Chapter 6
Multimodal Fusion

Abstract Multimedia data instances consist of metadata from multiple sources. Given a set of features extracted from these sources (e.g., features extracted from the visual, audio, and caption track of videos), how do we determine the best modalities? Once a set of modalities has been identified, how do we best fuse them to map to semantics? This chapter[†] presents a two-step approach. The first step finds *statistically independent modalities* from raw features. In the second step, we use *super-kernel fusion* to determine the optimal combination of individual modalities. We carefully analyze the tradeoffs between three design factors that affect fusion performance: *modality independence*, *curse of dimensionality*, and *fusion-model complexity*. Through analytical and empirical studies, we demonstrate that the two-step approach, which achieves a careful balance of the three design factors, can improve class-prediction accuracy over traditional techniques.

Keywords Feature combination · Multimodal fusion · PCA · ICA · Super kernel

6.1 Introduction

Multimedia data such as images and videos are represented by features from multiple media sources. Traditionally, images are represented by keywords and perceptual features such as color, texture, and shape [2, 3]. Videos are represented by features embedded in the tracks of visual, audio, caption text, etc. [4]. Besides, contextual information associated with a data instance, such as camera parameters, user profile, social interactions, and search logs, can also be considered for analyzing

[†] © ACM, 2004. This chapter is a minor revision of the author's work with Yi Wu, Kevin Chang, and John R. Smith [1] published in MULTIMEDIA'04. Permission to publish this chapter is granted under copyright license #2587660035739.

E. Y. Chang, *Foundations of Large-Scale Multimedia Information Management and Retrieval*, DOI: 10.1007/978-3-642-20429-6_6,

multimedia data. These features are extracted and then fused in a complementary way for understanding the semantics of multimedia data.

Traditional work on multimodal integration has largely been heuristic-based. It lacks theories to answer two fundamental 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 [5, 6]. 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 axises (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*.[1] In the first part of this chapter, we present 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). Suppose we can yield truly independent modalities, and each modality can derive accurate posterior probability for class prediction. We can simply use the *product-combination* rule to multiply the probabilities for predicting class membership. Unfortunately, the above two conditions do not hold in general for a multimedia data-analysis task (see Sect. 6.2 for detailed discussion). Using the product-combination rule to fuse information is thus inappropriate. Another popular fusion method is the *weighted-sum* rule, which performs a linear combination on the modalities. The weighted-sum rule enjoys the advantage of simplicity, but its linear constraint forbids high model complexity; hence it cannot adequately explore the inter-dependencies left unresolved by PCA and ICA. In this chapter, we present a discriminative approach (whereas in Chap. 8 we present a generative approach) to address multimodal fusion. Our discriminative approach employs the *super-kernel fusion* scheme to fuse individual modalities in a non-linear way (linear fusion is a special case of our method). The *super-kernel fusion* scheme finds the best combination of modalities through supervised training.

[1] The work of [7] shows that, when data dimension is high, the distances between pairs of objects in the space become increasingly similar to each other due to the *central limit theory*. This phenomenon is called the *dimensionality curse* [8], because it can severely hamper the effectiveness of data analysis.

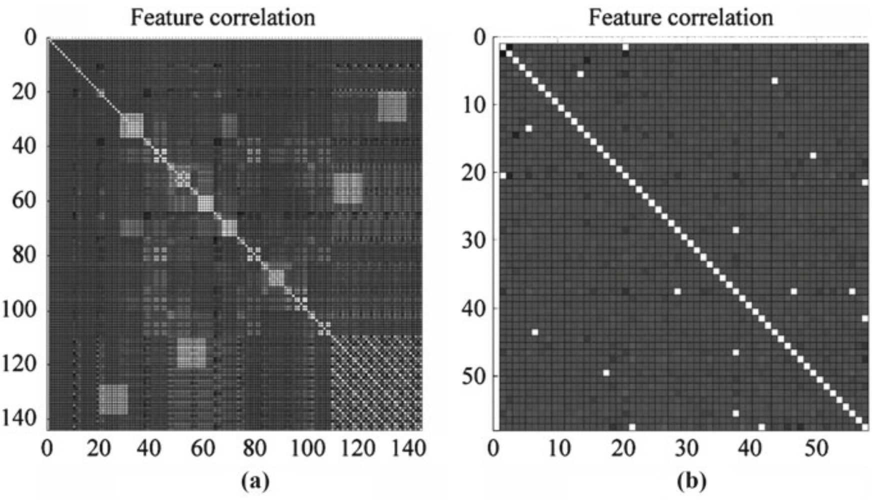

Fig. 6.1 Feature correlation matrix. **a** Before PCA/ICA, **b** after PCA/ICA

Let us use a simple example to explain the shortcomings of some traditional multimodal integration schemes that invite further research. Figure 6.1 shows the existence of feature dependencies in a real image dataset, before and after performing PCA/ICA. This figure plots the normalized correlation matrix in absolute value derived from a 2K-image dataset of 14 classes. (Detailed description for this image dataset is given in Sect. 6.5.) A total of 144 features are considered: the first 108 are color features; the other 36 are texture features. Correlation between features within the same media source and across different media sources is measured by computing the covariance matrix:

$$C = \frac{1}{N} \sum_{\mathbf{x}_i \in X} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad \text{with } \bar{\mathbf{x}} = \frac{1}{N} \sum_{\mathbf{x}_i \in X} \mathbf{x}_i, \tag{6.1}$$

where N is the total number of sample data, $\mathbf{x}_i$ is a feature vector to represent the ith sample, and X is the set of feature vectors for N samples. Normalized correlation between features i and j is defined by

$$\hat{C}(i, j) = \frac{C(i, j)}{\sqrt{C(i, i) \times C(j, j)}}. \tag{6.2}$$

In the figure, both the x- and y-axis depict the 144 features. The light-colored areas in the figure indicate high correlation between features, and the dark-colored areas indicate low correlation. If any feature correlates only with itself, only the diagonal elements will be light-colored. The off-diagonal light-colored areas in Fig. 6.1a indicate that this image dataset exhibits not only a high correlation of features within the same media source, but also between certain features from different media sources

Table 6.1 Related work summarization

No. of modality	Fusion methods	Evaluation
1	No	No need to do fusion; curse of dimensionality
m	Any	Loss of inter-dependency relationship between features
k	Any	High model complexity; no perfect independent components
	Product	Very sensitive to the accuracy of individual classifiers
D	Linear	Not suitable for independent feature spaces
	Super-kernel	Suitable

m no. of media sources, *k* no. of independent components, *D* no. of independent modalities

(e.g., color and texture). Color and texture are traditionally treated as orthogonal modalities, but this example shows otherwise. These correlated and even noisy "raw" features may affect the learning algorithm by obscuring the distributions of truly relevant and representative features. (The weighted-sum fusion rule cannot deal with these inter-dependencies.)

Figure 6.1b presents the feature correlation matrix after we applied both PCA and ICA to the data. The process yields 58 "improved" components. Although the components exhibit better independence, inter-dependencies between components still exist. This chapter first deals with grouping components like these 58 into a smaller number of independent modalities to avoid the *dimensionality curse*. We then explore non-linear combinations of the modalities to improve the effective multimodal fusion.

As the main contribution of this work, we propose a discriminative fusion scheme for multimedia data analysis. Given a list of features extracted from multiple media-sources, we tackle two core issues:

- Formulating independent feature modalities (Sect. 6.3).
- Fusing multiple modalities optimally (Sect. 6.4).

We carefully analyze the tradeoffs between three design factors that affect fusion performance: *modality independence*, *curse of dimensionality*, and *fusion-model complexity*. Through analytical and empirical studies on an image dataset and TREC-Video 2003 benchmarks, we show that a careful balance of the three design factors consistently leads to superior performance for multimodal fusion.

6.2 Related Reading

We discuss related work in *modality identification* and *modality fusion* (Table 6.1).

6.2.1 Modality Identification

Let D denote the number of modalities. Given $d_1, d_2, \ldots, d_m$ features extracted from m media sources, respectively, prior modality identification work can be divided into two representative categories.

1. $D = 1$, or treating all features as one modality. This approach does not require the fusion step. Goh et al. [9] used the raw color and texture features to form a high-dimensional feature vector for each image. Recently, statistical methods such as PCA and ICA have been widely used in the Computer Vision, Machine Learning, Signal Processing communities to denoise data and to identify independent information sources (e.g., [10–13]). In the multimedia community, the work of [14, 15] observed that audio and visual data of a video stream exhibit some statistical regularity, and that regularity can be explored for joint processing. Smaragdis et al. [16] proposed to operate on a fused set of audio/visual features and to look for combined subspace components amenable to interpretation. Vinokourov et al. [17] found a common latent/semantic space from multi-language documents using independent component analysis for cross-language document retrieval. The major shortcoming of these works is that the curse of dimensionality arises, causing ineffective feature-to-semantics mapping and inefficient indexing [2]. (Please refer to [7, 18, 19] for the discussion of dimensionality-curse and why dimension reduction can greatly enhance the effectiveness of statistical analysis and the efficiency of query processing.)
2. $D = m$, or treating each source as one modality. This approach treats the features as m modalities, with d_i features in the ith modality ($i = 1, 2, \ldots, m$). Most work in image and video retrieval analysis (e.g., [4, 20–23]) employs this approach. For example, the QBIC system [20] supported image queries based on combining distances from the color and texture modalities. Velivelli et al. [23] separated video features into audio and visual modalities. Adams et al. [4] also regarded each media track (visual, audio, textual, etc.) as one modality. For each modality, these works trained a separate classification model, and then used the weighted-sum rule to fuse a class-prediction decision. This modality-decomposition method can alleviate the "curse of dimensionality." However, since media sources are treated separately, the inter-dependencies between sources are left unexplored.

Our method is to apply independent component analysis on the raw feature sets to identify k "independent" components. Thereafter, we group these components into D modalities to (1) minimize the dependencies between modalities, and (2) mitigate the dimensionality-curse problem.

6.2.2 Modality Fusion

Given that we have obtained D modalities, we need to fuse D classifiers, one for each modality, for interpreting data.

PCA and ICA cannot perfectly identify independent components for at least two reasons. First, like the way that the k-mean algorithm works, all well-known ICA algorithms (fixed-point algorithm [24], Infomax [25, 26], kernel canonical analysis [17], and kernel independent analysis [27]) need a good estimate of the number of independent components k to find them effectively. Second, as we discussed in Sect. 6.1, ICA only performs the best attempt under some error-minimization criteria to find k independent components. But the resulting components, as shown in Fig. 6.1b, may still exhibit inter-dependencies.

Now, given D modalities, not entirely independent of each other, we need an effective fusion strategy. Various fusion strategies for multimodal information have been presented and were discussed in [28], including *product combination*, *weighted-sum*, *voting*, and *min–max aggregation*. Among them, *product combination* and *weighted-sum* are by far the most popular fusion methods.

1. *Product combination*. Supposing that D modalities are independent of each other, and we can estimate posterior probability for each modality accurately, the product-combination rule is the optimal fusion model from the Bayesian perspective. However, in addition to the fact that we will not have D truly independent modalities, we generally cannot estimate posterior probability with high accuracy. The work of [29] concluded that the product-combination rule works well only when the posterior probability of individual classifiers can be accurately estimated. In a multimedia data-understanding task, we often assert similarity between data based on our beliefs. (E.g., one can "believe" two videos to be 87% similar or 90% similar. This estimate does not come from classical probability experiments, so the sum of beliefs may not be equal to one.) Because of this subjective process, and because the product-combination rule is highly sensitive to noise, this strategy is not appropriate.
2. *Weighted-sum*. The weighted-sum strategy is more tolerant to noise because *sum* does not magnify noise as severely as *product*. Weighted-sum (e.g., [30]) is a linear model, not equipped to explore the inter-dependencies between modalities. Recently, Yan and Hauptmann [31] presented a theoretical framework for bounding the average precision of a linear combination function in video retrieval. Concluding that the linear combination functions have limitations, they suggested that non-linearity and cross-media relationships should be introduced to achieve better performance.

In this chapter, we depict a super-kernel scheme, which can fuse multimodal information non-linearly to explore the cross-modality relationship. Chapters 7 and 8 present two generative schemes. Both discriminative and generative models enjoy their pros and cons, which we will discuss throughout these three chapters.

6.3 Independent Modality Analysis

In this section, we present our approach to transform m raw features to D modalities. Given input in the form of an $m \times n$ matrix X (n denotes the number of training instances), our independent modality analysis procedure produces $M_1, M_2, \ldots, M_D$ modalities. The procedure consists of the following three steps:

1. Run principal component analysis (PCA) on X to remove noise and reduce the feature dimensionality. Let U denote the matrix containing the first k eigenvectors. The PCA representation of zero-mean feature vectors X is defined as $U^T X$.
2. Run independent component analysis (ICA) on the PCA output $U^T X$ to obtain estimates of independent feature components S and an estimate of a mixing matrix W. We can recover the independent components by computing $S = WU^T X$.
3. Run independent modality grouping (IMG) on S to form independent modalities $M_1, M_2, \ldots, M_D$.

6.3.1 PCA

PCA has been frequently used as a technique for removing noises and redundancies between feature dimensions [32]. PCA projects the original data to a lower dimensionality space such that the variance of the data is best maintained. Let's assume that we have n samples, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, and each $\mathbf{x}_i$ is an m-dimensional vector. We can represent the n samples as a matrix $X_{m\times n}$. It is known in linear algebra that any such matrix can be decomposed in the following form (known as singular value decomposition or SVD):

$$X = UDV^T,$$

where matrices $U_{m\times p}$ and $V_{n\times p}$ represent orthonormal basis vectors matrices (eigenvectors of the symmetric matrix XX^T and $X^T X$), with p as the number of largest principal components. The $D_{p\times p}$ matrix is a diagonal matrix, and the diagonal elements of D are the eigenvalues of XX^T and $X^T X$. Consider the projection onto the subspace spanned by the p largest principal components (PC's), i.e., $U^T X$.

6.3.2 ICA

Compared to PCA, the spirit of ICA is to find statistically independent hidden sources from a given set of mixture signals. Both ICA and PCA project data matrices into components in different spaces. However, the goals of the two methods are different. PCA finds the uncorrelated components of maximum variance. It is ideal for compressing data into a lower-dimensional space by removing the least significant components. ICA finds the statistically independent components. ICA is the ideal choice for separating mixed signals and finding the most representative components.

To formalize an ICA problem, we assume that there are k unknown independent components $S = \{s_1, s_2, \ldots, s_k\}$. What we observe is a set of m-dimensional samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, which are mixture signals coming from k independent components, $k \leq m$. We can represent all the observation data as a matrix $X_{m\times n}$. A linear mixture model can be formulated as:

$$X = AS,$$

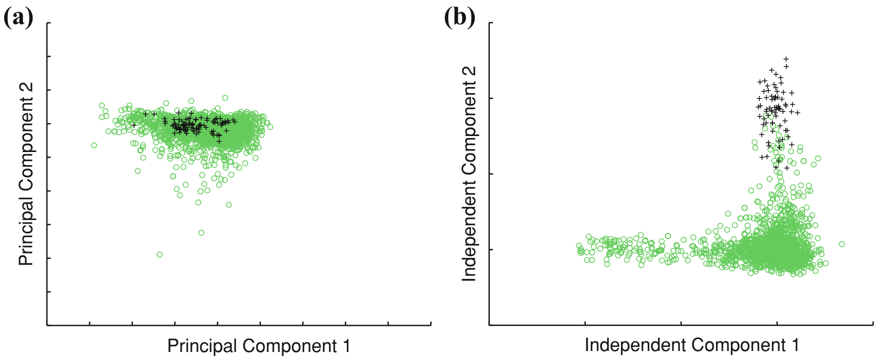

Fig. 6.2 Scatter plots of the 2K image dataset. **a** PCA, **b** ICA

where $A_{m \times k}$ is a mixing matrix. Our goal is to find $W = A^{-1}$; therefore, given training set X, we can recover the independent components (IC's) through the transformation of $S = WX$.

ICA establishes a common latent space for the media, which can be viewed as a method for learning the inter-relations between the involved media [16, 33]. For multimedia data, observation data $\mathbf{x}_i$ usually contains features coming from more than one medium. The different independent components $\{s_1, s_2, \ldots, s_k\}$ provide a meaningful segmentation of the feature space. The kth column of W^{-1} constitutes the original multiple features associated with the kth independent component. These independent components can provide a better interpretation for multimedia data. Figure 6.2a, b show the scatter plots of the 2K image dataset, projected to a two-dimensional subspace identified by the first two principal components and the first two independent components. Dark points correspond to the class of *tools* (one of the 14 classes), and green (light) points correspond to the other 13 classes. Compared with PC's in Fig. 6.2a, IC's found from ICA in Fig. 6.2b can better separate data from different semantic classes. Figure 6.2b strongly suggests an ICA interpretation to differentiate semantics. The main attraction of ICA is that it provides unsupervised groupings of data that have been shown to be well aligned with manual grouping in different media [11]. The representative and non-redundant feature representations form a solid base for later processing.

Lacking any prior information about the number of independent components, ICA algorithms usually assume that the number of independent components is the same as the dimension of observed mixtures, that is, $k = m$. PCA technique can be used as preprocessing to ICA to reduce noise in the data and control the number of independent components [34]. Then ICA is performed on the main eigenvectors of PCA representations ($k = p$, where p is the number of PC's) to determine which PC's actually are independent and which should be grouped together as parts of a multidimensional component. Finally, the independent components are recovered by computing $S = WU^T X$.

6.3.3 IMG

As discussed in Sect. 6.1, though ICA makes a best attempt to find independent components, the resulting k components might not be independent, and the number of components can be too large to face the challenge of "dimensionality curse" during the statistical-analysis and query-processing phrases. IMG aims to remedy these two problems by grouping k components into D modalities.

We divide k components into D groups to satisfy two requirements: (1) the correlation between modalities is minimized, and (2) the number of features in each modality is not too large. The first requirement maximizes modality independence. The second requirement avoids the problem of curse-of-dimensionality. To decide on D, we place a soft constraint on the number of components that a modality can have. We set the soft constraint as 30 because several prior works [7, 18, 19] indicate that when the number of dimensions exceeds 20–30, the curse starts to kick in. Since only the data can tell us exactly at what dimension the curse starts to take effect, the selection of D must go through a cross-validation process: we pick a small number of candidate D values and rely on experiments to select the best D.

For a given D, we employ a clustering approach to divide k into D groups. Ding et al. [35] provided theoretical analysis to show that minimizing inter-subgraph similarities and maximizing intra-subgraph similarities always lead to more balanced graph partitions. Thus, we apply *minimizing inter-group feature correlation* and *maximizing intra-group feature correlation* as our feature-grouping criteria to determine independent modalities. Suppose we have D modalities $M_1, M_2, \ldots, M_D$, each containing a number of feature components. The inter-group feature correlation between two modalities M_i and M_j is defined as

$$C(M_i, M_j) = \sum_{\forall\, S_i \in M_i,\, \forall\, S_j \in M_j} C(S_i, S_j), \tag{6.3}$$

where S_i and S_j are features belonging to modalities M_i and M_j respectively, and $C(S_i, S_j)$ is the normalized feature correlation between S_i and S_j. $C(S_i, S_j)$ can be calculated using (6.1) and (6.2). The intra-group feature correlation within modality M_i is defined as

$$C(M_i) = C(M_i, M_i). \tag{6.4}$$

To minimize inter-group feature correlation while maximizing intra-group feature correlation at the same time, we can formulate the following objective function for grouping all the features into D modalities,

$$\min \sum_{\substack{i=1 \\ j>i}}^{D} \left[\frac{C(M_i, M_j)}{C(M_i)} + \frac{C(M_i, M_j)}{C(M_j)} \right]. \tag{6.5}$$

Solving this objective function yields D modalities, with minimal inter-modality correlation and balanced features in each modality.

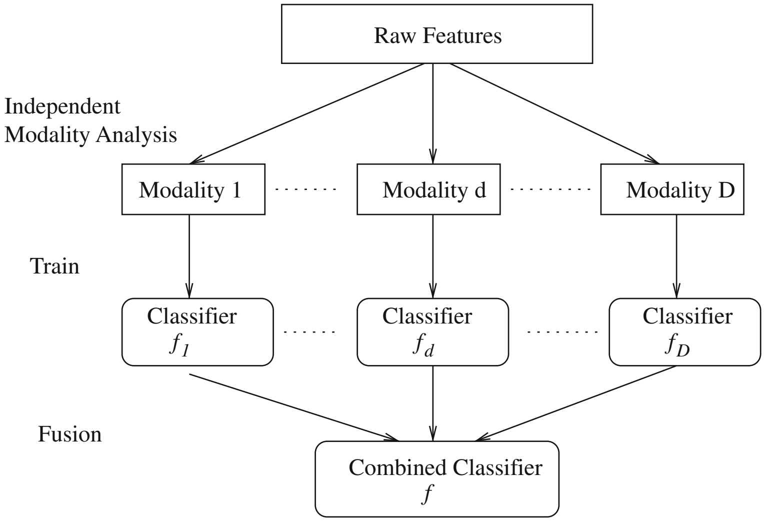

Fig. 6.3 Fusion architecture

6.4 Super-Kernel Fusion

Once D modalities have been identified by our independent modality analysis, we need to fuse multimodal information optimally. Suppose we train for the dth modality classifier f_d. We need to combine these D classifiers to perform class prediction for query instance $\mathbf{x}_q$. The fusion architecture is depicted in Fig. 6.3.

After $f_d,\ d = 1, \ldots, D$ have been trained, the information can be fused in several ways. Let f denote the fused classification function. The product-combination rule can be formulated as

$$f = \prod_{d=1}^{D} f_d.$$

And the most widely used weighted-sum rule can be depicted as

$$f = \sum_{d=1}^{D} \mu_d f_d,$$

where μ_d is the weight for individual classifier f_d. As we have discussed in Sect. 6.2, both these popular models suffer from several shortcomings, including being sensitive to prediction error and being limited by the linear-model complexity. (Please consult Sect. 6.2 for detailed discussion.) To overcome these shortcomings, we propose using *super-kernel fusion* to aggregate f_d's.

The algorithm of super-kernel fusion is summarized in Fig. 6.4, which consists of the following three steps:

Fig. 6.4 Super-kernel fusion algorithm

```
Algorithm Super-kernel Fusion
Input:
  $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$;  /* A set of training data
  $Y = \{y_1, y_2, \cdots, y_n\}$;  /* Labels of training data

Output:
  $f$;  /* Class-prediction function

Variable:
  $\{f_1, f_2, \cdots, f_D\}$;  /* A set of discriminative functions
  $\{M_1, M_2, \cdots, M_D\}$;  /* A set of $n \times |M_d|$ matrices
  $K$;  /* Super-kernel matrix with dimension of $n \times D$

Function calls:
  $f_d(\mathbf{x}_i^d)$;  /* Prediction score of $\mathbf{x}_i^d$ from $f_d$
  $Train(Matrix, Y)$;  /* Train a discriminative function
  $IMA(X)$;  /* Independent modality analysis
  $Prob(score)$;  /* Convert an SVM score to probability

Begin:
    /* Independent modality analysis to get D modality
    $\{M_1, M_2, \cdots, M_D\} \leftarrow IMA(X)$;
    /* Train classifiers for each modality
    for each $d = 1, 2, \cdots, D$
        $f_d \leftarrow Train(M_d, Y)$;
    /* Create super-kernel matrix K
    for each data $\mathbf{x}_i \in X$
        for each discriminative function $f_d$
            $K(i, d) \leftarrow Prob(f_d(\mathbf{x}_i^d))$;
    /* Super-kernel Fusion
    $f \leftarrow Train(K, Y)$;
    return $f$;
End
```

1. Train individual classifiers $\{f_d\}$. The inputs to the algorithm are the n training instances $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and their corresponding labels $\{y_1, y_2, \ldots, y_n\}$. After the independent modality analysis (IMA), the m-dimensional features are divided into D modalities. Each training instance $\mathbf{x}_i$ is represented by $\{\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^D\}$, where $\mathbf{x}_i^d$ is the feature representation for x_i in the dth modality. All the training instances are divided into D matrices $\{M_1, M_2, \ldots, M_D\}$, where each M_d is an $n \times |M_d|$ matrix, and $|M_d|$ is the number of features in the dth modality ($d = 1, 2, \ldots, D$). To train classifier f_d, we use M_d and the label information. Though many learning algorithms can be employed to train f_d, we employ an SVM as our base-classifier because of its effectiveness. For training

each f_d, the kernel function and kernel parameters are carefully chosen via cross validation (steps 1–3 in Fig. 6.4).

2. Estimate posterior probability. Once we have trained D classifiers for the D modalities, we create a super-kernel matrix K for modality fusion. This matrix is created by passing each training instance to each of the D classifiers to estimate its posterior probability. We use Platt's formula [36] to convert an SVM score to probability. As a result of this step, we obtain an $n \times D$ matrix consisting of n entries of D class-prediction probability (steps 4–6 in Fig. 6.4).
3. Fuse the classifiers. The *super-kernel* algorithm treats K a matrix of n training instances, each with a vector of D elements. Next, we again employ SVMs to train the super-classifier. The inputs to SVMs include K, training labels, a selected kernel function, and kernel parameters. At the end of the training process, we yield function f to perform class prediction. The complexity of the fusion model depends on the kernel chosen. For instance, we can select a polynomial, RBF or Laplacian function (steps 7–8 in Fig. 6.4).

Remark 6.1 A context-based query can be represented by a discriminative function f derived from the above supervised-learning process. Given a candidate data $\mathbf{x}$, the output of $f(\mathbf{x})$ indicates the degree of relevance that $\mathbf{x}$ has to the query. We apply f to the dataset and return top-k most relevant data as the query result.

At first it might seem that non-linear transformations would suffer from high model and computational complexity. But our proposed super-kernel fusion successfully avoids these problems by employing the *kernel trick*. (The kernel trick has been applied to several algorithms in statistics, including Support Vector Machines and kernel PCA.) The kernel trick let us generalize data similarity measurement to operate in a *projected space*, usually nonlinearly related to the *input space*. The *input space* (denoted as $\mathscr{I}$) is the original space in which data are located, and the *projected space* (denoted as $\mathscr{P}$) is that space to which the data are projected, linearly or non-linearly. The advantage of using the *kernel trick* is that, instead of explicitly determining the coordinates of the data in the projected space, the distance computation in $\mathscr{P}$ can be efficiently performed in $\mathscr{I}$ through a *kernel function*.[2] Specifically, given two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ is defined as the inner product of $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$, where Φ is a basis function that maps the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ from $\mathscr{I}$ to $\mathscr{P}$. The inner product between two vectors can be thought of as a measure of their similarity. Therefore, $K(\mathbf{x}_i, \mathbf{x}_j)$ returns the similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathscr{P}$. Since a kernel function can be either linear or nonlinear our super-kernel fusion scheme can model non-linear combinations of individual kernels.

One can employ any supervised learning algorithm is the function *Train* in the algorithm (line 2 in the figure). Algorithms that work well with kernel methods are Support Vector Machines [37] and Kernel Discriminative Analysis [38].

[2] Given a kernel function K, we can construct a corresponding kernel matrix $\mathbf{K}$, where $\mathbf{K}(i, j) = K(\mathbf{x}_i, \mathbf{x}_j)$.

Proposition 6.1 *Fused kernel matrix* $\underline{\mathbf{K}}$ *is a mathematically valid kernel, which is symmetric and positive semi-definite.*

Proof From Fig. 6.4, obviously, vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ have the same dimensions of D. Therefore, we can use traditional kernel functions such as *Gaussian radial basis kernel function*, *Laplacian kernel function*, and *Polynomial kernel function* to calculate the similarity between these vectors and to build the kernel matrix $\underline{\mathbf{K}}$. Those kernel functions have already been proven to be a mathematically valid kernel satisfying symmetric and positive semi-definite conditions [37]. The resulting kernel matrix $\underline{\mathbf{K}}$ is valid too. □

Finally, once the class-prediction function f has been trained, we can use the function to predict the class membership of a query point $\mathbf{x}_q$. Assume $\mathbf{x}_q$ is an m-dimensional feature vector in original feature space, we can convert it to an ICA feature representation $WU^T\mathbf{x}_q$, where W and U are transformation matrices obtained from PCA and ICA process, respectively (Sect. 6.3). Then, $WU^T\mathbf{x}_q$ is further divided into D modalities (information obtained from the IMG process), named as $\{\mathbf{x}_q^1, \mathbf{x}_q^2, \ldots, \mathbf{x}_q^D\}$. The class-prediction function for query point $\mathbf{x}_q$ can be written as

$$\hat{y}_q = f(f_1(\mathbf{x}_q^1), f_2(\mathbf{x}_q^2), \ldots, f_d(\mathbf{x}_q^D)).$$

6.5 Experiments

Our experiments were designed to evaluate the effectiveness of using *independent modality analysis* and *multimodal kernel fusion* to determine the optimal multimodal information fusion for multimedia data retrieval. Specifically, we wanted to answer the following questions:

1. Can independent modality analysis improve the effectiveness of multimedia data analysis?
2. Can super-kernel fusion improve fusion performance?

We conducted our experiments on two real-world datasets: one is a 2K image dataset, and the other is TREC-2003 video track benchmark. We randomly selected a percentage of data from the dataset to be used as training examples. The remaining data were used for testing. For each dataset, the training/testing ratio was empirically chosen via cross-validation so that the sampling ratio worked best in our experiments. To perform independent modality analysis, we applied traditional PCA and ICA algorithms[3] onto the given features (including all the training and testing data) to get the independent components following the steps described in Sect. 6.3. To perform class prediction, we employed the one-per-class (OPC) ensemble [39], which trains

[3] InfoMax was chosen as our ICA algorithm because of its robustness, though other ICA algorithms could also be applied.

all the classifiers, each of which predicts the class membership for one class. The class prediction on a testing instance is decided by voting among all the classifiers. The results presented here were the average of 10 runs.

- *Dataset* #1: 2*K image dataset*. The image dataset was collected from the Corel Image CDs. Corel images have been widely used by the computer vision, image processing, and multimedia research communities for conducting various experiments. This set contains 2K representative images from fourteen categories: *architecture, bears, clouds, elephants, fabrics, fireworks, flowers, food, landscape, people, textures, tigers, tools*, and *waves*. We tried different kernel functions, kernel parameters and training/testing ratios. Laplacian kernel with $\gamma = 0.001$ and 80% of the dataset as training data gave us the best results on the experiments of using raw features. We used the Laplacian kernel with $\gamma = 0.001$ for all subsequent experiments on this 2K image dataset. We randomly picked 80% of images for training and the remaining 20% were used for testing data. For each image, we extracted 144 features (documented in [40]) including color and texture features. This small dataset is used to provide insights into understanding the effectiveness of our methods, and the tradeoffs between design factors.
- *Dataset* #2: *TREC*-2003 *Video Track*. TREC-2003 video track used 133 h digital video (MPEG-1) from ABC and CNN news. The task is to detect the presence of the specified concept in video shots. The ground-truth of the presence of each concept was assumed to be binary (either present or absent in the data). Sixteen concepts are defined in the benchmark, including *airplane, animal, building, female speech, madeleine albright, nature vegetation, news subject face, news subject monologue, NIST non-studio setting, outdoors, people, physical violence, road, sport event, vehicle*, and *weather news*. The video concept detection benchmark is summarized as follows: 60% of the video shots were randomly chosen from the corpus to be used solely for the development of classifiers. The remaining 40% were used for concept validation.[4] RBF kernels with $\gamma = 0.0001$ gave us the best results on the experiments, so we used the same parameter settings in all subsequent experiments on this video dataset. For each video shot, we extracted a number of features [4]: *color histogram*, *edge orientation histogram*, *wavelet texture*, *color correlogram*, *co-occurrence texture*, *motion vector histogram*, *visual perception texture*, *Mel-frequency Cepstral coefficients*, *speech*, and *closed caption*.

6.5.1 Evaluation of Modality Analysis

The first set of experiments examined the effectiveness of independent modality analysis on the 2K image dataset. Table 6.2 compares five methods based on the classification accuracy results of 14 concepts: original 144 dimensional features

[4] IBM research center won most of the best concept models in the final TREC-2003 video concept competition. For the purpose of comparison, we employed the same training and testing data used by IBM.

Table 6.2 Classification accuracy (%) of image dataset

Category	Method 1	Method 2	Method 3	Method 4	Method 5
Architecture	88.00	89.95	90.77	95.38	**96.92**
Bears	74.70	76.72	75.00	75.00	**81.56**
Clouds	84.60	87.61	87.27	90.91	**92.32**
Elephants	83.90	84.67	84.83	87.21	**89.91**
Fabrics	85.10	85.90	87.22	87.82	**87.93**
Fireworks	93.50	95.69	94.91	96.46	**99.50**
Flowers	91.30	**95.53**	92.21	93.49	95.23
Food	92.20	95.58	93.36	95.76	**97.48**
Landscape	78.80	72.79	79.48	79.63	**81.82**
People	82.30	85.50	87.45	86.27	**89.36**
Textures	**96.50**	91.62	91.22	95.00	96.30
Tigers	91.50	92.34	91.13	92.64	**94.80**
Tools	99.50	98.15	96.74	**100.00**	99.20
Waves	86.10	89.49	84.71	87.27	**91.42**
Average	87.71	88.82	88.66	90.20	**92.70**

before any analysis (Method 1), super-kernel fusion using 108 dimensional color features and 36 dimensional texture features as 2 modalities (Method 2), 58 dimensional features after PCA (Method 3), 58 dimensional features after ICA (Method 4) and super-kernel fusion after IMG (Method 5).

As shown in the table, treating color and texture as two modalities improved the accuracy by around 1.0% compared to using raw feature representation. However, the accuracy was 4.0% lower than super-kernel fusion after IMG. This observation indicates that improvement can be made by using super-kernel fusion to cover the inter-dependency relationship between features. Moreover, after analyzing the statistical relationships between feature dimensions and getting rid of noise, super-kernel fusion can improve the performance much more. PCA improved accuracy by around 1.0% compared to the original feature format by reducing noise from features. ICA worked better than PCA, improving accuracy by 2.5% compared to the original feature format. However, the improvement is not significant, compared to the performance of super-kernel fusion after IMG. Independent modality analysis plus super-kernel fusion improved classification accuracy around 5.0% compared to the original feature representation. The result shows that the feature sets from independent modality analysis can better interpret the concepts, and super-kernel fusion can further incorporate information from multiple modalities. Next, we evaluated how to select optimal D and compared super-kernel fusion with other fusion methods.

6.5.2 Evaluation of Multimodal Kernel Fusion

The second set of experiments evaluated kernel fusion methods of combining multiple modalities. We grouped the "independent" components after PCA/ICA into

Table 6.3 Classification accuracy (%) of image dataset

Category	*D*	PC	LC	SKF
Architecture	2	96.40	96.53	**96.92**
Bears	2	76.10	75.35	**81.56**
Clouds	3	82.71	89.77	**92.32**
Elephants	2	86.11	80.91	**89.91**
Fabrics	2	85.11	87.46	**87.93**
Fireworks	2	97.63	99.13	**99.50**
Flowers	3	82.29	86.14	**95.23**
Food	2	93.45	89.53	**97.48**
Landscape	2	77.55	74.24	**81.82**
People	2	**90.71**	89.57	89.36
Textures	2	74.51	94.27	**96.30**
Tigers	3	87.31	**95.00**	94.80
Tools	2	91.48	94.20	**99.20**
Waves	2	86.92	82.13	**91.42**
Average	2.3	86.31	88.16	**92.70**

independent modalities and trained individual classifiers for each modality. We evaluated the effectiveness of multimodal kernel fusion on the 2k-image dataset and TREC-2003 video benchmark.

The optimal number of independent modalities D was decided by considering the tradeoff between dimensionality-curse and feature inter-dependency. Once D had been determined, feature components were grouped using the IMG algorithm in Sect. 6.3.3. When $D = 1$, all the feature components were treated as one vector representation, suffering from the curse of dimensionality. When D became larger, the curse of dimensionality was alleviated, but inter-modality correlation increased.[5] From our 58-dimensional feature data, the optimal modality D is 2 or 3, which enjoys the highest class-prediction accuracy. Table 6.3 shows the optimal D for different concepts (the second column).

Next, we compared different fusion models. Table 6.3 compares the class-prediction accuracy of product combination (PC), linear combination (LC), and super-kernel fusion (SKF). D indicates the number of independent modalities that the 58 independent components have been divided into. We found that super-kernel fusion performed on average 6.5% better than product-combination models and 4.5% better than linear-combination models. Note that the worst results were achieved when using the product rule, 2.0% worse than linear-combination models and 6.5% worse than those of super-kernel fusion. The reason is that if any of the classifiers reports the correct class *a posterior probability* as zero, the output will be zero, and the correct class cannot be identified. Therefore, the final result reported by the combiner in such cases is either a wrong class (worst case) or a reject (when all of the classes are assigned zero *a posterior probability*).

[5] The inter-modality correlation for all the D modalities is the summation of inter-modality correlations between every pair of modalities, which is $\sum_{i=1}^{D}{}_{j>i} C(M_i, M_j)$.

Table 6.4 AP (%) of video concept detection

Concept	IBM	PC	LC	SKF
Airplane	**24.93**	10.60	23.52	24.31
Animal	6.09	6.75	**8.59**	8.2
Building	8.02	7.92	4.68	**8.42**
Female Speech	67.23	49.10	67.23	**67.33**
Madeleine Albright	**47.41**	16.54	33.93	43.27
Nature Vegetation	37.84	31.02	33.65	**39.39**
News Subject Face	**8.12**	1.37	7.89	7.05
News Subject Mono.	**20.41**	3.1	8.87	13.48
NIST Non-Studio	69.1	69.65	66.38	**69.88**
Outdoors	65.16	**69.81**	53.87	66.16
People	11.82	12.95	16.41	**18.91**
Physical Violence	**3.04**	1.06	1.42	1.8
Road	10	7.72	**12.42**	8.38
Sport Event	48.45	24.20	40.49	**52.8**
Vehicle	**20.81**	14.05	15.63	16.54
Weather News	53.64	29.73	53.64	**86.7**
Average	31.38	22.28	28.04	**33.29**

Finally, we conducted fusion experiments on the video dataset. For this TREC video dataset, we got only probability outputs from single-modality classifiers through IBM. Therefore, we evaluated only fusion schemas on this video dataset. Table 6.4 compares the best results from IBM (IBM), product combination (PC), linear combination (LC), and super-kernel fusion (SKF) based on Average Precision of video concept detection. The numbers of modalities for sixteen concepts ranged from 2 to 6. Here we chose the NIST Average Precision (the sum of the precision at each relevant hit in the hitlist divided by the total number of relevant documents in the collection) as the evaluation criteria. Average Precision (AP) was used by NIST to evaluate retrieval systems in TREC-2003 video track competition. For TREC-2003 video track, a maximum of 1,000 entries This number was chosen in the IBM's work [4] for evaluation. were returned and ranked according to the highest probability of detecting the presence of the concept. The ground-truth of the presence of each concept was assumed to be binary (either present or absent in the data). For the 16 concepts in TREC-2003 video benchmark, super-kernel fusion performed around 5.2% better than the linear-combination models on average, 11.3% better than product-combination models. Super-kernel fusion also performed around 2.0% better than the best results provided by IBM.

6.5.3 *Observations*

After our extensive empirical studies on the two datasets, we can answer the questions proposed at the beginning of this section.

1. To deal with high-dimensional features from multiple media sources, it is necessary to do statistical analysis to reduce noise and find the most representative feature-components. Independent modality analysis can improve the effectiveness of multimedia data analysis by achieving a tradeoff between dimensionality curse and modality independency.
2. Super-kernel fusion is superior in its performance because its high model complexity can explore inter-dependencies between modalities.

6.6 Concluding Remarks

In this chapter, we have presented a framework of optimal multimodal information fusion for multimedia data analysis. First, we constructed *statistically independent modalities* from the given feature set from multiple media sources. Next, we proposed *super-kernel fusion* to learn the optimal combination of multimodal information. We carefully analyzed the tradeoffs between three design factors that affect fusion performance: *modality independence*, *curse of dimensionality*, and *fusion-model complexity*. Empirical studies show that our methods achieved markedly improved performance on a 2K image dataset and TREC-Video 2003 benchmarks.

This chapter shows a discriminative approach for fusing metadata of multiple modalities. In Chap. 8, we present a generative approach for conducting multimodal fusion. A discriminative approach tends to work more effectively, but it is difficult to interpret its results. On the contrary, a generative approach [41, 42] may have to rely on an assumed statistical model, but one can explain the yielded relationship between features and semantics.

References

1. Y. Wu, E.Y. Chang, K.C.C. Chang, J.R. Smith, Optimal multimodal fusion for multimedia data analysis, in *Proceedings of ACM Multimedia*, 2004, pp. 572–579
2. Y. Rui, T.S. Huang, S.F. Chang, Image retrieval: past, present, and future. J. Vis. Commun. Image Representation **10**, 1–23 (1997)
3. R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences, and trends of the new age. ACM Comput. Surv. **40**, 1–60 (2008)
4. B. Adams, A. Amir, C. Dorai, S. Ghosal, G. Iyengar, A. Jaimes, C. Lang, C.Y. Lin, A. Natsev, M. Naphade, C. Neti, H.J. Nock, H.H. Permutery, R. Singhx, J.R. Smith, S. Srinivasany, B.L. Tseng, T.V. Ashwin, D.Q. Zhang, IBM Research TREC-2002 video retrieval system, 2002
5. F. Stegmaier, Interoperable and unified multimedia retrieval in distributed and heterogeneous environments, in *Proceedings of ACM International Conference on Multimedia*, 2010, pp. 1705–1706
6. X. Anguera, J. Xu, N. Oliver, Multimodal photo annotation and retrieval on a mobile phone, in *Proceeding of ACM International Conference on Multimedia Information Retrieval*, 2008, pp. 188–194

7. K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is x "earest neighbor" meaningful? in *Proceedings of ICDT*, 1999, pp. 217–235
8. R. Bellman, *Adaptive Control Processes* (Princeton University Press, Princeton, 1961)
9. K. Goh, E. Chang, K.T. Cheng, Svm binary classifier ensembles for multi-class image classification, in *Proceedings of ACM International Conference on Information and Knowledgement Management* (*CIKM*), 2001, pp. 395–402
10. M.L. Cascia, S. Sethi, S. Sclaroff, Combining textual and visual cues for content-based image retrieval on the world wide web, in *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, 1998, pp. 24–28
11. L. Hansen, J. Larsen, T. Kolenda, On independent component analysis for multimedia signals. *Multimedia Image and Video Processing* (CRC Press, Boca Raton, 2000)
12. T. Kolenda, L.K. Hansen, J. Larsen, O. Winther, Independent component analysis for understanding multimedia content, in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, 2002, pp. 757–766
13. A. Vinokourov, D.R. Hardoon, J. Shawe-Taylor, Learning the semantics of multimedia content with application to web image retrieval and classification, in *Proceedings of Fourth International Symposium on Independent Component Analysis and Blind Source Separation*, 2003
14. J. Hershey, J. Movellan, Using audio–visual synchrony to locate sounds, in *Proceedings of NIPS*, 2001, pp. 813–819
15. J.W. Fisher III, T. Darrell, W. Freeman, P. Viola, Learning joint statistical models for audio–visual fusion and segregation, in *Proceedings of NIPS*, 2000, pp. 772–778
16. P. Smaragdis, M. Casey, Audio/visual independent components, in *International Symposium on Independent Component Analysis and Blind Source Separation*, 2003, pp. 709–714
17. A. Vinokourov, J. Shawe-Taylor, N. Cristianini, Inferring a semantic representation of text via cross-language correlation analysis, in *Proceedings of NIPS*, 2002, pp. 1473–1480
18. D.L. Donoho, High-dimensional data analysis: the curses and blessings of dimensionality. American Mathematical Society Lecture-Match Challenges of the 21st Century, 2000
19. R. Fagin, A. Lotem, M. Naor, Optimal aggregation algorithms for middleware, in *Proceedings of ACM PODS*, 2001
20. M. Flickner, H. Sawhney, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the qbic system. IEEE Comput. **28**(9), 23–32 (1995)
21. J.R. Smith, S.F. Chang, Visualseek: a fully automated content-based image query system, in *Proceedings of ACM Multimedia*, 1996, pp. 87–98
22. Y. Rui, T.S. Huang, S. Mehrotra, Content-based image retrieval with relevance feedback in mars, in *Proceedings of IEEE International Conference on Image Processing*, 1997, pp. 815–818
23. A. Velivelli, C.W. Ngo, T.S. Huang, Detection of documentary scene changes by audio–visual fusion, in *Proceedings of CIVR*, 2003, pp. 227–237
24. A. Hyvarinen, E. Oja, A fast fixed-point algorithm for independent component analysis, in *Proceedings of NIPS*, 1997, pp. 1483–1492
25. S. Amari, A. Cichocki, H.H. Yang, A new learning algorithm for blind signal separation, in *Proceedings of NIPS*, 1996, pp. 757–763
26. A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, in *Proceedings of NIPS*, 1995, pp. 1129–1159
27. F.R. Bach, M.I. Jordan, Kernel independent component analysis. J. Mach. Learn. Res. **3**, 1–48 (2002)
28. J. Kittler, M. Hatef, R.P.W. Duin, Combining classifiers, in *Proceedings of the International Pattern Recognition*, 1996, pp. 897–901
29. D.M. Tax, van M. Breukelen, R.P. Duin, J. Kittler, Combing multiple classifiers by averaging or by multiplying. Pattern Recognition **33**(9), 1475–1485 (2000)
30. K.M. Ting, I.H. Witten, Issues in stacked generalization. J. Artif. Intell. Res. **10**, 271–289 (1999)

31. R. Yan, A.G. Hauptmann, The combination limit in multimedia retrieval, in *Proceedings of ACM Multimedia*, 2003, pp. 339–342
32. I. Joliffe, Principal Component Analysis. (Springer, New York, 1986)
33. A.S. Lukic, M.N. Wernick, L.K. Hansen, S.C. Strother, An ICA algorithm for analyzing multiple data sets, in *Proceedings of IEEE International Conference on Image Processing*, 2002, pp. 821–824
34. M.S. Bartlett, H.M. Lades, T.J. Sejnowski, Independent component representation for face recognition, in *Proceedings of the SPIE Conference on Human Vision and Electronic Imaging III*, 1998, pp. 528–539
35. C.H.Q. Ding, X. He, H. Zha, M. Gu, H.D. Simon, A min–max cut algorithm for graph partitioning and data clustering, in *Proceedings of IEEE ICDM*, 2001, pp. 107–114
36. J. Platt, Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, *Advances in Large Margin Classifiers* (MIT Press, Cambridge, 2000), pp. 61–74
37. C.J.C. Burges, A tutorial on support vector machines for pattern recognition, in *Proceedings of ACM KDD*, 1998, pp. 121–167
38. V. Roth, V. Steinhage, Nonlinear discriminant analysis using kernel functions, in *Proceedings of NIPS*, 1999, pp. 568–574
39. T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. **2**, 263–286 (1995)
40. B. Li, E. Chang, Discovery of a perceptual distance function for measuring image similarity. ACM Multimedia J. (Special Issue on Content-Based Image Retrieval) **8**(6), 512–522 (2003)
41. Y. Wu, E.Y. Chang, B.L. Tseng, Multimodal metadata fusion using causal strength, in *Proceedings of ACM Multimedia*, 2005, pp. 872–881
42. W. Chen, D. Zhang, E.Y. Chang, Combinational collaborative filtering for personalized community recommendation, in *Proceedings of ACM KDD*, 2008, pp. 115–123