First, observe that if we were to include only one cluster in Ω, we can compute the impact factor $\mathcal{I}(C_i)$ of each cluster $C_i \in \mathbb{S}_{cand}$ as:

$$\mathcal{I}(C_i) = |C_i| + S_{it} \sum_{C_j : D(C_i, C_j)=1} |C_j| \tag{4.8}$$

That is, after resolving C_i we will know whether or not its $|C_i|$ faces co-refer with C_t, where $|C_i|$ is the number of faces in C_i. In addition, if the user gives the *yes* answer to C_i, then for each cluster C_j such that $D(C_i, C_j) = 1$ we will know that its $|C_j|$ faces do not co-refer to C_t. Eq (4.8) reflects that, given that the probability that the user will give the *yes* answer can be estimated as S_{it}.

Eq (4.8) allows us to rank different nodes based on their impact. For example, if we apply it to rank the clusters in Figure 4.12, the sorted order based on the impact will be $C_2 > C_3 > C_1 > C_4$.

Updating Impact Based on the Already Chosen Clusters Assume that we have already chosen some clusters to be included in Ω and now want to add to it one more cluster C_i from the remaining clusters $\mathbb{S}_{cand} \setminus \Omega$. Notice that we no longer can use Eq. (4.8) to compute the impact of C_i due to presence of constraints. For instance, if $\Omega = \{C_2\}$ then it is incorrect to compute the impact of adding C_3 to Ω using Eq. (4.8) – as for instance with probability 0.83 the user will say *yes* to C_2 and thus automatically resolving C_3 to *no* due to $D(C_2, C_3) = 1$ constraint. Using the process identified in the above naive solution will can compute the expected impact, however that computation will be exponential. Instead, we will use heuristic solution that will estimate the impact $\mathcal{I}(C_i|\Omega)$ of C_i given that certain clusters have already been chosen to be in Ω.

For instance, consider computing the impact $\mathcal{I}(C_3|\Omega)$ of C_3 from Figure 4.12 given that C_2 has already been added to Ω, that is, $\Omega = \{C_2\}$. It can be estimated as $\mathcal{I}(C_3|\Omega = \{C_2\}) = (1 - S_{2t})|C_3|$, because resolving C_2 with probability of S_{2t} will cause a resolution of C_3 and resolving C_3 will only cause a resolution of additional $|C_3|$ faces of the anwser for C_2 was *no*.

In general, we can estimate $\mathcal{I}(C_i|\Omega)$ as follows. Let $\mathcal{C}_\Omega$ be the set of clusters from the display set Ω that conflict with C_i, that is, $\mathcal{C}_\Omega = \{C_j : C_j \in \Omega \wedge D(C_i, C_j) = 1\}$. Similarly, let $\mathcal{C}_S$ be the set of the remaining clusters from $\mathbb{S}_{cand} \setminus \Omega$ that conflict with C_i, that is, $\mathcal{C}_S = \{C_j : C_j \in \mathbb{S}_{cand} \setminus \Omega \wedge D(C_i, C_j) = 1\}$. Both of these sets could be empty. Resolving a cluster C_i will create an additional $|C_i|$ impact only if the user answers to all clusters in $\mathcal{C}_\Omega$ as *no*. The probability that all these answers are *no* answers can be computed as $P_\Omega = \prod_{S_j \in \mathcal{C}_\Omega}(1 - S_j)$. In the special case where $\mathcal{C}_\Omega = \emptyset$, we set P_Ω as $P_\Omega = 1$. If the user answers to all clusters in $\mathcal{C}_\Omega$ as *no* and then she identifies C_i as a *yes* (which should happen with probability S_{it}), then additional conflicting clusters from $\mathcal{C}_S$ will be automatically labeled as *no*, thus contributing additional $\sum_{S_j \in \mathcal{C}_S} |S_j|$ faces to the impact. Overall, we get:

$$\mathcal{I}(C_i|\Omega) = P_\Omega \left(|C_i| + S_{it} \sum_{S_j \in \mathcal{C}_S} \right) \tag{4.9}$$

We can see that Eq. (4.9) is a generalization of Eq. (4.8). Specifically, Eq. (4.8) is equivalent to Eq. (4.9) wherein $\Omega = \emptyset$, that is, no clusters have been yet added to Ω.

Algorithm 1: The Greedy Algorithm of Selecting Top K Clusters

```
    input : Target cluster $C_t$; Candidate set $\mathbb{S}_{cand}$; Parameter $K$
    output: Display set $\Omega$, where $|\Omega| = K$
 1  $\Omega \leftarrow \emptyset$                              // Initialize the display set
 2  $Q \leftarrow \emptyset$                                  // Initialize the priority queue
 3  foreach node $C_i \in \mathbb{S}_{cand}$ do
 4      Compute $\mathcal{I}(C_i)$                          // Use Eq. (4.9) to compute impact
 5      $Q.insert(C_i, \mathcal{I}(C_i))$         // Insert $C_i$ into $Q$ ranked by impact $\mathcal{I}(C_i)$
 6  while $|\Omega| < K$ and $|Q| > 0$ do
 7      $C_{top} \leftarrow Q.pop()$                      // Get the first top element of $Q$
 8      $\mathcal{C}_\Omega \leftarrow \{C_j : C_j \in \Omega \wedge D(C_{top}, C_j) = 1\}$
 9      if $\mathcal{C}_\Omega = \emptyset$ then                 // $C_{top}$ doesn't conflict with $\Omega$
10          $\Omega \leftarrow \Omega \cup \{C_{top}\}$                           // Add $C_{top}$ to $\Omega$
11          $\mathbb{S}_{cand} \leftarrow \mathbb{S}_{cand} \setminus \{C_{top}\}$
12      else
13          Compute $\mathcal{I}(C_{top})$
14          $Q.insert(C_{top}, \mathcal{I}(C_{top}))$                  // Reinsert $C_{top}$ to $Q$
15  return $\Omega$
```

Greedy Algorithm Using Eq. (4.9) we can design a greedy algorithm whose pseudocode is shown in Algorithm 2. The algorithm starts with the empty display set $\Omega = \emptyset$ (Step 1); eventually it will greedily add K clusters to Ω one by one. Initially all clusters in the candidate set $\mathbb{S}_{cand}$ are ranked by using Eq. (4.9) (Step 4) and then inserted into the priority queue Q (Step 5). To choose the next element to add to Ω, the algorithm pops the top element/cluster C_{top} from Q (Step 7). The algorithm then checks whether the top ranked node C_{top} needs to be updated (Step 9). If not, then it adds C_{top} to Ω and either proceeds back to Step 7 to retrieve the next node from the priority queue, or stops if $|\Omega| = K$. If it needs to recompute the rank of C_i, it does so (Step 13), reinserts C_{top} back into the priority queue (Step 14), and proceeds back to Step 7 to retrieve the next node with the best rank.

For instance, for our running example in Figure 4.12 the algorithm will work as follows. The input of the algorithm is the candidate set $\mathbb{S}_{cand} = \{C_1, C_2, C_3, C_4\}$, parameter $K = 2$, and target node C_t. The algorithm starts with display set $\Omega = \emptyset$, and eventually it will add two clusters to Ω. Initially, all the candidate nodes are ranked based on impact $\mathcal{I}(C_i)$ and inserted into priority queue $Q = C_2 > C_3 > C_1 > C_4$. Then the algorithm will popup the top element from Q and set $C_{top} = C_2$. Since the current display set $\Omega = \emptyset$, then we compute $\mathcal{C}_\Omega = \emptyset$. Therefore, we directly add C_2 into display set $\Omega = \{C_2\}$. On the next iteration the algorithm will repeat Step 7 and will set $C_{top} = C_3$. Then it will determine that $\mathcal{C}_\Omega = \{C_2\}$ due to $D(C_2, C_3) = 1$. Since $\mathcal{C}_\Omega \neq \emptyset$, we need to re-compute the impact $\mathcal{I}(C_3)$ of C_3, and reinsert C_3 into the priority queue, which at this point will become $Q = C_1 > C_4 > C_3$. On the next iteration the algorithm will the first top element from Q, which is going to be $C_{top} = C_1$. It will next determine that $\mathcal{C}_\Omega = \emptyset$ for C_1, so it will add C_1 to Ω. At this point, $|\Omega| = 2$, so the algorithm will stop and return the display set $\Omega = \{C_2, C_1\}$.

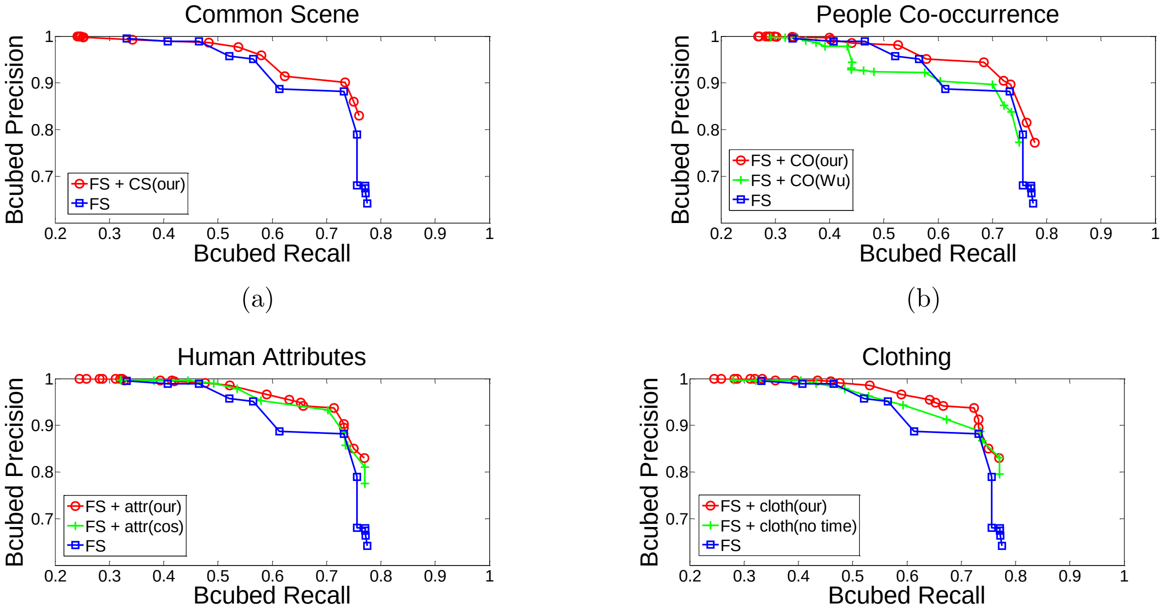
4.5 Experimental Evaluation

In this section, we evaluate our algorithm on three human-centered data collections: Gallagher, Wedding, and Surveillance.[2] The characteristics of these datasets are listed in Table 4.1. Gallagher[33] is a public family album containing photos of three children, other family members and their friends. The wedding dataset has been downloaded from Web Picasa. It captures people in a wedding ceremony, including the bride, the groom, their

[2] We note that while larger dataset exists, (e.g., LFW, PubFig), these datasets (LFW and PubFig) are not suitable for our work because they only provide single face rather than the whole image, whereas we focus on disambiguating faces in a photo collection.

Dataset	#Images	#Faces	#People	Image Pixels
Gallagher	591	1064	37	2576 × 1716
Wedding	643	1433	31	400 × 267
Surveillance	1030	70	45	704 × 480

Table 4.1: Experimental Dataset



(a)

(b)

(c)

(d)

Figure 4.13: Effectiveness of Extracted Context Features

relatives and friends. The surveillance dataset contains images that capture the daily life of faculty and students in the 2nd floor of a computer science building.

To evaluate the performance of the proposed approach, we use B-cubed precision and recall defined in Eqs. (1) and (2) as the evaluation metrics. First, we run some experiments to demonstrate the importance of using different context feature types. Then we compare our clustering results with those obtained by Picasa and affinity propagation[31] algorithms, to illustrate the overall effectiveness of our unified framework.

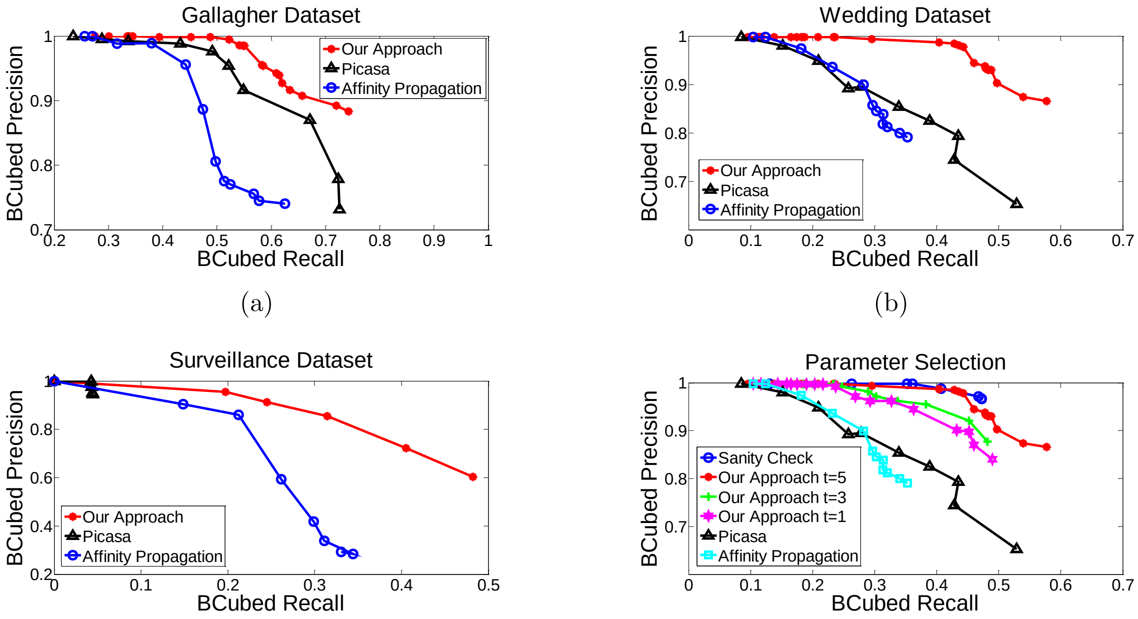
4.5.1 Context Feature Comparison

As shown in Figure4.13, a series of experiments are performed on the Gallagher dataset to test the effectiveness of the proposed 4 types of context similarities. Each plot in Figure 4.13 corresponds to one type of context similarity. Each plot compares the baseline algorithm that uses only face similarity (denoted as FS) with our framework which is allowed to use just one given context feature type instead of all 4 types.

Figure 4.13(a) illustrates that the clustering performance can be improved by combining common scene (CS) feature with facial similarities (FS). The improvement is not very significant because only 50 cluster pairs are linked by common sense feature in Gallagher dataset. Figure 4.13(b) shows the comparison between our approach (CO(our)) and Wu's approach (CO(Wu))[86] in leveraging people co-occurrence feature. The performance of our approach is much better than Wu's approach because we use the cluster groups as evidence to link two clusters, which is more reliable than the linkage of single cluster. Figure 4.13(c) demonstrates that our approach (attr(our)) outperforms the cosine similarity measurements (attr(cos)) in using human attributes feature. The advantage of our approach is because we automatically learn the relative importance of various attribute types in identifying different people. Figure 4.13(d) shows the advantage of our approach (cloth(our)) compared with the approach without considering time factor (cloth(no time)) in utilizing clothing information. This demonstrates the advantage of adding time decay factor to clothing information.

4.5.2 Automatic Clustering Results Comparison

To evaluate the performance of the proposed unified framework, we compare our clustering results with affinity propagation (AP)[31] and Picasa's face clustering toolkit, as shown in Figure 4.14. Four types of context information and facial visual similarities are integrated into our framework. B-cubed precision and recall are computed as the evaluation metrics. In



(a)

(b)

(c)

(d)

Figure 4.14: Comparison of Clustering Performance with Affinity Propagation and Picasa on Three Datasets (see Figure (a)(b)(c)). Figure(d) Comparison of Different Parameters.

our clustering framework, several parameters need to be selected, the split cluster size (sz), and number of times to perform "splitting-training-predicting" process t. In this experiment we set $sz = 10$ and $t = 5$.

When performing affinity propagation, we combine the 4 types of context features with equal weight, and then aggregate context features and facial feature with equal weight to construct the overall similarity. By adjusting the preference parameter p in AP, we are able to control the precision vs. recall tradeoff for AP. Picasa allows users to specify the cluster threshold (from 50 to 95) to control the precision vs. recall tradeoff. With the increasing of threshold, the recall reduces and the precision increases.

As demonstrated in Figure 4.14, our unified framework outperforms Affinity Propagation (AP) and Picasa in all the three datasets. The gained advantage is due to leveraging the bootstrapping idea to automatically learn and tune the merging rules per each dataset, and due to using the conservative merging strategy that guarantees the high precision. In addition, our framework is more reliable for data with lower quality images because the context features are less sensitive to image resolution. This is not the case for Picasa, as its performance drops dramatically with the decreasing of image quality. The experiments illustrate that our unified framework reaches high quality and at the same time is more reliable than the other two techniques.

Effectiveness and Efficiency

Figure 4.14(d) shows the comparison of clustering results when choosing different values for parameter t (the number of times to perform "splitting-training-predicting" process). The larger t can provide more reliable clustering results because it can reduce the uncertainties introduced by random splitting. However, The larger t will reduce the efficiency of the algorithm. The experiments illustrate that when $t = 5$, our performance is approaching the "sanity check" results (merging rules learned from ground truth). And when $t = 1$, our results are still better than affinity propagation and Picasa. Thus our approach is able to achieve a good result without sacrificing efficiency.

4.5.3 Human-in-the-loop

In this part, we perform a series of experiments to evaluate the effectiveness of the proposed approaches for leveraging human-in-the-loop methodology to improve data quality. As explained in Section 4.4, the user interface allows the user to choose the target cluster C_t and then displays the unlabeled clusters in a ranking order with respect to the chosen C_t. On each iteration, the interface displays K clusters to the user. On each iteration, after receiving user feedback, the algorithm updates the cluster status by merging the clusters answered "yes" with the target cluster C_t, and by labeling those answered "no" as "diff" ones. Then it re-ranks the unlabeled clusters and choose the next K clusters to display. To simulate the user behavior, we assume the user will select 10 largest groups (10 people with the largest number of photos) as target clusters, and then use the average results to evaluate the performance.

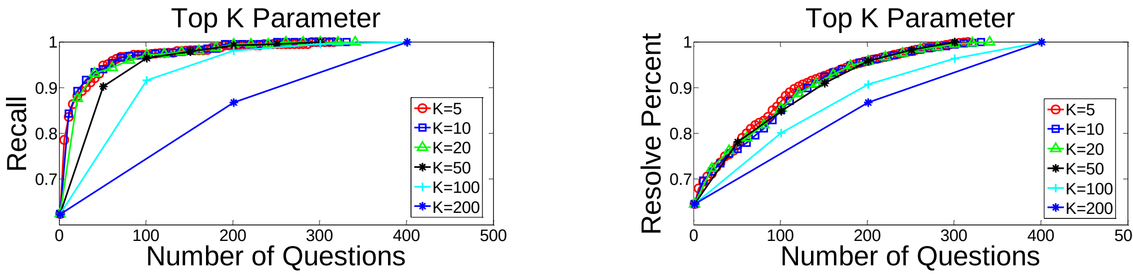
Evaluation Metrics

Generally, a good ranking strategy should be able to facilitate the user to quickly label all the related data with minimal effort. Therefore, we propose to evaluate our approach from two aspects: (1) how quickly the user can label all relevant to C_t faces; and (2) how fast the user can resolve all the uncertain (unlabeled) data in general. For this goal, we employ two evaluation metrics:

- **Recall.** Given a target cluster C_t, recall is defined as
  $Recall(C_t) = \frac{size(C_t)}{\|\{f_i|L(f_i)=L(C_t)\}\|}$,
  where $L(f_i)$ refers to the labeled ground truth for face f_i. Recall measures the proportion of correctly related faces compared with all the faces which refer to the specified entity.

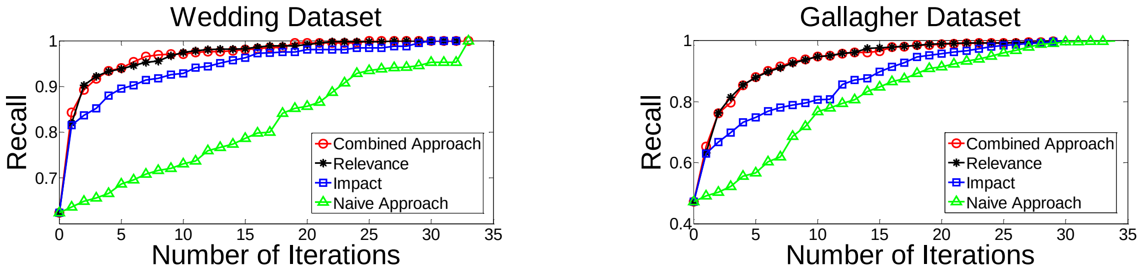
- **Face resolve ratio.** Face resolve ratio is defined as
  $Resolve(C_t) = \frac{size(C_t)+\sum_{D(C_t,C_l)=1} size(C_l)}{\|\{f_i\}\|}$,
  where $D(C_t, C_l) = 1$ denotes that we confirm C_t and C_l can not co-refer. It measures the proportion of resolved faces compared with all the faces in the dataset.

On each iteration, we compute recall and face resolve ratio, and then plot the improvement of them respectively with the number of iterations increasing. A good ranking strategy should be able to achieve the fast improvement for both recall and resolve ratio with the increase of iteration number.



(a) (b)

Figure 4.15: Comparison of Results with Different Parameter Setting

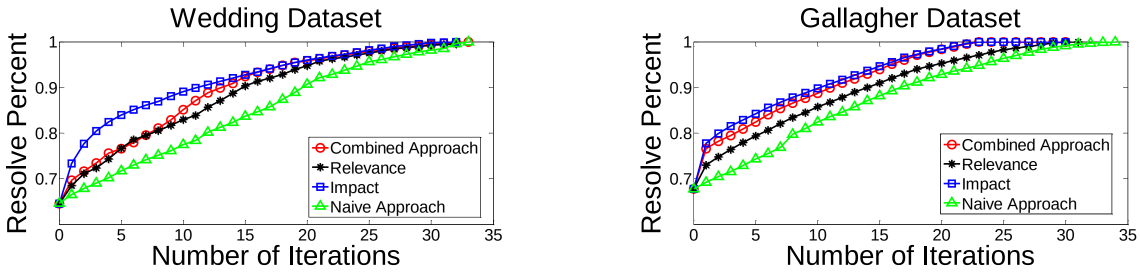


(a) (b)

Figure 4.16: Recall Improvement with Iterations Increasing

Parameter Selection

An interesting question to study is how to choose the size K of the display set Ω: that is, how many clusters to show to the user at once on each iteration. To choose the appropriate value for parameter K, we select different values (from 5 to 200) and plot the tendency of recall and resolved rate. As illustrated in Figure 4.15, we discover that the larger values of K lead to the slower improvement of data quality, and more questions required to resolve all the uncertain data. This is because if we choose a larger value for K, the cluster status cannot be updated promptly, and some unnecessary questions will be asked. However, a small K value will result in too many iterations and computation load. Therefore, to trade off between the two factors, we set $K = 10$ in our framework.



(a) (b)

Figure 4.17: Face Resolved Percentage Improvement with Iterations Increasing

Results Comparison and Analysis

In Section 4.4, we propose that the uncertain data should be sorted by considering two ranking metrics (relevance and impact), and we also present a combined approach that takes into consideration both of the two factors. To evaluate the effectiveness of the proposed approaches, we run experiments with several different ranking strategies as follows.

- *Naive approach*: sort the uncertain clusters in a random order.
- *Relevance based approach*: sort all the uncertain clusters based on their relevance (similarities) to the target cluster, without considering the impact of clusters. This method tends to rank the most relevant clusters higher.
- *Impact based approach*: sort all the uncertain clusters based on impact, without considering the relevance. This method does not include the candidate set selection process.
- *Combined approach*: sort the uncertain clusters considering both relevance and impact. As described in Section 4.4, this approach consists of two components, selection of candidate set based on relevance and ranking candidates based on impact.

Figure 4.16 and Figure 4.17 illustrate the comparison between the above strategies on the two photo collections. Figure 4.16 demonstrates the tendency of recall and Figure 4.17

demonstrates that of face resolve ratio, with the increase of iteration number. From the recall plots we can see that the proposed strategies allow to very quickly, in just a few feedback iterations, disambiguating most of the faces for the target cluster as compared to the naive random strategy. Analyzing the overall results, we discover that the relevance-based approach can achieve better performance on recall but lower performance on the resolve ratio. This is because this method aims to put the most relevant clusters first, which facilitate the improvement of recall; however, it does not consider the impact of clusters, which leads to the lower improvement of face resolve ratio. In contrast, the impact-based approach can achieve better performance on the face resolve ratio but lower performance on the recall. That is because this method tends to put the clusters with larger impact first, without the consideration of relevance, which might lead to some irrelevant clusters with larger size ranked higher. Compared with the two methods, the combined approach is able to achieve better performance in both recall and face resolve ratio, because it considers both relevance and impact by consisting of two steps, the selection of candidate set based on relevance and the ranking of candidate nodes based on impact. In addition, we discover that all the three strategies (relevance, impact and combined approaches) can achieve much better performance than the naive approach on both recall and face resolve ratio. Therefore, the comparison in Figure 4.16 and Figure 4.17 demonstrates the effectiveness of the proposed ranking metrics (relevance and impact), and also illustrates the superiority of the combined approach in the capability of improving both recall and face resolve ratio. The results imply that the combined approach not only helps the users to effectively disambiguate clusters of interest, but also facilitates quicker reduction in uncertainty in the entire collection.

4.6 Chapter Conclusion and Future Work

In this chapter we have proposed a unified framework for integrating heterogeneous context information (including common scene, people co-occurrence, human attributes and clothing) to improve the quality/recall of face clustering. The context information has been used for both: computing context similarities to link clusters that co-refer as well as for generating context constraints to differentiate clusters that do not co-refer. The proposed unified framework leverages bootstrapping to automatically learn the adaptive rules to integrate heterogeneous context information together to iteratively merge clusters, in order to improve the recall of clustering results. Finally, we discuss several methods to integrate human-in-the-loop in order to achieve very high-quality clustering results. Our experiments on the real-world datasets demonstrated the effectiveness of the extracted context features and of the overall unified framework.

It should be noted that no system is perfect and several conditions might decrease the accuracy of our system. One particular case is when only the face is provided instead of the entire image, and thus several types of contextual information simply cannot be leveraged by our approach. The unavailability of context features will influence the effectiveness of our system. Another challenging problem is to deal with the unreliable and noisy context features. We need to make sure that the automatically extracted context information is relatively reliable; otherwise, too much noise will impact the clustering performance although our system adopts very conservative strategy to make the merge decisions. In addition, some assumptions are made in our system, such as "faces co-occurred in the same image can not refer to the same person". However, in the real world, exceptions might exist, for example, images processed by some software (e.g., Photoshop). Under these conditions, the performance of our system will be impacted and techniques should be developed to detect doctored images. In general, to handle the aforementioned issues, one of the possible solutions is to leverage the help from users. In this chapter, several simple strategies to integrate human-in-the-loop techniques have been discussed. In the future, we plan to explore the human-in-the-loop strategies from a formal probabilistic perspective and in the context of answering SQL-style user queries.

Chapter 5

Query-driven Approach to Face Clustering and Tagging

In this chapter, we study *the problem of query-driven approaches to face tagging.*

The era of “big data” is certainly upon us. Recent estimates suggest that by 2018, more than 80% of total traffic on the internet will consist of video transmissions, and more than 50% of total traffic will originate from non-PCs (e.g., mobile) devices [2]. These statistics suggest that big multimedia data from heterogeneous sources will play a vital role in the future, supporting a variety of end applications. We see three main challenges for visual processing in the big-data era:

1. **Variability:** Diverse data sources mean that the data will arrive in a noisy, unclean state, making data-cleaning a fundamental issue [38].
2. **Velocity:** Data and applications will appear at fast, essentially real-time rates [2]. Data must be processed as it arrives, but we may not know the appropriate *schema* for applications that appear in the future.

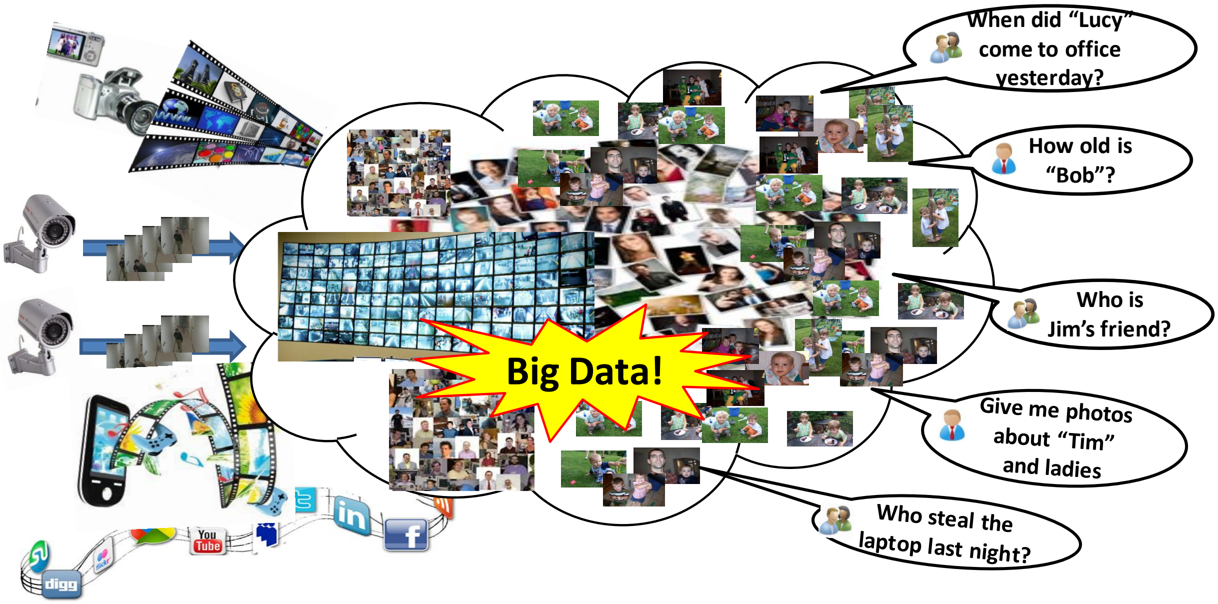

When did “Lucy” come to office yesterday?
How old is “Bob”?
Who is Jim’s friend?
Big Data!
Give me photos about “Tim” and ladies
Who steal the laptop last night?

Figure 5.1: In the era of big data, the “offline” setting for face clustering/tagging is not suitable. We propose query-driven paradigm to face clustering/tagging which can be seamlessly integrated into image analysis/retrieval process.

3. **Volume:** We simply will not have the computational resources to process all the data with all possibly relevant attribute tags. It will be crucial to not spend computation on processing that will not be used by a future application.

To make such issues concrete, we specifically focus on face-tagging and clustering, where the schema encodes attributes and relational correspondences for faces present in an image database. Such a schema allows for application queries such as “retrieve images of John”, “retrieve images of a person shown in the given query image”, or even “who appears most frequently in pictures with Bob?”. Importantly, we consider an **open-world** setting where the set of potential people are not known *a priori*, and where we have access to vision models that produce **noisy** identity and relational tags.

Related work: Face clustering and tagging have been incorporated in many commercial photo management systems such as Google’s Picasa and Apple iPhoto. Most such systems offer semi-automated techniques – the system performs an initial clustering based on various features, the result of which are returned to the users for cluster refinement and tagging. The academic community has also pursued similar methods for interactive or "human-in-the-loop" face tagging [79, 85, 27, 9], sometimes addressed in an active-learning framework [51, 50]. Notably, such methods, though interactive, are still applied in an "offline" setting assuming one has access to all the data and all tags of interest. Finally, several pieces of related work focus on incremental model updating [56], sometimes in a tagging context [22]. Rather than efficient model updating, we focus on efficient user queries.

Our approach: We introduce an approach based on "query-driven image enrichment". Essentially, we delay cleaning/processing the data until an application-specific query requires it. This way, both computer and human computation are spent in the service of a particular query, and so are never wasted. To implement this strategy, we introduce a notion of query-driven active learning. Rather than asking a human to provide a label that minimizes label uncertainty over all the data, *our system asks for a label that reduces the uncertainty in answers for particular query.* For example, if a user queries the system for images of "John", our system will likely interactively prompt the user to provide tags for John or face that get confused with John. We show that this produces much different results than "query-agnostic" prompts used in active learning.

Database-perspective: This work is written from an unabashedly database perspective. As the computer vision field moves into the era of big-data, we argue problems should be viewed from the database perspective. We refer the reader to established texts on databases [68], but in a nutshell, database management systems use appropriate data structures for accessing main memory versus the hard disk (such as b-trees [55]), support complex queries though a formal query language (such a relational calculus [24]), support concurrent queries from multiple users (through transactions [59]), and finally maintain data fidelity through logging. Large-scale image and video databases must support these features as well. To do so, *we describe our models and approach through the language of probabilistic relational*

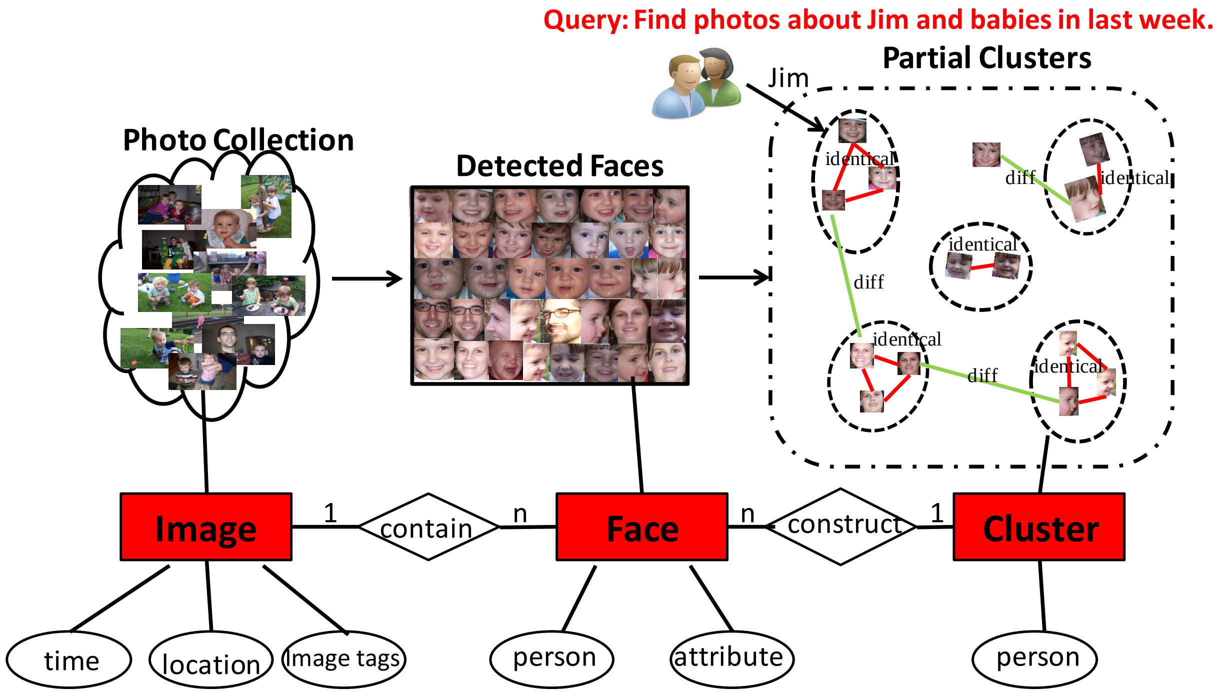


Figure 5.2: The input of query-driven face tagging problem. We visualize our database schema with a standard entity relationship diagram[68] consisting of entities (squares), attributes (ovals), and relations (diamonds). Such a schema can be efficiently implemented in a relational database.

database and entity-relationship schema.

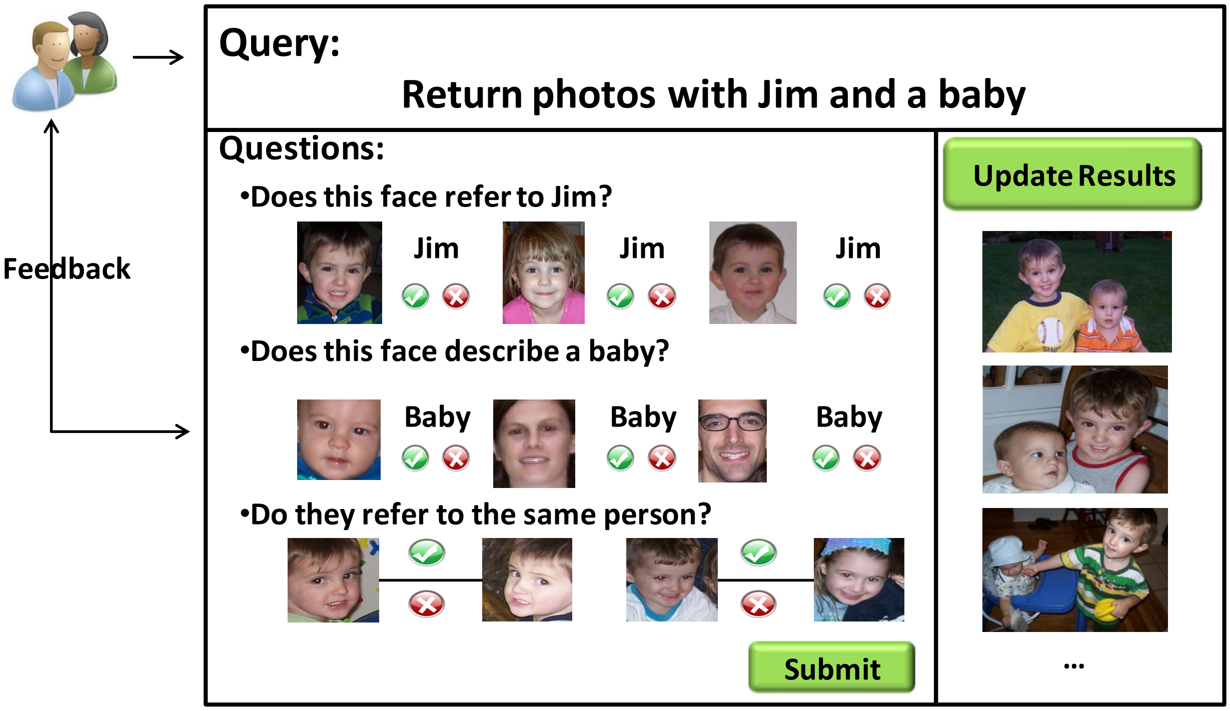
The rest of this chapter is organized as follows. We first define the schema for our database through entity relational diagrams in Section 5.1. In Section 5.2, we present the main approach to query-driven face tagging, beginning with probabilistic queries on the media data and then exploring strategies for interactive face tagging. The proposed approach is empirically evaluated in Section 5.3, specifically compared with previous approaches to query-agnostic active learning. Finally, we conclude in Section 5.4 by highlighting key points of our work.

5.1 Schema definition

We begin by describing our database schema. Suppose that we are given a human-centered photo album that contains M images $\{I_1, I_2, \ldots, I_M\}$, see Figure 5.2. Assume that n faces are detected, $F = \{f_i\}_{i=1}^n$, with each face denoted as f_i or $f_i^{I_k}$ (that is, f_i is extracted from image I_k). Suppose that middle-level semantic concepts can be extracted from images and faces leveraging the pre-trained classifiers or provided by users, such as image captured time denoted as $I_k.time$, Geo-location $I_k.loc$, and images tags $I_k.tag$. Besides, face attributes can be extracted from each face utilizing the pre-trained attribute extraction classifiers, including intrinsic attributes like "gender", "age", "ethnicity", and describable attributes "black hair", "big nose", etc., denoted as $f_i.attr$. Each face is associated with a identity attribute $f_i.t$. Importantly, the domain of $f_i.t$ is unknown (because of our open-world assumption).

Database constraints: Contextual constraints provide additional cues about the face identities. For example, co-occurring faces in one image usually refer to different people. Such a relationship can be defined as a "diff" constraint ε^- in our schema. Faces from the same face track in a video should refer to the same person, denoted as a "identical" (or "same") constraint ε^+. In our experiments, we generate a over-clustering of faces (with a tight threshold on appearance variation) to construct a set of N initial clusters $\{C_1, C_2, \ldots, C_N\}$. Assuming that each cluster is pure, we enforce the same ε^+ constraint for each cluster.

Weak supervision: We construct a probabilistic database using the entity-relationship diagram in Figure 5.2. On top of this probabilistic database, users can present a query to extract the interested knowledge. To indicate the target people in the query, users can specify several face samples of this targets. Therefore, the whole face dataset F will be partitioned into labeled face set $F_L = \{f_1, f_2, \ldots f_L\}$, with tags $T_L = \{t_1, t_2, \ldots t_L\}$, and unlabeled face set F_U. Our goal is to choose which faces to tag (or questions to ask users) in order to achieve the accurate query answers as soon as possible.



Query:

Return photos with Jim and a baby

Questions:

- Does this face refer to Jim?
- Does this face describe a baby?
- Do they refer to the same person?

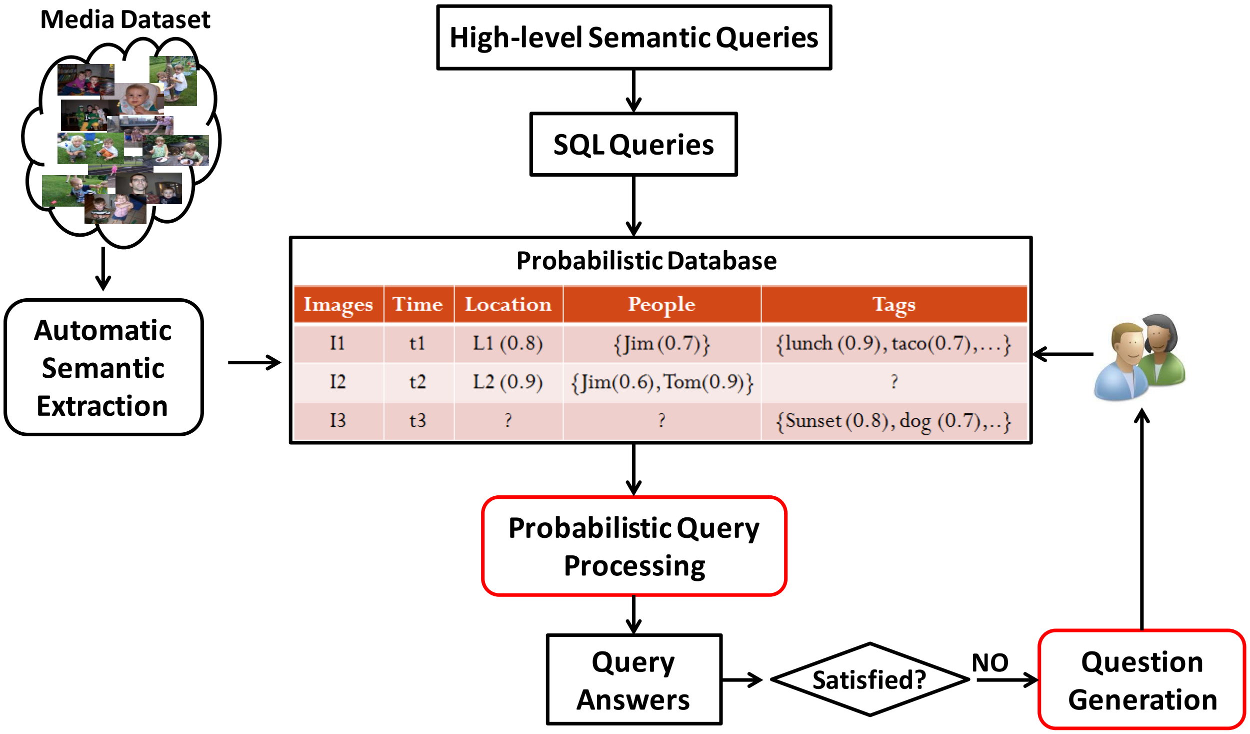
Figure 5.3: User Interface. Given a query, the system will automatically choose some questions to return to users for feedbacks, based on which the query answers will be updated.

5.2 Query-driven approach to Face Tagging

Querying: One advantage of a database is immediate support for powerful operators such as selections "show me all male faces", complex selections such as (e.g., "male faces with Jack") , aggregations ("who appeared most often with Jim"), and joins ("all pictures of a person who appeared with Jim"), SQL allows one to algebraically compose fairly complex queries by composing the basic operators such as selections and joins.

Figure 5.3 illustrates the designed interface, taking the above query as an example. Query answers are displayed to users as a ranking list based on the relevance to queries. To measure the quality of query answers, we choose to leverage the metrics of average precision (AP) [92] to evaluate the returned ranking list. Our goal is to improve the AP performance of the returned ranking list with the minimum user participation.

Interactive tagging: In this chapter, we concentrate on the query-driven face tagging



Images	Time	Location	People	Tags
I1	t1	L1 (0.8)	{Jim (0.7)}	{lunch (0.9), taco(0.7),...}
I2	t2	L2 (0.9)	{Jim(0.6), Tom(0.9)}	?
I3	t3	?	?	{Sunset (0.8), dog (0.7),..}

Figure 5.4: The general framework of query-driven face tagging. Queries are processed on a probabilistic database to generate the tentative answers. And then the query-driven active learning strategy will select questions to return to users for feedback, which will be leveraged to update the query answers.

problem. Compared with the conventional approaches to interactive processing, our goal is to improve the query answer rather than reduce the annotation uncertainty across the entire dataset. As illustrated in Figure 5.4, given a media dataset, we first built a probabilistic database by extracting semantic attributes using visual concept detectors (for example, tuned for faces and particular face attributes). We expect these detectors (and the resulting attributes) to be rather noisy and probabilistically uncertain. When users present a high-level semantic query (translated to SQL query algebra), the database manager will process the query and return an answer to the query. If users are not satisfied with the answer, the human-in-the-loop component will be activated. It will automatically generate questions to ask users for feedback, based on which the final query answer will be updated until users are satisfied.

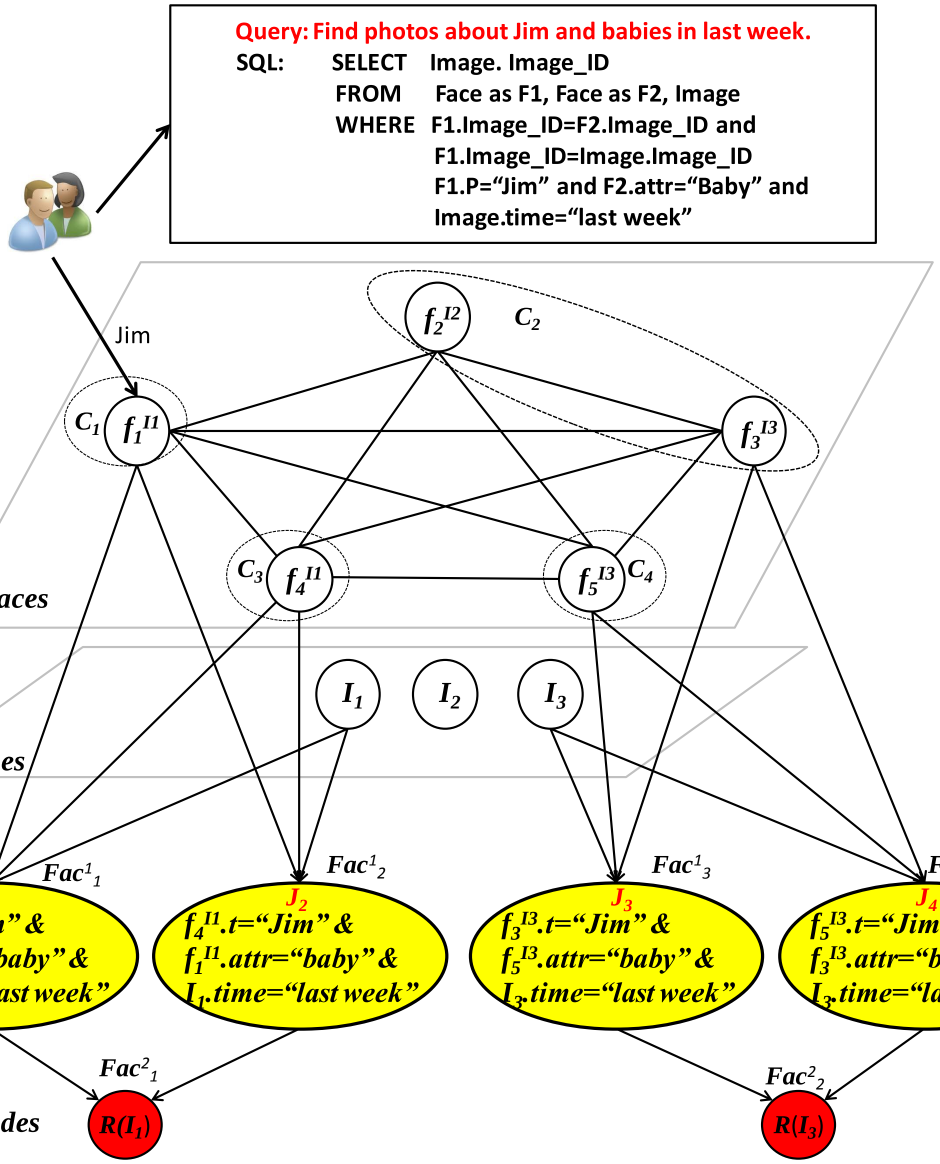
In the following, we begin in Section 5.2.1 by presenting how to process the queries on the probabilistic database. Then we proceed in Section 5.2.2 to explore the appropriate strategy to generate the question orders for feedback in order to maximally improve the quality of query answers.

5.2.1 Probabilistic Query Processing

Capturing probabilistic uncertainties is still somewhat challenging in a database model. One of more successful approaches implements a probabilistic graphics model (PGM) through a database system [72]. Sen et al. describe a relational data encoding of factor graphs that can leverage probability inference techniques to compute query results. Importantly, users can produce queries using a standard set algebra including selection, join, aggregation, etc. Sen et al describe a mapping for transforming such queries into a factor graph. In this chapter, we choose to focus on "select" queries – e.g., "find the photos of Jim and babies in the last week". There are several reasons – first, before we explore more complex nature of queries – e.g., joins, aggregations, or a full class of SQL, it makes sense to develop and validate the idea in the context of selections. Furthermore, selections, while limited in their degree of expressibility are arguably the most important class of queries in the context of visual data collections. Finally, as we will discuss later in the chapter, techniques for more complex queries can be built around the techniques we develop for selections.

Factor Graph Representation

We visualize a small example factor-graph in Figure 5.5. Note that this represents a probabilistic factor graph with random variables, and not an entity-relationship diagram. Specifically, the factor graph includes random variables capturing the uncertain identity label of each face, and possibly uncertain tags associated with each image. The factor-graph en-



Query: Find photos about Jim and babies in last week.

```
SQL:    SELECT   Image. Image_ID
        FROM     Face as F1, Face as F2, Image
        WHERE    F1.Image_ID=F2.Image_ID and
                 F1.Image_ID=Image.Image_ID
                 F1.P="Jim" and F2.attr="Baby" and
                 Image.time="last week"
```

Jim
C_1
f_1^{I1}
f_2^{I2}
C_2
f_3^{I3}
C_3
f_4^{I1}
f_5^{I3}
C_4
Faces
I_1
I_2
I_3
Images
Join tuples
Fac^1_1
Fac^1_2
Fac^1_3
Fac^1_4
J_1
f_1^{I1}.t="Jim" & f_4^{I1}.attr="baby" & I_1.time="last week"
J_2
f_4^{I1}.t="Jim" & f_1^{I1}.attr="baby" & I_1.time="last week"
J_3
f_3^{I3}.t="Jim" & f_5^{I3}.attr="baby" & I_3.time="last week"
J_4
f_5^{I3}.t="Jim" & f_3^{I3}.attr="baby" & I_3.time="last week"
Fac^2_1
Fac^2_2
Result nodes
$R(I_1)$
$R(I_3)$

Figure 5.5: An Example of Factor Graph. A given query is first transformed into SQL query, and then represented by factor graph. Correlations and dependencies are captured by factor graph to facilitate the probabilistic query reasoning.

coding [72] enumerates combinations of entity relationships with a “join operation”, and associates each join-tuple with a binary random variable indicating if it is true or not.

Dependencies: The edges in the graph can represent the correlations and dependencies between entities. Faces are linked to each other with undirected edges indicating appearance-based similarity. Directed edges represent relationship dependencies between entities. For example, join tuples depend on the faces and images, where dependability is defined by factor functions Fac_i^1, e.g., $Fac_1^1(J_1|f_1^{I_1}, f_4^{I_1}, I_1) = true$, if $(f_1^{I_1}.t =$ “Jim” $\wedge f_1^{I_4}.attr =$ “$baby$” $\wedge I_1.time =$ “$lastweek$”$)$. Thus, the probability of join tuple $J_1 = true$ is the joint probability of each of its dependent factors taking on the given states.

Result nodes: We define a result node associated with each image in the database, capturing a binary variable that specifies if it should be returned in the query answer. Result nodes rely on join tuple nodes, with the factor denoted as Fac_i^2, $Fac_1^2(R(I_1)|J_1, J_2) = 1$, if $(J_1 = true \vee J_2 = true)$. To infer the probability of result node r_i to be true denoted as $p(r_i)$, we need to first induce the posterior probability of the unlabeled face samples.

Inference on unlabeled data

We now describe an appearance model for generating uncertain face tags. Given a database-perspective, it is natural to use data-driven nonparametric appearance models to make predictions on unlabeled data. Gaussian processes elegantly combine the flexibility of data-driven nearest-neighbor methods with the regularization afforded through smooth parametric functional models.

For simplicity, let us write the person identity attribute of a face f_i as $t_i = f_i.t$. Assume that the user has provided a collection of tags. Based on ”same/diff” constraints, the database manager can spread these constraints to resolve the label of some other faces. This will partition the entity set of faces F into a labeled face set $F_L = \{f_1, f_2, \dots f_L\}$, with tags

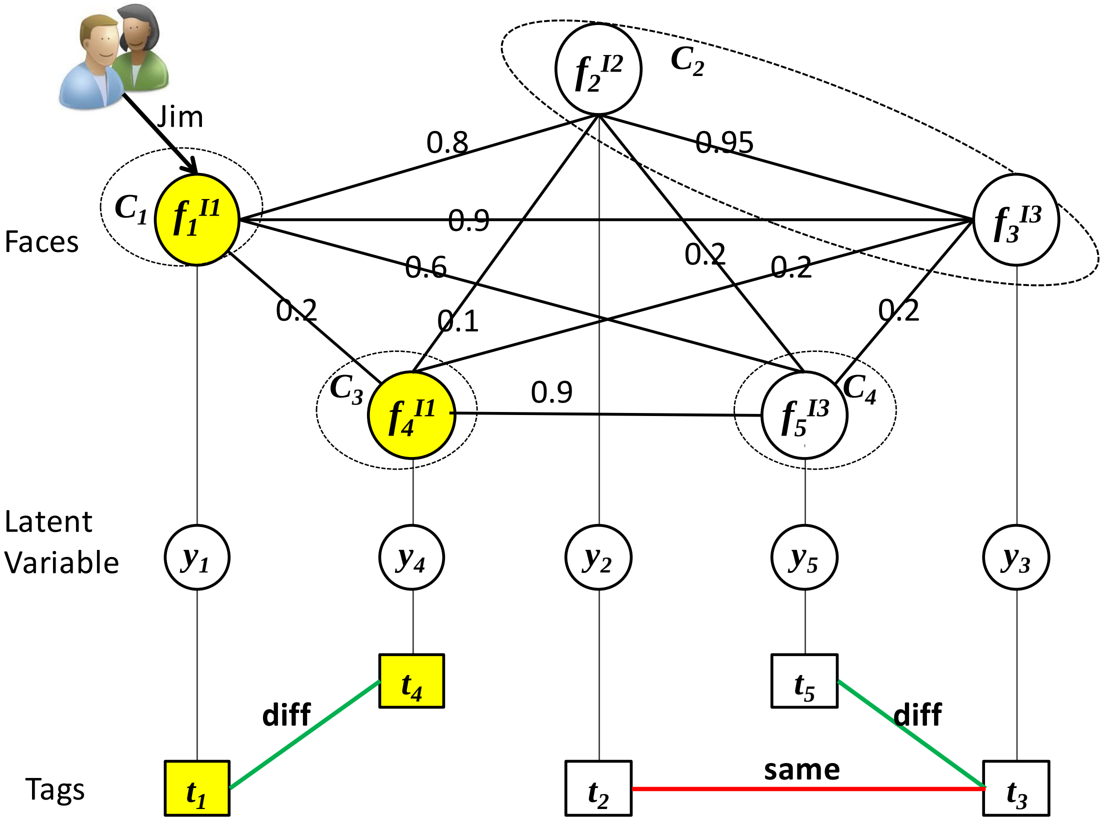


Figure 5.6: Gaussian process model is leveraged to infer the "local" probability of unlabeled samples.

$T_L = \{t_1, t_2, \dots t_L\}$, and unlabeled face set F_U. Our goal is to infer the posterior probability $p(t_u|F, T_L)$, given an unobserved face $f_u \in F_U$. We can then update the probabilistic database with revised identity attribute. For example, in Fig. 5.6, with face $f_1^{I_1}$ labeling as "Jim", face $f_4^{I_1}$ will be automatically labeled as "not Jim" according to the "diff" constraint, and thus the labeled face set turns to be $F_L = \{f_1^{I_1}, f_4^{I_1}\}$ with labels $T_L = \{1, -1\}$. We aim to induce the probability of unlabeled faces $(f_2^{I_2}, f_3^{I_3}, f_5^{I_3})$ referring to "Jim".

Kapoor *et al.* [51] use a Gaussian Process (GP) prior with contextual constraints to predict posterior distribution of tag labels over a set of unlabeled faces. We use a simplified form of their model, using the GP to only produce "local" predicts of face identity. We make use of the probabilistic database manager to enforce contextual constraints.

GP Classification: Gaussian process models have been widely explored in active learning, especially for visual classification. To use GPs for classification, we first introduce a latent variable vector $Y = \{y_1, y_2, \ldots, y_n\}$, where n is the number of labeled entities (faces). The discrete class label t_i for a face f_i is generated via the continuous latent variable y_i. Latent variables capture the assumption that similar faces should share similar predictions. The posterior distribution $p(Y|F)$ can be formally denoted as $p(Y|F) \sim N(0, \mathcal{K})$, where $\mathcal{K}$ refers to a kernel matrix with $\mathcal{K}_{ij} = k(f_i, f_j)$, which encodes the similarity between face pairs. In the experiments, $\mathcal{K}_{ij} = exp(-\frac{D_{ij}}{mean(D)})$, where D_{ij} refers to the distances between face pair f_i and f_j based on the face representations (Note that, this kernel function is positive semi-definite). Following standard GP constructions [71, 69, 32], given an unlabeled face f_u, the posterior probability over latent variable y_u has a simple form, $p(y_u|F, t_L) \sim N(\overline{y}_u, \sigma_u^2)$, where,

$$\overline{y}_u = k_u^T (K_{LL} + \sigma^2 I)^{-1} T_L \tag{5.1}$$

$$\sigma_u^2 = k_{uu} + \sigma^2 - k_u^T (K_{LL} + \sigma^2 I)^{-1} k_u \tag{5.2}$$

Here, the notation K_{LL}, k_u, k_{uu} respectively refer to the kernel matrix containing covariance between training samples, the kernel vector consisting of covariance between training samples and unlabeled samples, and the covariances of the test sample to itself. The class label t_u is given by the sign of predicted mean $\overline{y}_u$.

Constraints: Using the above model by itself only produces local predictions for each unlabeled face. These may not be consistent with contextual "same" and "diff" constraints encoded in our database. For example, in Figure 5.6 the probability of t_2 and t_3 referring to "Jim" should be equal, while t_3 and t_5 can not refer to "Jim" simultaneously due to the "diff" constraint. To enforce these constraints, we simply request the probabilistic database manager to return a valid query. Internally, the manager is solving an interference problem using standard algorithmic techniques such as sampling or belief propagation [72].

5.2.2 Query-driven Active Learning

To further improve the quality of query answers, we will generate some questions/faces to return to users for feedbacks, which will be leveraged to update the classification model in order to produce new query answers. In this chapter, we primarily focus on the question of face identity, however our framework can be applied to other types of questions.

Exhaustive Algorithm based on Impact

Intuitively, the sample with the most impact should be selected, where the impact refers to the uncertainty reduction to the query answers. Entropy [3] is the most common way to measure uncertainty. Given the probability distribution of sample s, entropy is defined as $H(s) = -\sum_{c \in class} p_c(s) log(p_c(s))$. Suppose that we obtain the current query result node distributions $\mathcal{R} = \{r_j\}_{j=1}^{m}$, then the uncertainty of this query answer can be measured using the total entropy $H(\mathcal{R}) = \sum_{j=1}^{m} H(r_j)$.

For each unlabeled face f_i, we can predict the new query answer set $\mathcal{R}^{f_i}$ with f_i resolved, by assume it to be true or false. Thus we can estimate the expected uncertainty of the new query answer set $E_{f_i}[H(\mathcal{R}^{f_i}|f_i)]$. Therefore, the impact of f_i can be defined as the uncertainty reduction $I(f_i) = H(\mathcal{R}) - E_{f_i}[H(\mathcal{R}^{f_i}|f_i)]$. Our goal is to choose the unlabeled face with the maximum impact, $f^* = \arg\max_{i \in U} I(f_i)$.

However, this algorithm is very expensive to compute because it requires repeating the query processing procedure for every unlabeled sample. Thus we need to explore the efficient approach to estimate the impact.

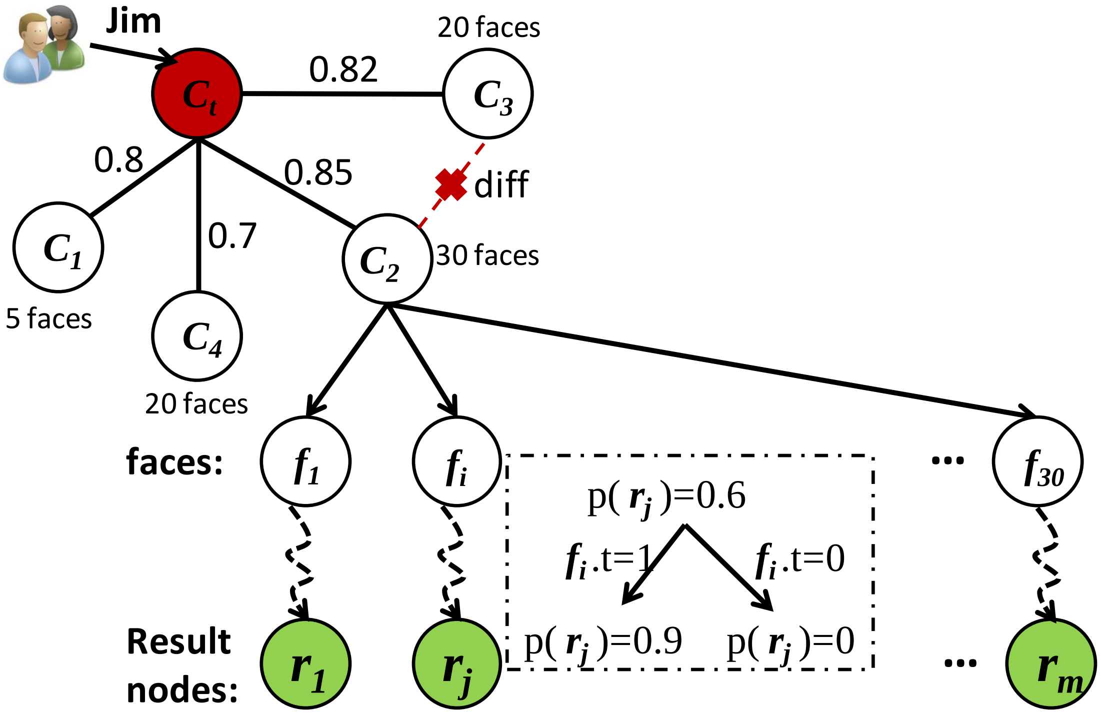


Figure 5.7: Query-driven entropy is introduced to measure the uncertainty of query answers induced by the uncertain samples. For instance, if f_i is resolved, we compute uncertainty reduction of result node r_j which depends on f_i.

Efficient Approach to Estimate Impact

In our scenario, partial clustering process is performed to cluster faces into initial groups where faces in the same group are assumed to refer to the same entity, therefore, we can choose the returned samples in the group level. For each group, one face is returned to users to represent the whole cluster. Our goal is to choose proper criteria to measure the impact of each unlabeled cluster.

Entropy. To estimate the impact of an unlabeled cluster C_k, as most prior face tagging work suggested [3, 51], we can compute entropy for each face f_i, denoted as $H(f_i)$. Then the impact of cluster C_k can be defined as the total entropy of faces, $I(C_k) = \sum_{f_i \in C_k} H(f_i)$.

Query-driven Entropy. However, in our problem, we aim to seek the samples which can maximally reduce the uncertainty of query answers. Therefore, we introduce the concept of query-driven entropy.

As illustrated in Figure 5.7, given a query and the corresponding result nodes $\{r_j\}_{j=1}^{m}$, we can discover the dependence between the face node and result node. For each face f_i, we can record the result node r_j dependent on f_i. This dependence is represented as $f_i \rightsquigarrow r_j$. We define the query-driven entropy of face f_i as the uncertainty reduction of the corresponding result node r_j with f_i resolved, where $f_i \rightsquigarrow r_j$. Therefore, the query-driven entropy of face f_i is defined as

$$I^q(f_i) = \sum_{f_i \rightsquigarrow r_j} H(r_j) - E_{f_i}[H(r_j|f_i)] \tag{5.3}$$

Here, $H(r_j)$ refers to the current entropy of result node r_j, and $E_{f_i}[H(r_j|f_i)]$ is to the expected entropy of updated result node r_j with f_i resolved. Therefore, the corresponding query-driven entropy for a cluster node C_k is defined as $I^q(C_k) = \sum_{f_i \in C_k} I^q(f_i)$.

Query-driven Entropy with Constraints. With further observation, we discover that constraint is an important factor that can not be omitted, because the identity of one cluster will impact the other with constraints. For instance, as illustrated in Figure 5.7, "diff" constraints exist between C_2 and C_3, therefore, if C_2 is resolved to be true, then C_3 will be false automatically. Thus the entropy of C_2 should also include the entropy of C_3 if C_2 resolved to be true. Based on this observation, the entropy should be defined considering constraints.

$$\widetilde{I^q}(C_k) = I^q(C_k) + p(C_k) \sum_{(k,l) \in \varepsilon^-} I^q(C_l) \tag{5.4}$$

where C_l refers to the cluster with "diff" constraints to C_k. Utilizing the above equation, we can compute the query-driven entropy with constraints $\widetilde{I^q}(C_k)$ for each unlabeled cluster C_k, and use it as criteria to choose the sample returned to users for tagging, denoted as $C^* = \arg\max_{k \in U} \widetilde{I^q}(C_k)$.

Select K questions in a batch

```
Algorithm 2: Greedy Algorithm for $K$ Questions
input  : Unlabeled set $\mathbb{S}$; Parameter $K$
output: Selected set $\Omega$, where $|\Omega| = K$
$\Omega \leftarrow \emptyset$
$Q \leftarrow \emptyset$
foreach node $C_i \in \mathbb{S}$ do
    Compute $\widetilde{I^q}(C_i)$
    $Q.insert(C_i, \widetilde{I^q}(C_i))$
while $|\Omega| < K$ and $|Q| > 0$ do
    $C_{top} \leftarrow Q.pop()$
    $\mathcal{C}_\Omega \leftarrow \{C_j : C_j \in \Omega \wedge (C_{top}, C_j) \in \varepsilon^-\}$
    if $\mathcal{C}_\Omega = \emptyset$ then
        $\Omega \leftarrow \Omega \cup \{C_{top}\}$
        $\mathbb{S}_{cand} \leftarrow \mathbb{S}_{cand} \setminus \{C_{top}\}$
    else
        Update $\widetilde{I^q}(C_{top})$
        $Q.insert(C_{top}, \widetilde{I^q}(C_{top}))$
return $\Omega$
```

So far, we have discussed the criteria to choose samples for feedbacks. However, in the real applications, it is inefficient to return only 1 question to users in each iteration. Therefore, next we will explore the strategy to choose K questions in each iteration. At the first thought, the samples ranked in the top K list can be returned to users. However, with further consideration, we discover that constraints make this problem a little more complex. For instance, as illustrated in Figure 5.7, suppose that we set $K = 2$, and C_2 and C_3 rank in the top 2 list, it is not wise to return them together, since the resolve of C_2 might lead to the resolve of C_3. Therefore, once we choose C_2, the impact of C_3 should be updated to $(1 - p(C_2))\widetilde{I^q}(C_3))$ (C_3 is needed to resolve only C_2 is false). Based on this consideration, we propose the greedy algorithm to choose K questions.

5.3 Experiments and Results

To evaluate the proposed query-driven face tagging paradigm, we conduct experiments on three human-centered data collections: Gallagher, Wedding, and Surveillance. Gallagher[33] is a public family album capturing the daily life of a family, containing three children, their parents and friends, about 37 people with a total of 1064 faces in 591 images. The wedding dataset downloaded from Web Picasa, captures people in a wedding ceremony, including the bride, the groom, their relatives and friends, containing 31 people with a total 1433 faces in 643 images. The surveillance dataset contains images that capture the daily life of faculty and students in the 2nd floor of a computer science building. There are 45 people appearing in 1030 images, but with only 70 faces detected due to the low image quality. Note that we choose the middle size photo collection rather than the one with very large volume for the sake of simplifying experiment evaluation.

5.3.1 Image Pre-processing

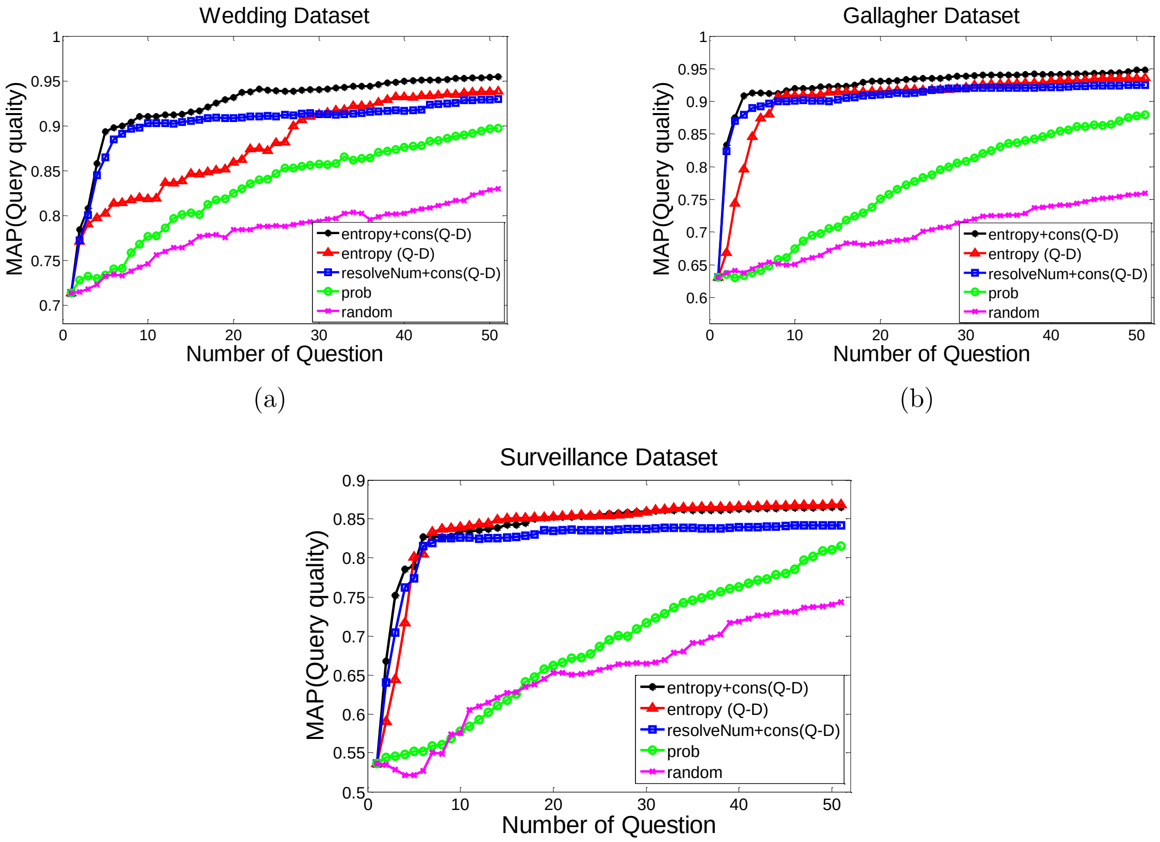
Before performing experiments, middle-level semantic information needs to be extracted, which can be used as query conditions. The information includes image captured time and Geo-location, which can be extracted from the EXIF data, assumed to be deterministic values. Besides, face attributes can also be obtained using the online attributes extraction system [52], which can return 73 types of attributes, including intrinsic attributes like “gender”, “age”, “ethnicity”, and describable attributes “black hair”, “big nose”, etc. For each attribute, the system can return a probabilistic value indicating how likely the attribute to be true.

Our face recognition pipeline uses Picasa's face detector [1] to detect face regions. After face alignment, each face region will be resized to 200 × 200 pixels, which will be further divided into 10 × 10 patches. From each patch, we can extract Local Binary Patterns (LBP) [4] feature as the facial representation. Principle components analysis will then be performed to reduce the feature dimensionality. The distance metric used is Euclidean distance. The partial clustering results are obtained by performing affinity propagation [31] with a very conservative threshold to guarantee the purity of each cluster.

5.3.2 Results and Analysis

In the experiments, we mainly concentrate on the person-related "selection" queries. To simulate user behaviors, we assume that users will specify the target person by labeling one group of faces, and present a query with other conditions. Our framework is able to answer the query in the active learning paradigm. Query answer is returned to users in a form of ranking list where each result node is sorted based on the relevance probability $p(r_i)$. Average precision has been widely employed to measure the quality of information retrieval task due to its good discrimination and stability. Therefore, we propose to leverage it as the metric to evaluate the quality of query answer. It is the average of the precision value obtained for the set of top k samples existing after each relevant sample is retrieved, and this value is then averaged over information needs, defined as follows.

$$AP = \sum_{k=1}^{m} precision(k) \Delta recall(k) \tag{5.5}$$



(a)

(b)

(c)

Figure 5.8: Comparison of Different Strategies on Three Datasets.

where k is the rank in the sequence of returned face list, m is total result number, $precision(k)$ is the precision at cut-off k in the list, and $\Delta recall(k)$ is the change in recall from items $k-1$ to k. We randomly choose queries and perform the process 100 times to compute mean AP as the criteria.

Comparison of Question Generation Strategy

Experiments are performed to evaluate the question generation strategy discussed in section 5.2.2. As illustrated in Figure 5.8, we compare the proposed criteria query-driven entropy with constraints (*entropy+cons(Q-D)*) with other methods. *Random* is a naive approach which chooses sample randomly. *Prob* refers to the strategy selecting samples with the highest relevance probability to the target node. We also compare with two other query-driven approaches to validate the superiority of our proposed criteria. The method *entropy(Q-D)* computes query-driven entropy without considering constraints. Paper [96] proposed to use the number of resolved faces considering constraints to measure the impact of a cluster node. Here we leverage it in a query-driven manner, referred as *resolvenum + cons(Q-D)*.

Figure 5.8 illustrates the comparison between the above strategies on the three data collections. It demonstrates the tendency of mean average precision (MAP) against the increase of question number. From the plots, we can see that the query-driven approaches allow quick improvement to the query answer quality (with just a few tagged face clusters), compared with the "*random*" and "*prob*" strategy. Analyzing the overall results, we discover that the "*entropy+cons(Q-D)*" approach can achieve the best performance because it considers contextual constraints to compute query-related entropy, which can appropriately estimate the impact of each sample. In contrast, the method "*entropy(Q-D)*" does not illustrate very good performance at the first several iterations due to the lack of consideration of constraints. The approach "*resolvenum + cons(Q-D)*" experiences a significant improvement in the first several iterations, but with slower process after consuming the benefit of constraints. We also discover that in surveillance dataset, the "*entropy+cons(Q-D)*" method loses the advantages, because most of frames in this dataset contain only single person, which leads to the lack of "diff" constraints.

Query-driven VS. Query-agnostic Approaches

We also perform experiments to compare query-driven and query-agnostic (or generic) approaches, from the perspective of improving query answer quality and face recognition performance (the goal of traditional face-tagging) respectively. To perform the query-agnostic approaches, we compute "*entropy*" and "*entropy+constraints*" based on the entropy of face nodes, without considering queries.

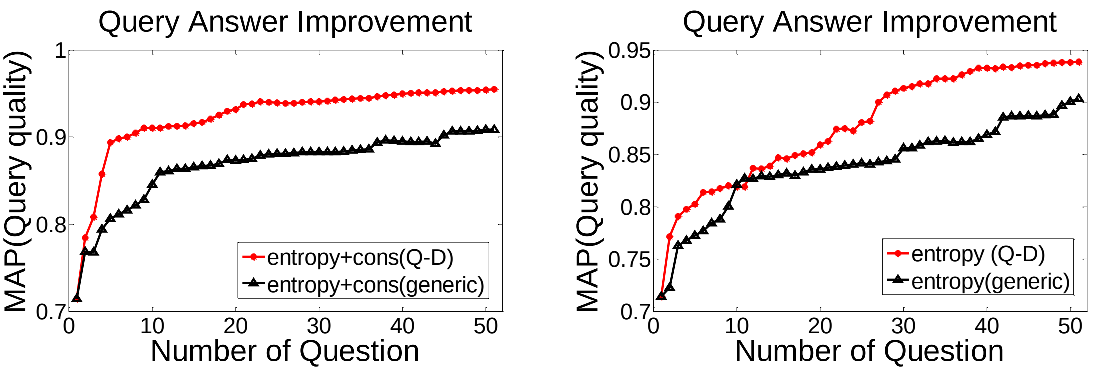


Figure 5.9: Query-driven VS. Query-agnostic with Query Answer Quality

Figure 5.9 demonstrates their comparison from the perspective of improving query answer quality. The results illustrate that query-driven approaches have significant superiority compared to query-agnostic approaches towards quality answer improvement.

In the real applications, we assume that many different users can present different queries, where the previous labeling results can be accumulated to be leveraged for the following queries. To simulate this behavior, we perform incremental query experiments illustrated in Figure 5.10. For each query, we assume that users will be satisfied with answers after 9 questions (experiments show that query answer quality can be significantly improved after a few questions). Then we simulate another user will continue to present different queries on the partial-cleaned dataset. We randomly generate 100 query orders to simulate this behavior, and average the performance. The results demonstrate that query-driven approach can achieve much better results with the incremental queries.

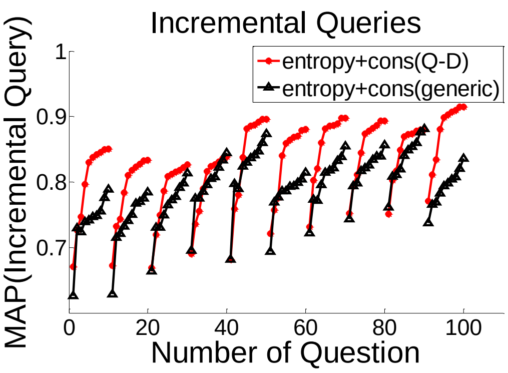


Figure 5.10: Incremental Queries.

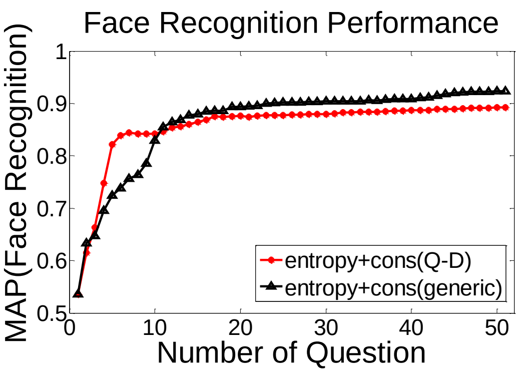


Figure 5.11: Face Recognition.

We also plot the comparison from the view of face recognition performance in Figure 5.11. Since the query-driven approaches are designed to favor the improvement of query quality, it will sacrifice the face recognition performance in the entire dataset, to some extent. However, the experiments show that the query-driven approaches still have relatively good performance.

Select K questions in each iteration

It is inefficient to ask users only one question in each iteration, therefore, we propose to return K questions in each round. An interesting question is how to choose the size K in each iteration. To choose the appropriate value for parameter K, we select different values (from 1 to 30) and plot the tendency of MAP change. As illustrated in Figure 5.12, we discover that the larger values of K lead to the slower improvement of query answer quality. This is because if we choose a larger value for K, the classifier cannot be updated promptly, and some unnecessary questions will be asked. However, a small K value will result in too many iterations and computation load. Therefore, to trade off between the two factors, we can set $K = 5$ in the application. Figure 5.13 illustrates the performance comparison with $K = 5$.

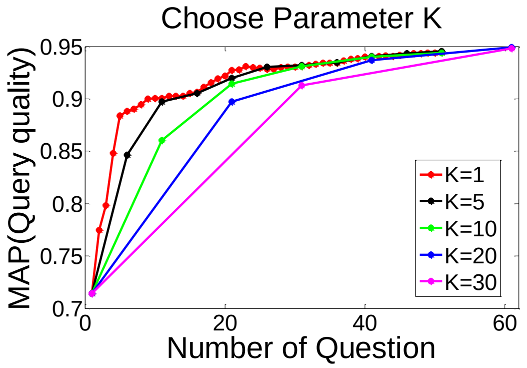


Figure 5.12: Parameter Selection.

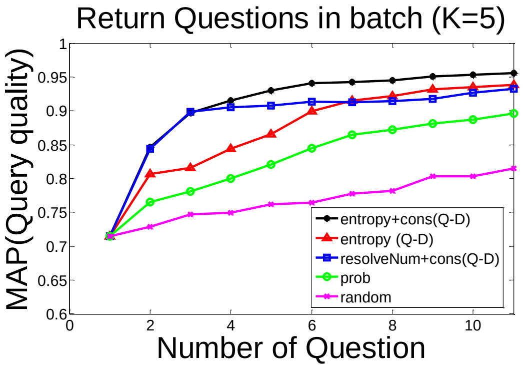


Figure 5.13: Run with $K = 5$.

5.4 Chapter Conclusion and Future Work

In this chapter, we have introduced a query-drive paradigm for face clustering/tagging which can be seamlessly integrated into image analysis/retrieval process, to address the challenges of big data. Our goal is to explore query-driven active learning strategies to achieve accurate query answers with minimum user participation. In our framework, queries have been represented by factor graph to facilitate probabilistic reasoning, where Gaussian process models have been used for probability inference. We have proposed the criteria considering query-driven entropies and contextual constraints to select tagged samples to maximally improve query answer quality. Experiments on real-world datasets have demonstrated the superiority of the proposed query-driven approaches compared to query-agnostic methods towards query answer improvement.

In this work, we only considered the simple selection queries. In the future, more complex nature of queries, such as joins, aggregations, or a full class of SQL, can be explored. When returning questions to users, we only consider questions about face identity, other types of questions such as human attributes, or the activities included in the image, can be further studied in the future.

Chapter 6

Conclusion and Future Work

In this chapter, we first conclude our contributions in exploring entity resolution techniques to improve multimedia semantic interpretation in the era of big multimedia data. We then identify and present a few open areas that have not been touched or completely solved in this dissertation.

6.1 Summary of Dissertation Research

In this dissertation, we have systematically explored the automatic as well as interactive entity resolution techniques to improve multimedia semantic interpretation, particularly for the person identification task, in the era of big multimedia data. First, we have explored how to leverage the automatic techniques to exploit relationships, contextual information, domain semantics, etc., to enhance the performance of face clustering and recognition. Specifically, we have studied the person identification problem in the context of surveillance videos and proposed a context-based framework for low-quality data, which leverages the entity resolution framework called RelDC to integrate multiple contextual information. Then we have investigated the face clustering problem and proposed a unified framework that employs bootstrapping to automatically learn adaptive rules to integrate heterogeneous context information together. Besides the automatic data cleaning techniques, we have also explored the interactive techniques to further improve the performance. Particularly, we have proposed the query-driven approach to face clustering/tagging which can be seamlessly integrated into the image analyses/retrieval process. The goal is to investigate the query-driven active learning strategies in order to achieve the accurate query answers with minimum user participation.

In the problem of person identification in the context of Smart Video Surveillance, we have demonstrated how an instance of indoor person identification problem (for video data) can be converted into the problem of entity resolution (which typically deals with textual data). The area of entity resolution has become very active as of recently, with many research groups proposing powerful generic algorithms and frameworks. Thus, establishing a connection between the two problems has the potential to benefit the person identification problem, which could be viewed as a specific instance of ER problem. Our experiments of using a simplified version of RelDC framework for entity resolution have demonstrated the effectiveness of our approach.

As to the face clustering problem, we have proposed a unified framework for integrating heterogeneous context information (including common scene, people co-occurrence, human attributes and clothing) to improve the quality/recall of face clustering. The context information has been used for both: computing context similarities to link clusters that co-refer as well as for generating context constraints to differentiate clusters that do not co-refer. The proposed unified framework leverages bootstrapping to automatically learn the adaptive rules to integrate heterogeneous context information together to iteratively merge clusters, in order to improve the recall of clustering results. Finally, we discuss several methods to integrate human-in-the-loop in order to achieve very high-quality clustering results. Our experiments on the real-world datasets demonstrated the effectiveness of the extracted context features and of the overall unified framework.

Finally, we have investigated the interactive techniques to further improve the performance of person identification. We have introduced a query-drive paradigm for face clustering/tagging which can be seamlessly integrated into image analysis/retrieval process, to address the challenges of big data. Our goal is to explore query-driven active learning strategies to achieve accurate query answers with minimum user participation. In our framework, queries have been represented by factor graph to facilitate probabilistic reasoning, where Gaussian process models have been used for probability inference. We have proposed criteria considering query-driven entropies and contextual constraints to select tagged samples to maximally improve query answer quality. Experiments on real-world datasets have demonstrated the superiority of the proposed query-driven approaches compared to query-agnostic methods towards query answer improvement.

In short, we summarize the following conclusions:

- Connected the problem of person identification with the task of entity resolution, and proposed a context-based framework which leverages RelDC approach to integrate multiple contextual information to improve the performance of person identification.
- Investigated the appropriate methods to leverage context features in cluster level, and proposed a unified framework which leverages bootstrapping to automatically learn the adaptive rules to integrate heterogeneous context information to iteratively improve the recall of clustering results.
- Discussed several methods to integrate human-in-the-loop mechanism which leverages human intelligence to achieve very high-quality clustering results.
- Introduced a query-drive paradigm for face tagging which can be seamlessly integrated

into image analysis/retrieval process to address the challenges of big data, and explored the query-driven active learning strategy in order to achieve accurate query answers with minimum user participation.

6.2 Future Research Directions

We identify a few open research directions listed below. The directions extend our research in this dissertation.

- Our work about "person identification for smart surveillance videos" is just a first step in exploiting ER techniques for video data cleaning tasks. The future work can explore how additional features derived from video, as well as additional semantics in the form of context and metadata (e.g., knowledge of building layout, offices, meeting times, etc.) can be used to further improve person identification.
- In the work of "context-based face clustering with human-in-the-loop", we only discussed several simple strategies to integrate human-in-the-loop techniques. In the future, one research direction is to explore the human-in-the-loop strategies from a formal probabilistic perspective.
- In the work "query-driven face tagging", we only considered the simple selection queries. In the future, more complex nature of queries, such as joins, aggregations, or a full class of SQL, can be explored. Another limitation is that we only choose to return questions about face identity to users, other types of questions such as human attributes, or the activities included in the image, can be further studied in the future.

System Implementation. In the future, we plan to implement a system which can incorporate these proposed approaches into the real applications. This system can leverage the automatic and interactive data cleaning algorithms to improve the semantic interpretation of multimedia data. This system will be able to provide fundamental support to I-sensorium project, which provides a shared experimental laboratory housing state-of-the-art sensing, actuation, networking and mobile computing devices to enable researchers to emulate sentient spaces and applications in their target domains of interest.

The system will consist of several components including semantic extraction component, automatic cleaning component, and interactive cleaning component. The goal of semantic extraction component is to obtain the initial face clustering/recognition results from raw images/videos, leveraging the traditional approaches which generally include several functions such as face/pedestrian detection, feature extraction, similarity computation, etc. The acquired initial results will be passed to the automatic cleaning component where automatic processing techniques will be leveraged to improve the data quality. Subsequently, these improved semantic data will construct the probabilistic semantic database, on top of which users can present queries. To achieve the accurate answers to these queries, interactive cleaning component will adopt the form of "question-answer" in order to reduce uncertainties with human inputs. These human inputs can also be leveraged to improve the automatic cleaning techniques.

Bibliography

[1] `http://picasaweb.google.com`.

[2] Cisco visual networking index: Forecast and methodology, 20132018. Cisco Systems, Inc, 2014.

[3] P. P. M. B. A.D. Holub. Entropy-based active learning for object recognition. *Second IEEE Workshop on Online Learning for Classification, (OLC). Computer Vision and Pattern Recognition (CVPR)*, 2008.

[4] T. Ahonen, A. Hadid, and M. Pietik. Face description with local binary patterns: Application to face recognition. In *IEEE Trans. Pattern Anal*, 2006.

[5] E. Amigo and et al. A comparison of extrinsic clustering evaluation metrics based on formal constraints. In *Technical Report*, 2008.

[6] L. An, B. Bhanu, and S. Yang. Boosting face recognition in real-world surveillance videos. In *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 270–275, 2012.

[7] L. An, M. Kafai, and B. Bhanu. Dynamic bayesian network for unconstrained face recognition in surveillance camera networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(2):155–164, 2013.

[8] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 586–597. VLDB Endowment, 2002.

[9] D. Anguelov, K.-c. Lee, S. B. Gokturk, and B. Sumengen. Contextual identity recognition in personal photo albums. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[10] A. Arasu, S. Chaudhuri, and R. Kaushik. Transformation-based framework for record matching. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 40–49. IEEE, 2008.

[11] M. Balcan, A. Blum, P. P.Choi, J. Lafferty, B. Pantano, M. R. Rwebangira, and X. Zhu. Person identification in webcam images: An application of semi-supervised learning. In *ICML Workshop on Learning from Partially Classified Training Data*, 2005.

[12] T. L. Berg, A. C. Berg, and et al. Names and faces in the news. In *IEEE ICPR*, 2004.

[13] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 143–154. ACM, 2005.

[14] K. Chakrabarti, M. Ortega-Binderberger, K. Porkaew, and S. Mehrotra. Similar shape retrieval in mars. In *IEEE International Conference on Multimedia and Expo (II)*, pages 709–712, 2000.

[15] K. Chakrabarti, K. Porkaew, and S. Mehrotra. Efficient query refinement in multimedia databases. In *ICDE*, page 196, 2000.

[16] S. Chaudhuri, K. Ganjam, V. Ganti, R. Kapoor, V. Narasayya, and T. Vassilakis. Data cleaning in Microsoft SQL Server 2005. In *ACM SIGMOD Conference*, 2005.

[17] S. Chen, D. V. Kalashnikov, and S. Mehrotra. Adaptive graphical approach to entity resolution. In *Proc. of ACM IEEE Joint Conference on Digital Libraries (JCDL 2007)*, Vancouver, British Columbia, Canada, June 17–23 2007.

[18] W. Chen, W. Fan, and S. Ma. Incorporating cardinality constraints and synonym rules into conditional functional dependencies. *Information Processing Letters*, 109(14):783–789, 2009.

[19] Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting relationships for object consolidation. In *Proc. of International ACM SIGMOD Workshop on Information Quality in Information Systems (ACM IQIS 2005)*, Baltimore, MD, USA, June 17 2005.

[20] Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Adaptive graphical approach to entity resolution. In *JCDL*, 2007.

[21] Z. S. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting context analysis for combining multiple entity resolution systems. In *Proc. of ACM SIGMOD International Conference on Management of Data (ACM SIGMOD 2009)*, Providence, RI, USA, June 29–July 2 2009.

[22] K. Choi, K.-A. Toh, and H. Byun. An efficient incremental face annotation for large scale web services. *Telecommunication Systems*, 47(3-4):197–214, 2011.

[23] J. Chomicki and J. Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Information and Computation*, 197(1):90–121, 2005.

[24] E. F. Codd. A data base sublanguage founded on the relational calculus. In *Proceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control*, pages 35–68. ACM, 1971.

[25] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: Consistency and accuracy. In *Proceedings of the 33rd international conference on Very large data bases*, pages 315–326. VLDB Endowment, 2007.

[26] M. Cristani, M. Bicego, and V. Murino. Audio-visual event recognition in surveillance video sequences. In *IEEE Transactions on Multimedia*, 2007.

[27] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang. Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 367–376. ACM, 2007.

[28] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[29] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. In *AVBPA*, 1997.

[30] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *VLDB J.*, 21(2):213–238, 2012.

[31] B. J. Frey and D. Dueck. Clustering by passing messages between data points. In *Science*, 2007.

[32] A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler. Labeling examples that matter: Relevance-based active learning with gaussian processes. In *Pattern Recognition - 35th German Conference, GCPR 2013, Saarbrücken, Germany, September 3-6, 2013. Proceedings*, pages 282–291, 2013.

[33] A. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *IEEE CVPR*, 2008.

[34] A. Gallagher and T. Chen. Understanding images of groups of people. In *IEEE CVPR*, 2009.

[35] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, and T. Chua. Camera constraint-free view-based 3D object retrieval. *IEEE Transactions on Image Processing*, 21(4):2269–2281, 2012.

[36] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai. 3D object retrieval and recognition with hypergraph analysis. *IEEE Transactions on Image Processing*, 21(9):4290–4303, 2012.

[37] Y. Gao, M. Wang, Z. Zha, J. Shen, X. Li, and X. Wu. Visual-textual joint relevance learning for tag-based social image search. *IEEE Transactions on Image Processing*, 22(1):363–376, 2013.

[38] L. Gomes. Delusions of big data and other huge engineering efforts. *IEEE Spectrum*, 2014.

[39] R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. R. Namburu. *Data Mining for Scientific and Engineering Applications.* Kluwer Academic Publishers, 2001.

[40] J. Han, R. B. Altman, V. Kumar, H. Mannila, and D. Pregibon. Emerging scientific applications in data mining. *Commun. ACM*, 45(8):54–58, 2002.

[41] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-S. Chua. Beyond search: Event-driven summarization for web videos. *TOMCCAP*, 7(4):35, 2011.

[42] R. Hong, M. Wang, G. Li, L. Nie, Z.-J. Zha, and T.-S. Chua. Multimedia question answering. *IEEE MultiMedia*, 19(4):72–78, 2012.

[43] J.Tang, S. Yan, R. Hong, G. Qi, and T. Chua. Inferring semantic concepts from community-contributed images and noisy tags. In *ACM Multimedia*, 2009.

[44] D. V. Kalashnikov. Super-EGO: Fast multi-dimensional similarity join. *The International Journal on Very Large Data Bases (VLDB Journal)*, 4(2):561–585, 2013.

[45] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, 20(11), Nov. 2008.

[46] D. V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (ACM TODS)*, 31(2):716–767, June 2006.

[47] D. V. Kalashnikov, S. Mehrotra, S. Chen, R. Nuray, and N. Ashish. Disambiguation algorithm for people search on the web. In *Proc. of the IEEE 23rd International Conference on Data Engineering (IEEE ICDE 2007)*, Istanbul, Turkey, April 16–20 2007. short publication.

[48] D. V. Kalashnikov, S. Mehrotra, and Z. Chen. Exploiting relationships for domain-independent data cleaning. In *SIAM International Conference on Data Mining (SIAM Data Mining 2005)*, Newport Beach, CA, USA, April 21–23 2005.

[49] M. Kantardzic and J. Zurada. *Next Generation of Data-Mining Applications.* John Wiley & Sons, 2005.

[50] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[51] A. Kapoor, G. Hua, A. Akbarzadeh, and S. Baker. Which faces to tag: Adding prior constraints into active learning. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 1058–1065, 2009.

[52] N. Kumar and et al. Describable visual attributes for face verification and image search. In *IEEE TPAMI*, 2011.

[53] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic face identification system using flexible appearance models. pages 393–401, 1995.

[54] Y. J. Lee and K. Grauman. Face discovery with social context. In *BMVC*, 2011.

[55] P. L. Lehman et al. Efficient locking for concurrent operations on b-trees. *ACM Transactions on Database Systems (TODS)*, 6(4):650–670, 1981.

[56] L.-J. Li and L. Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *International journal of computer vision*, 88(2):147–168, 2010.

[57] A. Lopatenko and L. Bravo. Efficient approximation algorithms for repairing inconsistent databases. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 216–225. IEEE, 2007.

[58] A. McCallum and B. Wellner. Object consolidation by graph partitioning with a conditionally-trained distance metric. In *KDD Workshop on Data Cleaning, Record Linkage and Object Consolidation*, 2003.

[59] D. McCarthy and U. Dayal. The architecture of an active database management system. In *ACM Sigmod Record*, volume 18, pages 215–224. ACM, 1989.

[60] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: Probabilistic matching for face recognition. In *FG*, pages 30–35, 1998.

[61] R. Nuray-Turan, D. V. Kalashnikov, and S. Mehrotra. Self-tuning in graph-based reference disambiguation. In *Proc. of the 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007), Springer LNCS*, Bangkok, Thailand, April 9–12 2007.

[62] R. Nuray-Turan, D. V. Kalashnikov, and S. Mehrotra. Exploiting web querying for web people search. *ACM Transactions on Database Systems (ACM TODS)*, 37(1), Feb. 2012.

[63] R. Nuray-Turan, D. V. Kalashnikov, and S. Mehrotra. Adaptive connection strength models for relationship-based entity resolution. *ACM Journal of Data and Information Quality (ACM JDIQ)*, 4(2), 2013.

[64] R. Nuray-Turan, D. V. Kalashnikov, S. Mehrotra, and Y. Yu. Attribute and object selection queries on objects with probabilistic attributes. *ACM Transactions on Database Systems (ACM TODS)*, 37(1), Feb. 2012.

[65] M. Ortega-Binderberger and S. Mehrotra. Relevance feedback techniques in the mars image retrieval system. *Multimedia Syst.*, 9(6):535–547, 2004.

[66] M. Ortega-Binderberger and S. Mehrotra. Relevance feedback techniques in the mars image retrieval system. *Multimedia Syst.*, 9(6):535–547, 2004.

[67] F. Z. Qureshi. Object video streams: A framework for preserving privacy in video surveillance.

[68] R. Ramakrishnan, J. Gehrke, and J. Gehrke. *Database management systems*, volume 3. McGraw-Hill New York, 2003.

[69] C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

[70] M. Saini, P. K. Atrey, S. Mehrotra, and M. Kankanhalli. W3-privacy: Understanding what, when, and where inference channels in multi-camera surveillance video. *Multimedia Tools and Applications*, pages 1–24, 2012.

[71] M. Seeger. Gaussian processes for machine learning. *Int. J. Neural Syst.*, 14(2):69–106, 2004.

[72] P. Sen, A. Deshpande, and L. Getoor. Prdb: managing and exploiting rich correlations in probabilistic databases. *VLDB J.*, 18(5):1065–1090, 2009.

[73] K. Shimizu, N. Nitta, and et al. Classification based group photo retrieval with bag of people features. In *ICMR*, 2012.

[74] K. Sung and T. Poggio. Example-based learning for view-based human face detection. pages 39–51, 1998.

[75] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain. Image annotation by *k*nn-sparse graph-based label propagation over noisily tagged web images. *ACM Transactions on Intelligent Systems and Technology*, 2(2):14–23, 2011.

[76] J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua. Inferring semantic concepts from community-contributed images and noisy tags. In *ACM Multimedia*, pages 223–232, 2009.

[77] J. Tang, Z.-J. Zha, D. Tao, and T.-S. Chua. Semantic-gap-oriented active learning for multilabel image annotation. *IEEE Transactions on Image Processing*, 21(4):2354–2360, 2012.

[78] Y. Tian, L. M. G. Brown, A. Hampapur, M. L. Senior, and C. Shu. Ibm smart surveillance system (s3): event based video surveillance system with an open and extensible framework. In *Mach. Vis. Appl.*, 2008.

[79] Y. Tian, W. Liu, R. Xiao, F. Wen, and X. Tang. A face annotation framework with partial clustering and interactive labeling. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[80] R. Vaisenberg, S. Mehrotra, and D. Ramanan. Semartcam scheduler: *ema*ntics driven eal-ime data collection from indoor *am*era networks to maximize event detection. *J. Real-Time Image Processing*, 5(4), 2010.

[81] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.

[82] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012.

[83] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song. Unified video annotation via multigraph learning. *IEEE Trans. Circuits Syst. Video Techn.*, 19(5):733–746, 2009.

[84] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu. Multimodal graph-based reranking for web image search. *IEEE Transactions on Image Processing*, 21(11):4649–4661, 2012.

[85] M. Wang, B. Ni, X.-S. Hua, and T.-S. Chua. Assistive tagging: A survey of multimedia tagging with human-computer joint exploration. *ACM Computing Surveys (CSUR)*, 44(4):25, 2012.

[86] P. Wu and F. Tang. Improving face clustering using social context. In *ACM Multimedia*, 2010.

[87] J. Yagnik and A. Islam. Learning people annotation from the web via consistency learning. In *MIR*, 2007.

[88] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *Proceedings of the VLDB Endowment*, 4(5):279–289, 2011.

[89] G. Yang and T. S. Huang. Human face detection in a complex background. *Pattern Recognition*, 27(1):53–63, 1994.

[90] M. Yang and K. Yu. Real-time clothing recognition in surveillance videos. In *ICIP*, 2011.

[91] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. pages 34–58, 2002.

[92] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*, pages 102–111, 2006.

[93] K. C. Yow and R. Cipolla. Feature-based human face detection. *IMAGE AND VISION COMPUTING*, 15:713–735, 1996.

[94] C. Zhang and Z. Zhang. A survey of recent advances in face detection, 2010.

[95] L. Zhang, D. V. Kalashnikov, and S. Mehrotra. A unified framework for context assisted face clustering. In *ACM International Conference on Multimedia Retrieval (ACM ICMR 2013)*, Dallas, Texas, USA, April 16–19 2013.

[96] L. Zhang, D. V. Kalashnikov, and S. Mehrotra. Context-assisted face clustering framework with human-in-the-loop. *International Journal of Multimedia Information Retrieval*, pages 69–88, June 2014.

[97] L. Zhang, D. V. Kalashnikov, S. Mehrotra, and R. Vaisenberg. Context-based person identification framework for smart video surveillance. *Machine Vision and Applications*, pages 1–15, 2013.

[98] L. Zhang, R. Vaisenberg, S. Mehrotra, and D. V. Kalashnikov. Video entity resolution: Applying er techniques for smart video surveillance. In *IQ2S Workshop in Conjunction with IEEE PERCOM 2011*, 2011.

[99] L. Zhang, K. Zhang, and C. Li. A topical pagerank based algorithm for recommender systems. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 713–714, 2008.

[100] W. Zhang and et al. Beyond face: Improving person clustering in consumer photos by exploring contextual information. In *ICME*, 2010.

[101] M. Zhao, Y. Teo, and et al. Automatic person annotation of family photo album. In *CIVR*, 2006.

[102] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *CVPR*, 2004.

[103] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2879–2886, 2012.