Chapter 1
Introduction: Key Subroutines of Multimedia Data Management

Abstract This chapter presents technical challenges that multimedia information management faces. We enumerate five key subroutines required to work together effectively so as to enable robust and scalable solutions. We provide pointers to the rest of the book, where in-depth treatments are presented.

Keywords Mathematics of perception · Multimedia data management · Multimedia information retrieval

1.1 Overview

The tasks of multimedia information management such as clustering, indexing, and retrieval, come up against technical challenges in at least three areas: data representation, similarity measurement, and scalability. First, data representation builds layers of abstraction upon raw multimedia data. Next, a distance function must be chosen to properly account for similarity between any pair of multimedia instances. Finally, from extracting features, measuring similarity, to organizing and retrieving data, all computation tasks must be performed in a scalable fashion with respect to both data dimensionality and data volume. This chapter outlines design issues of five essential subroutines, and they are:

1. Feature extraction,
2. Similarity (distance function formulation),
3. Learning (supervised and unsupervised),
4. Multimodal fusion, and
5. Indexing.

E. Y. Chang, *Foundations of Large-Scale Multimedia Information Management and Retrieval*, DOI: 10.1007/978-3-642-20429-6_1,
© Tsinghua University Press, Beijing and Springer-Verlag Berlin Heidelberg 2011

1.2 Feature Extraction

Feature extraction is fundamental to all multimedia computing tasks. Features can be classified into two categories, *content* and *context*. Content refers directly to raw imagery, video, and mucic data such as pixels, motions, and tones, respectively, and their representations. Context refers to metadata collected or associated with content when a piece of data is acquired or published. For instance, EXIF camera parameters and GPS location are contextual information that some digital cameras can collect. Other widely used contextual information includes surrounding texts of an image/photo on a Web page, and social interactions on a piece of multimedia data instance. Context and content ought to be fused synergistically when analyzing multimedia data [1].

Content analysis is a subject studied for more than a couple of decades by researchers in disciplines of computer vision, signal processing, machine learning, databases, psychology, cognitive science, and neural science. Limited progress has been made in each of these disciplines. Many researchers now are convinced that interdisciplinary research is essential to make ground breaking advancements. In Chap. 2 of this book, we introduce a model-based and data-driven hybrid approach for extracting features. A promising model-based approach was pioneered by neural scientist Hubel [2], who proposed a feature learning pipeline based on human visual system. The principal reason behind this approach is that human visual system can function so well in some challenging conditions where computer vision solutions fail miserably. Recent neural-based models proposed by Lee [3] and Serre [4] show that such model can effectively deal with viewing of different positions, scales, and resolutions. Our empirical study confirmed that such model-based approach can recognize objects of rigid shapes, such as watches and cars. However, for objects that do not have invariant features such as pizzas of different toppings, and cups of different colors and shapes, the model-based approach loses its advantages. For recognizing these objects, the data-driven approach can depict an object by collecting a representative pool of training instances. When combining model-based and data-driven, the hybrid approach enjoys at least three advantages:

1. *Balancing feature invariance and selectivity*. To achieve feature selectivity, the hybrid approach conducts multi-band, multi-scale, and multi-orientation convolutions. To achieve invariance, it keeps signals of sufficient strengths via pooling operations.
2. *Properly using unsupervised learning to regularize supervised learning*. The hybrid approach introduces unsupervised learning to reduce features so as to prevent the subsequent supervised layer from learning trivial solutions.
3. *Augmenting feature specificity with diversity*. A model-based only approach cannot effectively recognize irregular objects or objects with diversified patterns; and therefore, we must combine such with a data-driven pipeline.

Chapter 2 presents the detailed design of such a hybrid model involving disciplines of neural science, machine learning, and computer vision.

1.3 Similarity

At the heart of data management tasks is a distance function that measures *similarity* between data instances. To date, most applications employ a variant of the *Euclidean distance* for measuring similarity. However, to measure similarity meaningfully, an effective distance function ought to consider the idiosyncrasies of the application, data, and user (hereafter we refer to these factors as contextual information). The quality of the distance function significantly affects the success in organizing data or finding relevant results.

In Chaps. 4 and 5, we present two methods, first an unsupervised in Chap. 4 and then a supervised in Chap. 5, to quantify similarity. Chapter 4 presents dynamic partial function (DPF), which we formulated based on what we learned from some intensive data mining on large image datasets. Traditionally, similarity is a measure of all respects. For instance, the Euclidean function considers all features in equal importance. One step forward was to give different features different weights. The most influential work is perhaps that of Tversky [5], who suggests that similarity is determined by matching features of compared objects. The weighted Minkowski function and the quadratic-form distances are the two representative distance functions that match the spirit. The weights of the distance functions can be learned via techniques such as relevance feedback, principal component analysis, and discriminative analysis. Given some similar and some dissimilar objects, the weights can be adjusted so that similar objects can be better distinguished from the other objects.

However, the assumption made by these distance functions, that all similar objects are similar in the same respects [6], is questionable. We propose that *similarity is a process that provides respects for measuring similarity*. Suppose we are asked to name two places that are similar to England. Among several possibilities, Scotland and New England could be two reasonable answers. However, the respects England is similar to Scotland differ from those in which England is similar to New England. If we use the shared attributes of England and Scotland to compare England and New England, the latter pair might not be similar, and vice versa. This example depicts that objects can be similar to the query object in different respects. A distance function using a fixed set of respects cannot capture objects that are similar in different sets of respects. Murphy and Medin [7] provide early insights into how similarity works in human perception: "The explanatory work is on the level of determining which attributes will be selected, with similarity being at least as much a consequence as a cause of a concept coherence." Goldstone [8] explains that similarity is the process that determines the respects for measuring similarity. In other words, a distance function for measuring a pair of objects is formulated only after the objects are compared, not before the comparison is made. The respects for the comparison are activated in this formulation process. The activated respects are more likely to be those that can support coherence between the compared objects. DPF activates different features for different object pairs. The activated features are those with minimum differences — those which provide coherence between the objects. If coherence can be maintained (because sufficient a number of features are similar), then the objects

paired are perceived as similar. Cognitive psychology seems able to explain much of the effectiveness of DPF.

Whereas DPF learns similar features in an unsupervised way, Chap. 5 presents a supervised method to learn a distance function from contextual information or user feedback. One popular method is to weight the features of the Euclidean distance (or more generally, the L_p-norm) based on their importance for a target task [9–11]. For example, for answering a *sunset* image-query, color features should be weighted higher. For answering an *architecture* image-query, shape and texture features may be more important. Weighting these features is equivalent to performing a *linear* transformation in the space formed by the features. Although linear models enjoy the twin advantages of simplicity of description and efficiency of computation, this same simplicity is insufficient to model similarity for many real-world data instances. For example, it has been widely acknowledged in the image/video retrieval domain that a query concept is typically a nonlinear combination of perceptual features (color, texture, and shape) [12, 13]. Chapter 5 presents a *nonlinear* transformation on the feature space to gain greater flexibility for mapping features to semantics.

At first it might seem that capturing nonlinear relationships among contextual information can suffer from high computational complexity. We avoid this concern by employing the *kernel trick*, which has been applied to several algorithms in statistics, including support vector machines(SVM) and kernel PCA. The kernel trick lets us generalize distance-based algorithms to operate in the *projected space*, usually nonlinearly related to the *input space*. The *input space* (denoted as $\mathcal{I}$) is the original space in which data vectors are located, and the *projected space* (denoted as $\mathcal{P}$) is that space to which the data vectors are projected, linearly or nonlinearly. The advantage of using the *kernel trick* is that, instead of explicitly determining the coordinates of the data vectors in the projected space, the distance computation in $\mathcal{P}$ can be efficiently performed in $\mathcal{I}$ through a kernel function.

Through theoretical discussion and empirical studies, Chaps. 4 and 5 show that when similarity measures have been improved, data management tasks such as clustering, learning, and indexing can perform with marked improvements.

1.4 Learning

The principal design goal of a multimedia information retrieval system is to return data (images or video clips) that accurately match users' queries (for example, a search for pictures of a deer). To achieve this design goal, the system must first comprehend a user's query concept (i.e., a user's perception) thoroughly, and then find data in the low-level input space (formed by a set of perceptual features) that match the concept accurately. Statistical learning techniques can assist achieving the design goal via two complementary avenues: semantic annotation and query-concept learning.

Both semantic annotation and query-concept learning can be cast into the form of a supervised learning problem, which consists of three steps. First, a representative set of perceptual features is extracted from each training instance.

Second, each training feature–vector (other representations are possible) is assigned semantic labels. Third, a classifier is trained by a supervised learning algorithm, based on the labeled instances, to predict the class labels of a query instance. Given a query instance represented by its features, the semantic labels can be predicted. In essence, these steps learn a mapping between the perceptual features and a human perceived concept or concepts.

Chapter 3 presents the challenges of semantic annotation and query-concept learning. To illustrate, let D denote the number of low-level features (extracted by methods presented in Chap. 2), N the number of training instances, N^+ the number of positive training instances, and N^- the number of negative training instances ($N = N^+ + N^-$). Two major technical challenges arise:

1. *Scarcity of training data*. The features-to-semantics mapping problem often comes up against the $D > N$ challenge. For instance, in the query-concept learning scenario, the number of low-level features that characterize an image (D) is greater than the number of images a user would be willing to label (N) during a relevance feedback session. As pointed out by David Donoho, the theories underlying "classical" data analysis are based on the assumptions that $D < N$, and N approaches infinity. But when $D > N$, the basic methodology which was used in the classical situation is not similarly applicable.
2. *Imbalance of training classes*. The target class in the training pool is typically outnumbered by the non-target classes ($N^- \gg N^+$). For instance, in a k-class classification problem where each class has about the same number of training instances, the target class is outnumbered by the non-target classes by a ratio of k:1. The class boundary of imbalanced training classes tends to skew toward the target class when k is large. This skew makes class prediction less reliable.

To address these challenges, Chap. 3 presents a small sample, active learning algorithm, which also adjusts its sampling strategy in a concept-dependent way. Chapter 9 presents a couple of approaches to deal with imbalanced training classes. When conducting annotation, the computation task faces the challenge of dealing with a substantially large N. From Chap. 10–12, we discuss parallel algorithms, which can employ thousands of CPUs to achieve near-linear speedup, and indexing methods, which can substantially reduce retrieval time.

1.5 Multimodal Fusion

Multimedia metadata can be collected from multiple channels or sources. For instance, a video clip consists of visual, audio, and caption signals. Besides, a Web page where the video clip is embedded, and the users who have viewed the video can provide contextual signals for analyzing that clip. When mapping features extracted from multiple sources to semantics, a fusion algorithm must incorporate useful information while removing noise. Chapters 6, 7 and 8 are devoted to address multimodal fusion.

Chapter 6 focuses on addressing two questions: (1) what are the *best* modalities? and (2) how can we optimally fuse information from multiple modalities? Suppose we extract l, m, n features from the visual, audio, and caption tracks of videos. At one extreme, we could treat all these features as one modality and form a feature vector of $l + m + n$ dimensions. At the other extreme, we could treat each of the $l + m + n$ features as one modality. We could also regard the extracted features from each media-source as one modality, formulating a visual, audio, and caption modality with $l, m,$ and n features, respectively. Almost all prior multimodal-fusion work in the multimedia community employs one of these three approaches. But, can any of these feature compositions yield the optimal result?

Statistical methods such as principle component analysis (PCA) and independent component analysis (ICA) have been shown to be useful for feature transformation and selection. PCA is useful for denoising data, and ICA aims to transform data to a space of independent axes (components). Despite their best attempt under some error-minimization criteria, PCA and ICA do not guarantee to produce independent components. In addition, the created feature space may be of very high dimensions and thus be susceptible to the *curse of dimensionality*. Chapter 6 first presents an *independent modality analysis* scheme, which identifies independent modalities, and at the same time, avoids the curse-of-dimensionality challenge. Once a good set of modalities has been identified, the second research challenge is to fuse these modalities in an optimal way to perform data analysis (e.g., classification). Chapter 6 presents the *super-kernel fusion* scheme to fuse individual modalities in a non-linear way. The *super-kernel fusion* scheme finds the best combination of modalities through supervised training.

Chapter 6 addresses the problem of fusing multiple modality of multimedia data *content*. Chapter 7 addresses the problem of fusing *context* with *content*. Semantic labels can be roughly divided into two categories: wh labels and non-wh labels. Wh-semantics include time (when), people (who), location (where), landmarks (what), and event (inferred from when, who, where, and what). Providing the when and where information is trivial. Already cameras can provide time, and we can easily infer an approximate location from GPS or CellID. However, determining the what and who requires contextual information in addition to time, location, and photo content. More precisely, contextual information can include time, location, camera parameters, user profile, and even social graphs. Content of images consists of perceptual features, which can be divided into holistic features (e.g., color, shape and texture characteristics of an image), and local features (edges and salient points of regions or objects in an image). Besides context and content, another important source of information (which has been largely ignored) is the relationships between semantic labels (which we refer to as semantic ontology). To explain the importance of having a semantic ontology, let us consider an example with two semantic labels: outdoor and sunset. When considering contextual information alone, we may be able to infer the outdoor label from camera parameters: focal length and lighting condition.

We can infer sunset from time and location. Notice that inferring outdoor and sunset do not rely on any common contextual modality. However, we can say that

a sunset photo is outdoor with certainty (but not the other way). By considering semantic relationships between labels, photo annotation can take advantage of contextual information in a "transitive" way.

To fuse context, content, and semantic ontology in a synergistic way, Chap. 7 presents EXTENT, an inferencing framework to generate semantic labels for photos. EXTENT uses an influence diagram to conduct semantic inferencing. The variables on the diagram can either be decision variables (i.e., causes) or chance variables (i.e., effects). For image annotation, decision variables include time, location, user profile, and camera parameters. Chance variables are semantic labels. However, some variables may play both roles. For instance, time can affect some camera parameters (such as exposure time and flash on/off), and hence these camera parameters are both decision and chance variables. Finally, the influence diagram connects decision variables to chance variables with arcs weighted by causal strength.

To construct an influence diagram, we rely on both domain knowledge and data. In general, learning such a probabilistic graphical model from data is an NP hard problem. Fortunately, for image annotation, we have abundant prior knowledge about the relationships between context, content, and semantic labels, and we can use them to substantially reduce the hypothesis space to search for the right model. For instance, time, location, and user profile, are independent of each other. Camera parameters such as exposure time and flash on/off depend on time, but are independent of other modalities. The semantic ontology provides us the relationships between words. The only causal relationships that we must learn from data are those between context/content and semantic labels (and their causal strengths).

Once causal relationships have been learned, causal strengths must be accurately accounted for. Traditional probabilistic graphical models such as Bayesian networks use conditional probability to quantify the correlation between two variables. Unfortunately, conditional probability characterizes *covariation*, not *causation* [14–16]. A basic tenet of classical statistics is that correlation does not imply causation. Instead, we use recently developed *causal-power* theory [17] to account for causation. We show that fusing context and content using causation achieves superior results over using correlation.

Finally, Chap. 8 presents a fusion model called combinational collaborative filtering (CCF) using a latent layer. CCF views a community of common interests from two simultaneous perspectives: *a bag of users* and *a bag of multimodal features*. A community is viewed as a bag of participating users; and at the same time, it is viewed as a bag of multimodal features describing that community. Traditionally, these two views are independently processed. Fusing these two views provides two benefits. First, by combining *bags of features* with *bags of users*, CCF can perform *personalized* community recommendations, which the *bags of features* alone model cannot. Second, augmenting *bags of users* with *bags of features*, CCF improves information density to perform more effective recommendations. Though the chapter uses community recommendation as an application, one can use the CCF scheme for recommending any objects, e.g., images, videos, and songs.

1.6 Indexing

With the vast volume of data available for search, indexing is essential to provide scalable search performance. However, when data dimension is high (higher than 20 or so), no nearest-neighbor algorithm can be significantly faster than a linear scan of the entire dataset. Let n denote the size of a dataset and d the dimension of data, the theoretical studies of [18–21] show that when $d \gg \log\ n$, a linear search will outperform classic search structures such as k-d-trees [22], SR-trees [23], and SS-trees [24]. Several recent studies (e.g., [19, 20, 25]) provide empirical evidence, all confirming this phenomenon of *dimensionality curse*.

Nearest neighbor search is inherently expensive, especially when there are a large number of dimensions. First, the search space can grow exponentially with the number of dimensions. Second, there is simply no way to build an index on disk such that all nearest neighbors to any query point are physically adjacent on disk. The prohibitive nature of exact nearest-neighbor search has led to the development of *approximate nearest-neighbor search* that returns instances approximately similar to the query instance [18, 26]. The first justification behind approximate search is that a feature vector is often an approximate characterization of an object, so we are already dealing with approximations [27]. Second, an approximate set of answers suffices if the answers are relatively close to the query concept. Of late, three approximate indexing schemes, *locality sensitive hashing* (LSH) [28], M-trees [29], and clustering [27] have been employed in applications such as image-copy detection [30] and bio-sequence-data matching [31]. These approximate indexing schemes speed up similarity search significantly (over a sequential scan) by slightly lowering the bar for accuracy.

In Chap. 11, we present our *hypersphere indexer*, named SphereDex , to perform approximate nearest-neighbor searches. First, the indexer finds a roughly central instance among a given set of instances. Next, the instances are partitioned based on their distances from the central instance. SphereDex builds an *intra-partition* (or local) index within each partition to efficiently prune out irrelevant instances. It also builds an *inter-partition* index to help a query to identify a good starting location in a neighboring partition to search for nearest neighbors. A search is conducted by first finding the partition to which the query instance belongs. (The query instance does not need to be an existing instance in the database.) SphereDex then searches in this and the neighboring partitions to locate nearest neighbors of the query. Notice that since each partition has just two neighboring partitions, and neighboring partitions can largely be sequentially laid out on disks, SphereDex can enjoy sequential IO performance (with a tradeoff of transferring more data) to retrieve candidate partitions into memory. Even in situations (e.g., after a large batch of insertions) when one sequential access might not be feasible for retrieving all candidate partitions, SphereDex can keep the number of non-sequential disk accesses low. Once a partition has been retrieved from the disk, SphereDex exploits geometric properties to perform intelligent intra-partition pruning so as to minimize the computational cost for finding the top-k approximate nearest neighbors. Through empirical studies on two very large,

high-dimensional datasets, we show that SphereDex significantly outperforms both LSH and M-trees in both IO and CPU time. Though we mostly present our techniques for approximate nearest-neighbor queries, Chap. 11 also briefly describes the extensibility of SphereDex to support farthest-instance queries, especially hyperplane queries to support key data-mining algorithms like SVMs.

1.7 Scalability

Indexing deals with retrieval scalability. We must also address scalability of learning, both supervised and unsupervised. Since 2007, we have parallelized five mission-critical algorithms including SVMs [32], frequent itemset mining [33], spectral clustering [34], probabilistic latent semantic analysis (PLSA) [35], and latent dirichlet allocation (LDA) [36]. In this book, we present parallel support vector machines (PSVM) in Chap.10 and an enhanced PLDA+ in Chap.12.

Parallel computing has been an active subject in the distributed computing community over several decades. In PSVM, we use Incomplete Cholesky Factorization to approximate a large matrix so as to reducing the memory use substantially. For speeding up LDA, we employ data placement and pipeline processing techniques to substantially reduce the communication bottleneck. We are able to achieve 1,500 speedup when 2,000 machines are simultaneously used: i.e., a two-month computation task on a single machine can now be completed in an hour. These parallel algorithms have been released to the public via Apache open source (please check out the Appendix).

1.8 Concluding Remarks

As we stated in the beginning of this chapter, multimedia information management research is multidisciplinary. In feature extraction and distance function formulation, the disciplines of computer vision, psychology, cognitive science, neural science, and database have been involved. In indexing and scalability, distributed computing and database communities have contributed a great deal. In devising learning algorithms to bridge the semantic gap, machine learning and neural science are the primary forces behind recent advancements. Together, all these communities are increasingly working together to develop robust and scalable algorithms. In the remainder of this book, we detail the design and implementation of these key subroutines of multimedia data management.

References

1. E.Y. Chang, Extent: Fusing context, content, and semantic ontology for photo annotation, in *Proceedings of ACM Workshop on Computer Vision Meets Databases (CVDB) in conjunction with ACM SIGMOD*, 5–11 (2005)
2. D.H. Hubel, T.N. Wiesel, Receptive fields and functional architecture of monkey striate cortex. J. Physiol. **195**(1), 215–243 (1968)
3. H. Lee, R. Grosse, R. Ranganath, A. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in *Proceedings of International Conference on Machine Learning (ICML)* (2009)
4. T. Serre, Learning a dictionary of shape-components in visual cortex: comparison with neurons, humans and machines. PhD thesis, Massachusetts Institute of Technology ,2006
5. A. Tversky, Feature of similarity. Psychol. Review **84**, 327–352 (1977)
6. X.S. Zhou, T.S, Huang, Comparing discriminating transformations and svm for learning during multimedia retrieval. in *Proceeding of ACM Conference on Multimedia*, 137–146 (2001)
7. G. Murphy, D. Medin, The role of theories in conceptual coherence. Psychol. Review **92**, 289–316 (1985)
8. R.L. Goldstone, Similarity, interactive activation, and mapping. J. Exp. Psychol.: Learing, Memory, and Cognition **20**, 3–28 (1994)
9. C.C. Aggarwal, Towards systematic design of distance functions for data mining applications, in *Proceedings of ACM SIGKDD*, 9–18 (2003)
10. R. Fagin, R. Kumar, D. Sivakumar, Efficient similarity search and classification via rank aggregation, in *Proceedings of ACM SIGMOD Conference on Management of Data*, 301–312, June 2003
11. T. Wang, Y. Rui, S.M. Hu, J.Q. Sun, Adaptive tree similarity learning for image retrieval. Multimed. Syst. **9**(2), 131–143 (2003)
12. Y. Rui, T. Huang, Optimizing learning in image retrieval, in *Proceedings of IEEE CVPR*, 236–245, June 2000
13. S. Tong, E. Chang, Support vector machine active learning for image retrieval, in *Proceedings of ACM International Conference on Multimedia*,107–118, October 2001
14. D. Heckerman, A bayesian approach to learning causal networks, in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 107–118 (1995)
15. J. Pearl, Causality: Models, Reasoning and Inference. (Cambridge University Press, Cambridge, 2000)
16. J. Pearl, Causal inference in the health sciences: A conceptual introduction. Health Serv. Outcomes Res. Methodol. **2**, 189–220 (Special issue on causal inference, Kluwer Academic Publishers, 2001)
17. L.R. Novick, P.W. Cheng, Assessing interactive causal influence. Psychol. Review **111**(2), 455–485 (2004)
18. S. Arya, D. Mount, N. Netanyahu, R. Silverman, A. Wu , An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, in *Proceedings of the 5th SODA*, 573–82 (1994)
19. P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in *Proceedings of VLDB*, 604–613 (1998)
20. J.M. Kleinberg , Two algorithms for nearest-neighbor search in high dimensions, *in Proceedings of the 29th STOC*, (1997)
21. R. Weber, H.J. Schek, S. Blott, A quantitative analysis and performance study for similarity search methods in high-dimensional spaces, in *Proceedings 24th International Conference Very Large Data Bases VLDB* 194–205 (1998)
22. J. Bentley, Multidimensional binary search trees used for associative binary searching. Commun. of ACM **18**(9), 509–517 (1975)
23. N. Katayama, S. Satoh, The SR-tree: An index structure for high-dimensional nearest neighbor queries, in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 369–380 (1997)

24. D.A. White, R. Jain, Similarity indexing with the SS-Tree, in *Proceedings of IEEE ICDE*, 516–523 (1996)
25. E. Kushilevitz, R. Ostrovsky, Y. Rabani, Efficient search for approximate nearest neighbor in high dimensional spaces, in *Proceedings of the 30th STOC*, 614–623 (1998)
26. K. Clarkson, An algorithm for approximate closest-point queries, in *Proceedings of the 10th SCG*, 160–164 (1994)
27. C. Li, E. Chang, H. Garcia-Molina, G. Wilderhold, Clindex: Approximate similarity queries in high-dimensional spaces. IEEE Transactions on Knowledge and Data Engineering (TKDE), **14**(4),792–808, July 2002
28. A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing. VLDB J., 518–529 (1999)
29. P. Ciaccia, M. Patella, Pac nearest neighbor queries: Approximate and controlled search in high-dimensional and metric spaces, in *Proceedings of IEEE ICDE*, 244–255 (2000)
30. A. Qamra, Y. Meng, E.Y. Chang, Enhanced perceptual distance functions and indexing for image replica recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **27**(3) (2005)
31. J. Buhler, Efficient large-scale sequence comparison by locality-sensitive hashing. Bioinformatics **17**, 419–428 (2001)
32. E.Y. Chang, K. Zhu, H. Wang, H. Bai, J. Li, Z. Qiu, H. Cui, Parallelizing support vector machines on distributed computers, in *Proceedings of NIPS* (2007)
33. H. Li, Y. Wang, D. Zhang, M. Zhang, E.Y. Chang, PFP: Parallel fp-growth for query recommendation, in *Proceedings of ACM RecSys*, 107–114 (2008)
34. Y. Song, W. Chen, H. Bai, C.J. Lin, E.Y. Chang, Parallel spectral clustering, in *Proceedings of ECML/PKDD*, 374–389 (2008)
35. W. Chen, D. Zhang, E.Y. Chang, Combinational collaborative filtering for personalized community recommendation, in *Proceedings of ACM KDD*, 115–123 (2008)
36. Z. Wang, Y. Zhang, E.Y. Chang, M. Sun, PLDA+ parallel latent dirichlet allocation with data placement and pipeline processing. ACM Trans. Intel. Syst. Technol. **2**(3) (2011)