An Information-Theoretic Framework for Semantic-Multimedia Retrieval

JOÃO MAGALHÃES
Universidade Nova de Lisboa
and
STEFAN RÜGER
Knowledge Media Institute, The Open University

This article is set in the context of searching text and image repositories by keyword. We develop a unified probabilistic framework for text, image, and combined text and image retrieval that is based on the detection of keywords (concepts) using automated image annotation technology. Our framework is deeply rooted in information theory and lends itself to use with other media types.

We estimate a statistical model in a multimodal feature space for each possible query keyword. The key element of our framework is to identify feature space transformations that make them comparable in complexity and density. We select the optimal multimodal feature space with a minimum description length criterion from a set of candidate feature spaces that are computed with the average-mutual-information criterion for the text part and hierarchical expectation maximization for the visual part of the data. We evaluate our approach in three retrieval experiments (only text retrieval, only image retrieval, and text combined with image retrieval), verify the framework's low computational complexity, and compare with existing state-of-the-art ad-hoc models.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*

General Terms: Algorithms, Measurement, Experimentation

Additional Key Words and Phrases: Indexing, search, multimedia, automated keyword annotation

ACM Reference Format:

Magalhães J. and Rüger S. 2010. An information-theoretic framework for semantic-multimedia retrieval. ACM Trans. Inform. Syst. 28, 4, Article 19 (November 2010), 32 pages.
DOI = 10.1145/1852102.1852105 http://doi.acm.org/10.1145/1852102.1852105

1. INTRODUCTION

Users are nowadays widely familiar with efficient text-based searches on the World Wide Web. However, when one wishes to search multimedia collections

Authors' addresses: J. Magalhães, Department of Computer Science, Faculdade de Ciencias e Technolgias, Universidad Nova de Lisboa, Lisbon, Portugal; email: jmag@di.fct.unl.pt; S. Rüger, Knowledge Media Institute, The Open University, Milton Keynes, UK; email: s.rueger@open.ac.uk.

DOI 10.1145/1852102.1852105 http://doi.acm.org/10.1145/1852102.1852105

through a text query, the missing relation between multimedia low-level features and human knowledge becomes a serious bottleneck. On one hand, the system must mimic human perception and extract the relevant semantics from multimedia data. On the other hand, the system must be able to interpret the human request and rank documents according to their relevance.

This becomes an increasing problem as search engines collect large amounts of visual and audio data. While in text retrieval we express our query in the form of the document (text), in multimedia systems this is more difficult. The user is not aware of the low-level representation of multimedia, e.g., color, texture, shape features, pitch, rhythms, or tones. These low-level feature spaces are ideal to find multimedia documents with similar colors, textures, shapes, pitch, rhythm, and so on, but are not adequate to find multimedia by semantic content, for example, all pictures with flowers or all high-pitched singers. Thus, the extension of existing search engines to support multimedia information becomes a critical aspect.

Nowadays, conventional search engines that make use of semantics on the information side depend on manual annotations and other information extracted from the surrounding content, for example, HTML links in case of Web content. This way of extracting multimedia semantics is flawed and costly. Our daily routines are intrinsically attached to systems that allow us to search for specific news articles, scientific papers, photos, music, videos, or information in general. Demand for techniques that handle multimedia documents is increasing with the wide spread use of multimedia-dedicated applications.[1] All these applications use manual annotations done by users and metadata provided by content owners to enable multimedia search. Thus, doing the entire process automatically, or even semiautomatically, can greatly decrease the operational and maintenance costs of such applications.

The automated method that we shall propose explores low-level features of multimedia to infer the presence of concepts to enable search-by-keyword. We group low-level features into sparse feature spaces and dense feature spaces. Audio, text, and visual features fall into one of these two categories. However, in this article we only consider text and visual data. We employ two methods to deal with sparse feature spaces (the average mutual information) and dense feature spaces (hierarchical-EM). Keyword models are estimated as maximum-entropy models in the representation selected by the minimum description length criterion. The following section will formalize the problem addressed in this article.

1.1 Problem Definition

To infer the presence of concepts in multimedia documents, a new breed of information retrieval model is required, one that seamlessly integrates heterogeneous data. Thus, in this article we assume that in any given collection, $\mathcal{D}$, of N multimedia documents:

$$\mathcal{D} = \{d^1, d^2, \ldots, d^N\}, \tag{1}$$

[1] www.getty.com; www.corbis.com; www.flickr.com; and www.last.fm.com.

each document is characterized by a vector:

$$d^j = \left(d_T^j, d_V^j, d_W^j\right), \tag{2}$$

composed by a feature vector, d_T, describing the text-part of the document, a feature vector, d_V, describing the visual part of the document, and a keyword vector, d_W, describing the semantics of the document. More specifically, we have the following.

—The feature vector d_T contains text-based features such as text terms obtained via a stemmer, bag-of-word, part-of-speech or named entities.

—The feature vector d_V contains low-level visual features such as texture, color, or shape.

—The feature vector d_W contains keyword confidence scores concerning the presence of the corresponding concept in that document.

Algorithms and techniques to compute low-level text and visual features are widely studied. There is no single best way to use keyword features representing multimedia information because of the ambiguity and subjectivity of the information that they try to describe—the semantic content of a multimedia document. The semantic description of multimedia information, the feature vector d_W, is the core topic of this article. To describe the semantics of multimedia information we define the set:

$$\mathcal{W} = \{w_1, \ldots, w_L\} \tag{3}$$

as a vocabulary of L keywords. These keywords are linguistic representations of abstract or concrete concepts that we want to detect in multimedia documents. The feature vector d_W is formally defined as:

$$d_W^j = \left(d_{W,1}^j, d_{W,2}^j, \ldots, d_{W,L}^j\right), \tag{4}$$

where each component $d_{W,t}^j$ is a score indicating the confidence that keyword w_t is present in that particular document. The concepts may not be explicitly present in the multimedia information; methods are required to compute the likelihood that the keyword is actually present in the multimedia document.

Equation (2) shows us the other information that we have about documents: text and visual features. Thus, to compute the components of the keyword vector d_W^j we shall use text and visual feature data. This leads us to the definition of each component of the keyword vector as:

$$d_{W,t}^j = p\left(y_t^j = 1 | d_T^j, d_V^j, \beta_t\right), \tag{5}$$

where the random variable $y_t^j = \{1, 0\}$ indicates the presence/not-presence of keyword w_t on document d^j, given its text feature vector d_T^j, its visual feature vector d_V^j and the keyword model β_t. Equation (2) integrates heterogeneous representations of a multimedia document (text, image, and semantic) and Equation (5) will make multimedia information searchable with the same type of queries for all types of media.

1.2 Organization

In Magalhães and Rüger [2007] we introduced an information-theoretic framework for Equation (5). The current article proposes and presents a definitive and thorough account of our framework [Magalhães 2008]. In Section 3 we propose a statistical framework that can simultaneously model text-only documents, image-only documents, and documents with both text and images. Section 4 details the estimation of an optimal feature space to represent multimedia information and Section 5 details the estimation of keyword models in that feature space. Section 6 presents a thorough evaluation of the framework. Next, we shall discuss related work.

2. RELATED WORK

In text retrieval, the search process is triggered by a text query that can be directly matched to the corpus of the documents in the collection. Since we want to offer a common query interface for both text and images we need to define a common vocabulary of keywords to query all possible types of documents. Therefore the present work is related to text topic detection, image annotation, and multimodal content annotation. We will now look at these three areas with a view to seamlessly integrating text and image data into the same framework.

Text topic detection models preprocess data by removing stop-words and rare words, stemming, and finally term-weighting. Due to the high-dimensional feature space of text data, most text categorization algorithms are linear models such as naïve Bayes [McCallum and Nigam 1998], maximum entropy [Nigam et al. 1999], Support Vector Machines [Joachims 1998], regularized linear models [Zhang and Oles 2001], and Linear Least Squares Fit [Yang and Chute 1994]. Joachims [1998] applies SVMs directly to the text terms. Text is ideal for applying SVMs without the need of a kernel function because data is already sparse and high-dimensional. Linear models fitted by least squares such as the one by Yang and Chute [1994] offer good precision, and in particular regularized linear methods, such as the one we propose and the one by Zhang and Oles [2001], perform similarly to SVMs, with the advantage of yielding a probability density model. The maximum entropy classification model proposed by Nigam et al. [1999] defines a set of features that are dependent on the class being evaluated while we use a unique set of features for all keywords. The proposed maximum entropy framework has the same characteristics and performance as linear models (logistic regression, least squares) but with the crucial advantage that while these approaches have no automatic mechanism to select a vocabulary size, we use the minimum description length principle to select its optimal size. Yang [1999] and Yang and Liu [1999] have compared a number of topic detection algorithms and reported their performance on different text collections. Their results indicate that k-Nearest Neighbor, SVMs, and LLSF are the best classifiers. Note that nearest neighbor approaches have certain characteristics (see Hastie et al. [2001]) that make them computationally too complex to handle large-scale indexing.

The simplest image annotation models deploy a traditional multiclass supervised learning model and learn the class-conditional probability density distribution of each keyword, w, given its training data, x. Bayes' law is used to model $p(x \mid w)$, the feature data density distribution of a given keyword. Several techniques to model $p(x \mid w)$ with different types of probability density distributions have been proposed: Yavlinsky et al. [2005] deployed a nonparametric distribution; Carneiro and Vasconcelos [2005] a semiparametric density estimation; Westerveld and de Vries [2003] a finite-mixture of Gaussians; while Vailaya et al. [2001] apply a vector quantization technique. Density-based approaches are among the most successful. However, density distributions are not adequate for text because the density models do not get enough support from such sparse data. Other types of approaches are based on a translation model between keywords and images (global, tiles, or regions). Inspired by automatic text translation research, Duygulu et al. [2002] developed a method of annotating images with words. First, regions are created using a segmentation algorithm like normalized cuts. For each region, features are computed and then blobs are generated by clustering the image features for these regions across an image collection. The problem is then formulated as learning the correspondence between the discrete vocabulary of blobs and the image keywords. Following the same translation approach Jeon et al. [2003]; Lavrenko et al. [2003]; and Feng et al. [2004] have developed a series of translation models that use different models for keywords (multinomial/binomial) and image representations (hard clustered regions, soft clustered regions, tiles). Hierarchical models have also been used in image annotation such as Barnard and Forsyth's [2001] generative hierarchical aspect model inspired by a hierarchical clustering/aspect model. The data are assumed to be generated by a fixed hierarchy of nodes with the leaves of the hierarchy corresponding to soft clusters. Blei and Jordan [2003] propose the correspondence latent Dirichlet allocation model; a Bayesian model for capturing the relations among regions, words, and latent variables. The exploitation of hierarchical structures of either the data or of the parameters increases the number of parameters (model complexity) to be estimated with the same amount of training data.

Maximum entropy models have also been applied to image annotation [Jeon and Manmatha 2004; Argillander et al. 2005] and object recognition [Lazebnik et al. 2005]. These three approaches have specific features for each class (keywords in our case), which increase the complexity of the system. It is curious to note the large difference in precision results between Jeon and Manmatha [2004] and Argillander et al. [2005]; we believe that it is related to the lack of regularization and to a differing number of features. These approaches were not as successful as density-estimation-based models since maximum entropy works best in high-dimensional sparse feature spaces. The proposed maximum entropy framework tackles this problem by expanding the feature space in a spirit similar to Hoffman's probabilistic Latent Semantic Indexing [Hofmann 1999].

These single-modality-based approaches are far from our initial goal, but by analyzing them we can see which family of models can be used to simultaneously model text, image, and multimodal content. Each modality captures different aspects of that same reality, thus carrying valuable information about each keyword of the vocabulary. The simplest approach to multimodal analysis is to design one classifier per modality and combine the output of these classifiers. Westerveld et al. [2003] combine the visual model and the text model under the assumption that they are independent, thus the probabilities are simply multiplied. Naphade and Huang [2001] model visual features with Gaussian Mixtures Models (GMM), audio features with Hidden Markov Models (HMM) and combine them in a Bayesian network. In multimedia documents the different modalities contain co-occurring patterns that are synchronized/related in a given way because they represent the same reality. Synchronization/relation and the strategy to combine the multimodal patterns is a key point of the Semantic Pathfinder System proposed by [Snoek et al. 2006b; Snoek et al. 2006a]. Their system uses a feature vector that concatenates a rich set of visual features, text features from different sources (ASR, OCR), and audio features. Three types of classifiers are available: logistic regression (which without regularization is known to overfit [Chen and Rosenfeld 1999]), Fisher linear discriminant, and SVMs (offering the best accuracy). The fusion of the different modalities is possible to be done at different levels and it is chosen by cross-validation for each keyword. The extremely high computational complexity required to compute the visual features and to iteratively select the best classifier, the best type of fusion, and the SVMs parameter optimization are serious drawbacks of this system. IBM's Marvel system [Amir et al. 2005] has a similar architecture with different learning algorithms to analyze the semantics of multimedia content. These two approaches offer the best performance on the TRECVID2005 conference. Both approaches combine the high-dimensional sparse text features and the low-dimensional dense features on the same feature vector. This might represent a problem for the optimization procedure because the information present in each dimension can be very different. Ideally each dimension should contain the same amount of information and the data density/sparseness should be similar across the entire feature space. The first step of our framework aims to find this optimal trade-off point by compressing the text feature space dimension and by expanding the visual feature space dimension.

3. AN INFORMATION-THEORETIC FRAMEWORK

Our first objective is to compute the components of keyword feature vectors d_W representing the semantics of multimedia documents. For this, we will estimate and select a model, from a set, Θ, of candidate models that best represents the keyword w_t in terms of text data and visual data. The statistical model, $\beta_t \in \Theta$, of Equation (5) can assume many forms (e.g., nearest neighbor, neural networks, linear models, support vector machines) according to the family of algorithms and also to the complexity of the specific algorithm within a particular family of algorithms. The choice of the family of algorithms is done by examining the requirements that multimedia information retrieval applications face in a real-world scenario:

(1) arbitrary addition and removal of keywords to/from the query vocabulary,

(2) easy update of existing keyword models with new training data,

(3) seamless integration of heterogeneous types of data,

(4) computationally efficient indexing of multimedia information, and
(5) good retrieval effectiveness.

The first two requirements concern an important practical aspect of large-scale multimedia indexes—the integrity of the index when keyword models are modified. When a keyword model is modified (added, removed, or updated) the index can be affected in two ways: if keyword models are dependent then the entire index becomes obsolete; if keyword models are independent then only the part of the index concerning that keyword becomes obsolete. This leads to a solution where keyword models are independent so that a modification in one keyword model will have a minor influence on the indexes. Thus, presence of keywords shall be represented by Bernoulli random variables y_t.

The remaining three requirements can be difficult to accommodate in a unifying model: supporting multimodal information, ability to quickly index new multimedia content, and achieving good accuracy. When modelling multimodal keywords, one has to deal with both dense feature spaces and sparse feature spaces. On one hand visual feature data can be very dense, making its modelling difficult due to the irregular frontiers caused by keyword cross-interference. Expanding the original feature space into higher-dimensional ones results in a sparser feature space where the classes' separation can be made easier. On the other hand, text feature spaces are typically too sparse, making their modelling difficult because there is not enough support data to estimate the details of keyword models. In these situations we have to compress the feature space into a lower dimensional space where data is compressed into a denser space. These transformations of the original feature space into a space where the data is optimally distributed is represented as:

$$\mathrm{F}\left(d_T^j, d_V^j\right) = \left(\mathrm{F}_\mathrm{T}\left(d_T^j\right), \quad \mathrm{F}_\mathrm{V}\left(d_T^j\right)\right), \tag{6}$$

where $\mathrm{F}_\mathrm{T}(d_T^j)$ corresponds to the text data transformation and $\mathrm{F}_\mathrm{V}(d_V^j)$ corresponds to the visual data transformation. This renders the final expression for the components of keyword feature vectors as:

$$d_{\mathrm{W},t}^j = p\left(y_t^j = 1 \mid \mathrm{F}\left(d_T^j, d_V^j\right), \beta_t\right). \tag{7}$$

The transformation of multimedia document features only needs to be computed once for all keyword models. In other words the transformation is independent of the keyword models. The interesting implication of this fact is that it can reduce the indexing computational complexity. Because the transformation generates a high-dimensional space, one can limit the keyword model search space, Θ, to the family of linear models, which have a very low computational complexity in the classification phase (but not necessarily in the learning phase). Besides the low computational complexity, linear models offer other interesting advantages: support of high-dimensional data (easy integration of heterogeneous data), naturally embedded background knowledge in the form of priors (ideal for keyword model update), and good accuracy (retrieval effectiveness).

4. OPTIMAL DATA REPRESENTATION

The transformations $\mathrm{F_T}(d_T)$ and $\mathrm{F_V}(d_V)$ change the representation of the original text and visual feature spaces. As mentioned, transformations $\mathrm{F_T}(d_T)$ and $\mathrm{F_V}(d_V)$ will adopt specific strategies adequate to the characteristics of each type of data. However, in both cases there is the problem of selecting the optimal transformation from the large number of possible transformations and their varying complexities. In practice, the selection of the optimal transformation is equivalent to old questions like “how many text features?” and “how many visual clusters?” that are usually addressed by some heuristic method. In this section we shall formally address this problem. The proposed feature space transformations are inspired by information theory. The space transformation, F, can be seen as a codebook composed by a set of $M = M_T + M_V$ codewords representing the data space. Given the codebook of a feature space, one is able to represent all samples of that feature space as a linear combination of keywords from that codebook. Information theory [Cover and Thomas 1991] provides us with a set of information measures that not only assess the amount of information that one single source of data contains, but also the amount of information that two or more sources of data have in common. Thus, we employ the minimum description length criterion [Rissanen 1978], to infer the optimal complexity, M_T and M_V, of each feature space transformation, $\mathrm{F_T}(d_T)$ and $\mathrm{F_V}(d_V)$. Note that we use the word “optimal” from an information theory point of view. The treatment of the model selection problem presented in this section is based on Hastie et al. [2001] and MacKay [2004].

4.1 Assessing the Data Representation Error

The process of changing the original feature space representation into the new representation with a given candidate transformation, $\hat{\mathrm{F}}$, has an associated error. If we represent $\hat{\mathrm{F}}$ as the estimated transformation, and G as the lossless transformation that we are trying to estimate, we can compute the mean-squared deviation between the estimated model and the desired response as the error:

$$\begin{aligned}\mathrm{Err}_{\mathcal{D}}(d) &= \mathrm{E}[(G(d) - \hat{\mathrm{F}}(d))^2] \\ &= \sigma_e^2 + (\mathrm{E}[\hat{\mathrm{F}}(d)] - G(d))^2 + E[\hat{\mathrm{F}}(d) - \mathrm{E}[\hat{\mathrm{F}}(d)]]^2. \qquad (8)\end{aligned}$$

The first term is the variance of the modelled process and cannot be avoided. The second term measures the difference between the true mean of the process and the estimated mean. The third term is the variance of the estimated model around its mean. This expression can be written as:

$$\mathrm{Err}_{\mathcal{D}}(d) = \sigma_e^2 + \mathrm{Bias}^2(\hat{\mathrm{F}}(d)) + \mathrm{Variance}(\hat{\mathrm{F}}(d)). \qquad (9)$$

The more complex we make candidate transformation $\hat{\mathrm{F}}$ the lower the bias but higher the variance. Equation (9) expresses the transformation bias-variance tradeoff: simple transformations can only represent the training data’s coarse details (high bias) causing a high prediction error (high variance) because the transformation ignores important aspects of the data structure; complex transformations can represent training data structures in great detail (low bias) but the prediction error increases (high variance) because the transformations do not generalize to other data. The optimal transformation is the one that achieves the best generalization error on the new unseen samples. There are two types of methods to select the transformation that has the best generalization error: empirical methods use validation data different from the training data to assess the model generalization error on the test data, for example, cross-validation and bootstrap; criteria-based methods provide an estimate of the model generalization error in the test data based on the error in the training data and the complexity of the model, for example, Bayesian Information Criterion. The minimum description length criterion is in the later group, and we chose it as the model selection criterion for the feature space transformation.

4.2 The MDL Principle

Model selection is a widely studied subject, see [Hastie et al. 2001], and the minimum description length (MDL) criterion is among the most common criteria of model selection. Rooted in information theory, the MDL principle was initially thought of as a method to find the minimum number of bits required to transmit a particular message, *msg*. To transmit this message a codebook, *cbk*, such as Huffman coding can be used to compress the message. Thus, the total number of bits required to transmit the message is:

$$\mathrm{DL}(msg,\, cbk) = \mathrm{DL}(msg \mid cbk) + \mathrm{DL}(cbk), \tag{10}$$

corresponding to the description length of message *msg* encoded with codebook *cbk* plus the description length of codebook *cbk*. The MDL principle says that the optimal trade-off between these two quantities is achieved with codebook $cbk_{\min}$, which minimizes the expression. The minimum description length is written as:

$$\mathrm{MDL}(msg) = \mathrm{DL}(msg \mid cbk_{\min}) + DL(cbk_{\min}), \tag{11}$$

where $cbk_{\min}$ is the optimal codebook, which allows message *msg* to be transmitted with the minimum number of bits.

The relation between the MDL criterion and the problem of model selection is straightforward: it assesses the trade-off between the data likelihood (the message) under a given model (the codebook) and the complexity of that model. In the problem we are addressing, data $\mathcal{D}$ will be transformed into a new feature space by a transformation $\hat{\mathrm{F}}$. Hence, Equation (10) is written as the sum of the likelihood of data $\mathcal{D}$ on the new feature space and the complexity of feature space transformation $\hat{\mathrm{F}}$. Formally, we have:

$$\mathrm{DL}(\hat{\mathrm{F}}_i, \mathcal{D}) = -\sum_{d\in\theta} \log p(d \mid \hat{\mathrm{F}}_i) + \frac{npars}{2} \cdot \log N, \tag{12}$$

where *npars* is the number of parameters of the transformation $\hat{\mathrm{F}}$, and N is the number of samples in the training dataset. Hence, the MDL criterion is designed "to achieve the best compromise between likelihood and . . . complexity relative to the sample size" [Barron and Cover 1991]. Finally, the optimal feature space transformation is the one that minimizes Equation (12), which results in:

$$\mathrm{F} = \arg\min_{\hat{\mathrm{F}}} \mathrm{DL}\,(\hat{\mathrm{F}}, \mathcal{D}). \tag{13}$$

The MDL criterion provides an estimate of the model error on the test data. Note that it is not an absolute estimate—it is only relative among candidate models. The minimum description length approach is formally identical to the Bayesian Information Criterion but is motivated from an information-theoretic perspective, see MacKay [2004].

4.3 Independent Features Processing

When modelling multimodal keywords, one has to deal with heterogeneous feature spaces. Different features capture different aspects of data in a compressed way, for example, the dominant color of an image, the pitch of a two-second audio segment, or a bag of words of a one hundred-page document. The resulting feature spaces can be either low-dimensional and dense or high-dimensional and sparse. Note that if the computed feature space is high-dimensional and dense then one can probably use the original data itself or a scaled down version of it.

Specific feature spaces require specific processing methods, for example, imagine a single dimension of a color moments feature space and a single dimension of bag-of-words feature space. Their distributions across a given dataset are quite different, posing disparate challenges and demanding different transformations. For this reason, we processed feature spaces independently.

Since we are processing features independently and combining them, one must assess the representation error of the data under each specific transformation. Thus, a common criterion to select the transformation for each specific feature space is a key aspect in the framework. The common criterion guarantees that the optimal value (the minimum) is selected for all feature spaces.

This approach has the advantage of creating a generic framework capable of handling heterogeneous sources of data. Finally, note that if we have a homogeneous set of features, we could apply the same kind of treatment to all feature spaces or even model the keywords directly in the original feature space. However, this is generally not the case with multimodal data.

4.4 Dense Space Transformations

Some of the input feature spaces (depending on media type) can be very dense, making its modelling difficult due to cross-interference between classes. Expanding the original feature space into higher-dimensional ones results in a sparser feature space, where the modelling of the data can be easier. This technique is applied by many related methods, such as kernels. The low-level visual features that we use are dense and low-dimensional, hence keyword data may overlap, thereby increasing the cross-interference. This means that not only the discrimination between keywords is difficult but also that the estimation of a density model is less effective due to keyword data overlapping. One solution is to expand the original feature space into a higher-dimensional feature space, where keyword data overlap is minimal. Thus, we define F_V as the transformation that increases the number of dimensions of a dense space with m dimensions into an optimal space with K_V dimensions:

$$\mathrm{F}_{\mathrm{V}}(d_{V,1},\ldots,d_{V,m}) = \begin{bmatrix} \mathrm{f}_{V,1},(d_{V,1}\ldots,d_{V,m}) \\ \vdots \\ \mathrm{f}_{V,k_V},(d_{V,1}\ldots,d_{V,m}) \end{bmatrix}^T, k_V \gg m. \tag{14}$$

In other words, for an input feature space with m dimensions, the transformation $\mathrm{F}_{\mathrm{V}}(d_{V,1},\ldots,d_{V,m})$ generates a k_V dimensional feature space with $k_V \gg m$, where each dimension, i, of the new feature space corresponds to the function $\mathrm{f}_{V,i},(d_{V,1}\ldots,d_{V,m})$. The optimal number of such functions, k_V, will be selected by the MDL principle. The method to estimate the functions is defined next.

4.4.1 *Visual Transformation by Vector Quantization.* The original visual feature vector $d_V = (d_{V,1},\ldots,d_{V,m})$ is composed of several low-level visual features with a total of m dimensions. These m dimensions span J visual feature types (e.g., marginal HSV colour moments, Gabor filters and Tamura), that is, the sum of the number of dimensions of each one of the J visual feature space equals m. This implies that each visual feature type j is transformed individually by the corresponding $\mathrm{F}_{\mathrm{V,j}}(d_{V,j})$ and the output is concatenated into the vector:

$$\mathrm{F}_{\mathrm{V}}(d_V) = (\mathrm{F}_{V,1}(d_{V,1}),\ldots,\mathrm{F}_{V,j}(d_{V,j})), \tag{15}$$

where the dimensionality of the final F_{V} transformation is the sum of the dimensionality of each individual visual feature space transformation $\mathrm{F}_{V,j}$:

$$k_V = k_{V,1} + \cdots + k_{V,j} + \cdots + k_{V,J}. \tag{16}$$

The form of visual feature space transformations $\mathrm{F}_{\mathrm{V},j}$ is based on Gaussian mixture density models. The components of a GMM capture the different modes of the problem's data. Each component shall correspond to a dimension of the optimal feature space where modes are split and well separated, thereby creating a feature space where keywords can be modelled with a simple and low cost algorithm. The transformations are defined under the assumption that subspaces are independent. This allows us to process each visual feature subspace, j, individually, and model it as a Gaussian mixture model (GMM):

$$p(d_V) = p(d_V \mid \theta_j) = \sum_{m=1}^{k_{V,j}} \alpha_{m,j}\, p\big(d_V \mid \mu_{m,j}, \sigma^2_{m,j}\big), \tag{17}$$

where d_V is the low-level feature vector, θ_j represents the set of parameters of the model of the j visual feature subspace: the number, $k_{V,j}$, of Gaussian components, the complete set of model parameters with means, $\mu_{m,j}$, covariances $\sigma^2_{m,j}$, and component priors $\alpha_{m,j}$. The component priors have the convexity constraint $\alpha_{1,j},\ldots,\alpha_{k_{V,j}} \geq 0$ and $\sum_{m=1}^{k_{V,j}} \alpha_{m,j} = 1$. Thus, for each visual feature space j, we have the Gaussian mixture model with $k_{V,j}$ components, which now

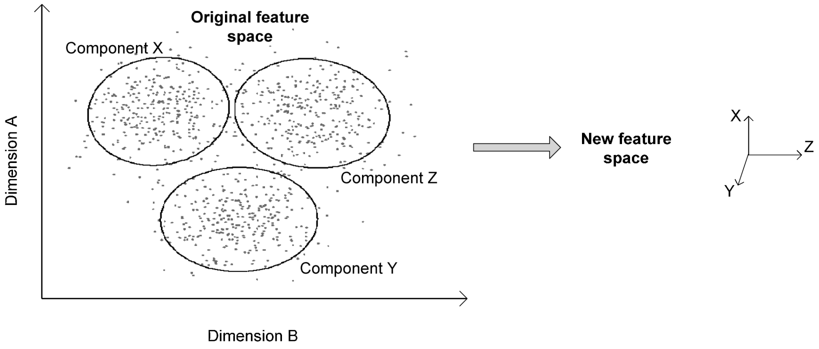

Fig. 1. Example of feature space expansion with a GMM.

defines the transformation:

$$\mathrm{F}_{V,j}(d_V) = \begin{bmatrix} \alpha_{1,j} p\big(d_V \mid \mu_{1,j}, \sigma^2_{1,j}\big) \\ \vdots \\ \alpha_{kV,j} p\big(d_V \mid \mu_{kV,j}, \sigma^2 k_V, j\big) \end{bmatrix}^2, \tag{18}$$

where each dimension corresponds to a component of the mixture model. Figure 1 illustrates two-dimensional samples of a dataset and a density model of those data points, more specifically, a GMM with three components. In this explanatory example, the original space has two dimensions while the new feature space shall have three dimensions. This way, when the amount of data points is large enough, we can compute a GMM with a large number of components.

The critical question arising from Equation (18) is that one does not know the optimal complexity of the GMM in advance. The complexity is equivalent to the number of parameters, which in our case is proportional to the number of mixture components $k_{V,j}$:

$$npars_j = k_{V,j} + \mathrm{dim}_j \cdot k_{V,j} + k_{V,j} \frac{\mathrm{dim}_j \cdot (\mathrm{dim}_j + 1)}{2}, \tag{19}$$

where dim_j, is the dimensionality of the visual subspace, j. Note the relation between Equation (18) and Equation (12). To address the problem of finding the ideal complexity, we implemented a hierarchical EM algorithm that starts with a large number of components and progressively creates different GMM models with a decreasing number of components.

4.4.2 *Hierarchical EM.* The hierarchical EM algorithm was implemented in C++ and is based on the one proposed by Figueiredo and Jain [2002]. It follows the component-wise EM algorithm with embedded component elimination. The mixture fitting algorithm presents a series of strategies that avoids some of the EM algorithm's drawbacks: sensitivity to initialization, possible convergence to the boundary of the parameter space and the estimation of different feature importance.

The algorithm starts with a number of components that is much larger than the real number and gradually eliminates the components that start to get few support data (singularities). This avoids the initialization problem of EM since the algorithm only produces mixtures with components that have enough support data. Component stability is checked by assessing its determinant (close to singularity) and its prior (few support data). If one of these two conditions is not met, we delete the component and continue with the remaining ones. This strategy can cause a problem when the initial number of components is too large: no component receives enough initial support, causing the deletion of all components. To avoid this situation, component parameters are updated sequentially and not simultaneously as in standard EM. That is, first update component 1 parameters (μ_1, σ_1^2, then recompute all posteriors, update component 2 parameters (μ_2, σ_2^2, recompute all posteriors, and so on. After finding a good fit for a GMM with k components, the algorithm deletes the weakest component and restarts itself with $k - 1$ Gaussians and repeats the process until a minimum number of components is reached. Each fitted GMM is stored and in the end the set of fitted models describe the feature subspace at different levels of granularities.

The hierarchical EM algorithm for Gaussian mixture models addresses the objective of finding the optimal feature space by, (1) creating transformations with different complexities, and (2) splitting data modes into different space dimensions, hence enabling the application of low-cost keyword modelling algorithms.

4.4.3 *Experiments.* Experiments assessed the behavior of the hierarchical EM algorithm on a real world photographic image collection. The collection is a 4500 image subset of the widely used Corel CDs Stock Photos [Duygulu et al. 2002]. The low-level visual features that we use in our implementation are the following.

—*Marginal HSV distribution moments*. This 12-dimensional color feature captures the 4 central moments of each color component distribution;

—*Gabor texture*. This 16-dimensional texture feature captures the frequency response (mean and variance) of a bank of filters at different scales and orientations; and

—*Tamura texture*. This 3-dimensional texture feature is composed the image's coarseness, contrast, and directionality.

The evolution of the model likelihood and complexity with a decreasing number of components are the two most important characteristics of the hierarchical EM that we wish to study. The algorithm is applied to individual visual feature subspaces. Each GMM model starts with $k_{V,j} = 200$ Gaussians, and the algorithm fits models with a decreasing number of components until a minimum of a single Gaussian. One of the assumptions of the minimum description length principle is that the number of samples is infinite. Thus, to increase the accuracy of the MDL criterion we created 3 by 3 tiles of the training images. This increased the number of training samples by a factor of 9, which greatly improves the quality of the produced GMMs because of the existence of more data to support the model parameters.

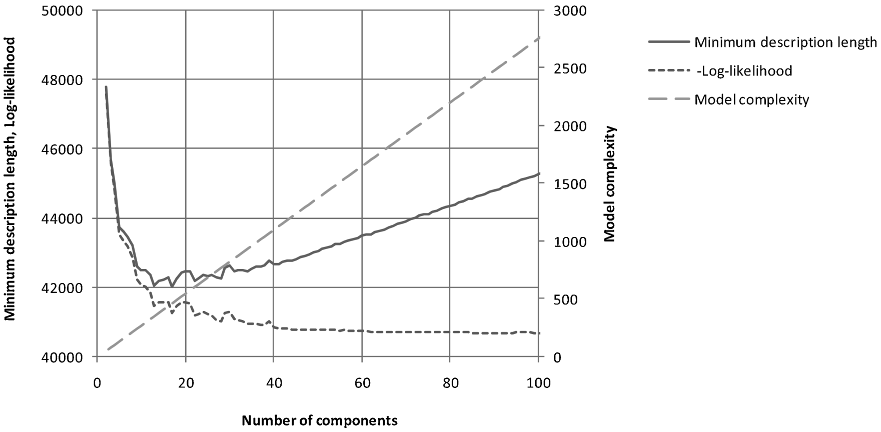

Fig. 2. Model selection for the Gabor filters features (Corel5000).

4.4.4 *Results and Discussion.* An advantage of the chosen algorithm to find the optimal transformation is its natural ability to generate a series of transformations with different levels of complexity. This allows assessing different GMMs with respect to the trade-off between decreasing levels of granularity and the likelihood of their fit to the data.

Figure 2 illustrates the output of a GMM model fitting to the output of one Gabor filter. The minimum description length curve (solid line) shows the trade-off between the model's complexity (dashed line) and the model's likelihood (dotted line). Note that we are actually plotting *–log-likelihood*for better visualization and comparison. The models likelihood curve is quite stable for models with a large number of components (above 40). On the other extreme of the curve one can see that for models with fewer than 40 components, the likelihood starts to exhibit poorer performance. The small glitches in the likelihood curve are the result of component deletion from a particularly good fit (more noticeable between 10 and 20 components). This effect is more visible when a component has been deleted from a model with a low number of components because the remaining ones are not enough to cover the data that was supporting the deleted one. The model complexity curve shows the penalty increasing linearly with the number of components according to Equation (19). The most important curve of this graph is the minimum description-length curve. At the beginning, it closely follows the likelihood curve because the complexity cost is low. As the model complexity increases, the model likelihood also becomes better but no longer at the same rate as initially (less than 10 components). This causes the model penalty to take a bigger part in the MDL formula, and after 20 components the MDL criterion indicates that those models are not better than previous ones. Thus, according to the MDL criterion the optimal transformation for this Gabor filter is the model with 18 components.

The selection of the transformation of the Tamura visual texture features is illustrated in Figure 3. The behavior is the same as for the Gabor features with the only difference being that the change from the descending part of the MDL curve to the ascending part is not so pronounced. This indicates that the

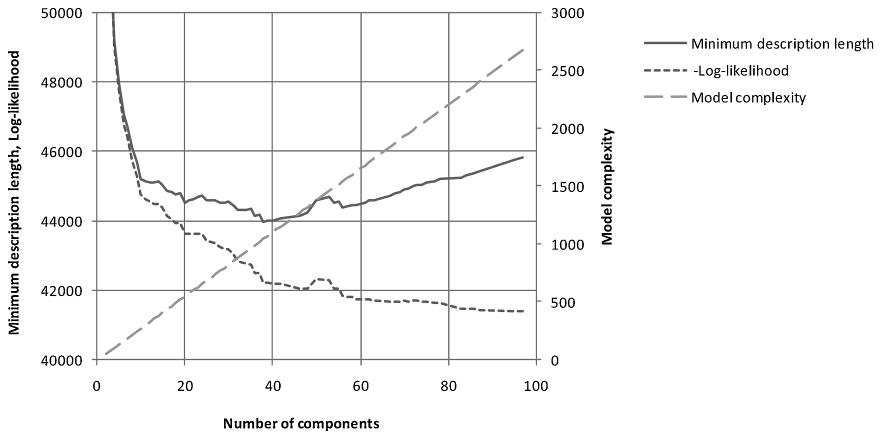

Fig. 3. Model selection for the Tamura features (Corel5000).

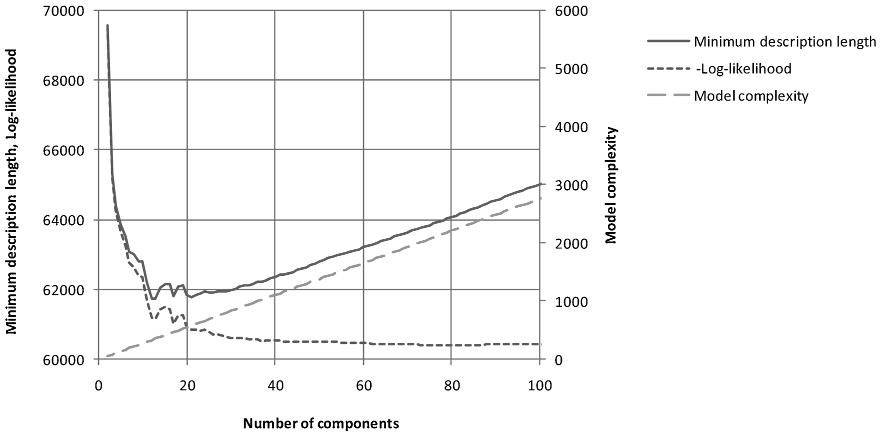

Fig. 4. Model selection for the marginal moments of HSV colour histogram features (Corel5000).

optimal model, $k_{V,j} = 39$, is not so distinct from the neighboring models with $k_{V,j}$ between 30 and 50. Finally, Figure 4 illustrates the optimal transformation selection experiments for a color channel of the marginal HSV color moments histograms. The behavior is again similar to the previous ones and the optimal model, $k_{V,J} = 12$, is quite distinct from the surrounding neighbours. Note that the likelihood curve glitches are again present in this feature space, which is an indication that the GMMs are well fitted to the data with a low number of components and that a deletion of a component leaves uncovered data, causes the likelihood jitter.

4.5 Sparse Space Transformations

Text features are high-dimensional sparse data, which pose some difficulties to parametric generative models because each parameter receives little data support. In discriminative models one observes over-fitting effects because the data representation might be too optimistic by leaving out a lot of the underlying data structure information. High-dimensional sparse data must be compressed into a lower dimensional space to ease the application of generative models. This optimal data representation is achieved with a transformation function defined as:

$$\mathrm{F_T}(d_{T,1},\ldots,d_{T,n}) = \begin{bmatrix} \mathrm{f}_{T,1}(d_{T,1,}\ldots d_T, n) \\ \vdots \\ f_T, k_T(d_{T,1},\ldots,\ d_{T,n}) \end{bmatrix}^T, k_T \ll n, \tag{20}$$

where n is the number of dimensions of the original sparse space, and k_T is the number of dimensions of the resulting optimal feature space. In other words, the sparse spaces transformation $\mathrm{F}_T(d_{T,1},\ldots,d_{T,n})$ receives as input, a feature space with n dimensions and generates a k_T dimensional feature space, where each dimension i of the new optimal feature space corresponds to the function $\mathrm{f}_{T,i}(d_{T,1},\ldots,d_{T,n})$. The optimal number of such functions will be selected by the MDL principle, and the method to estimate the functions is defined next.

4.5.1 *Text Codebook by Feature Selection.* To reduce the number of dimensions in a sparse feature space, we rank terms $t_1,\ldots,t_n$ by their importance to the modelling task and select the most important ones. The information gain criterion ranks the text terms by their importance, and the number of text terms is selected by the minimum description length. The criterion to rank the terms is the average mutual information technique, also referred to as information-gain. [Yang 1999], expressed as:

$$\mathrm{IG}(t_i) = \frac{1}{L}\sum_{j=1}^{L}\mathrm{MU}(y_i, t_i), \tag{21}$$

where t_i is term i, and y_j indicates the presence of keyword w_j. The information gain criterion is the average of the mutual information between each term and all keywords. Thus, one can see it as the mutual information between a term t_i and the keyword vocabulary. The mutual information criterion assesses the common entropy between a keyword entropy $H(y_i)$ and the keyword entropy given a term t_i, $H(y_j \mid t_i)$. Formally the mutual information criterion is defined as:

$$\mathrm{MU}(y_j, t_i) = \sum_{y_j=\{0;1\}}\sum_{d_{T,i}} p(y_j, d_{T,i})\log\frac{p(y_j, d_{T,i})}{p(y_j)p(d_{T,i})}, \tag{22}$$

where $d_{T,i}$ is the number of occurrences of term t_i in document d. Yang and Pedersen [1997] and Forman [2003] have shown experimentally that this is one of the best criteria for feature selection. A document d is then represented by k_T text terms as the mixture:

$$p\,(d) = \sum_{i=1}^{k_T}\alpha_i\,p\,(t_i \mid d) = \sum_{i=1}^{k_T}\alpha_i\frac{d_{T,i}}{|d|}, \tag{23}$$

where $d_{T,i}$ is the number of occurrences of term t_i in document d. The parameters of this mixture are the priors α_i of corresponding term t_i. This results in a total

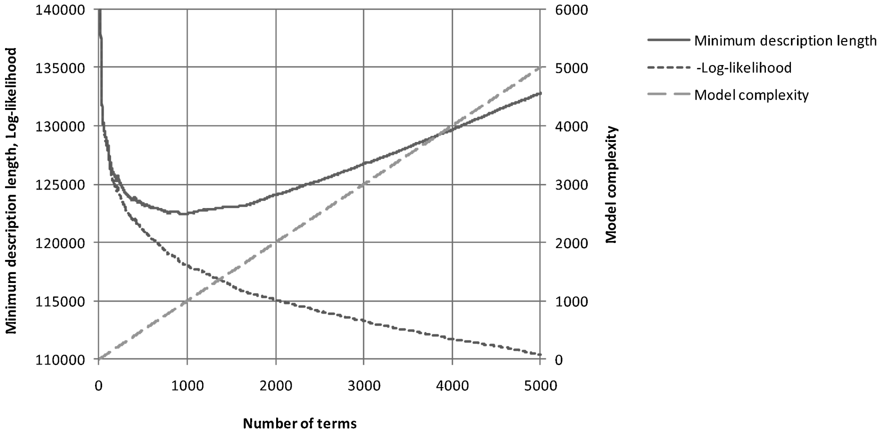

Fig. 5. Model selection for the bag-of-word features (Reuters).

number of parameters:

$$npars = k_T. \tag{24}$$

A list of models is constructed by progressively adding terms to each model according to the order established by the information gain criterion. In this particular case of sparse text features the complexity of the transformation is equivalent to the number k_T of text terms. The application of the MDL criterion in Equation (12) is now straightforward. Finally, terms are weighted by their inverse document frequency, resulting in the feature space transformation function

$$\mathrm{f}_{T,i}\left(d_T\right) = -d_{T,r(i)} \cdot \log\left(\frac{N}{\mathrm{DF}\left(d_{T,r(i)}\right)}\right), \tag{25}$$

where is the number of documents in the collection, $\mathrm{DF}(D_{T,i})$ is the number of documents containing the term t_i, and $r(i)$ is a permutation function that returns the ith text term of the information-gain rank.

4.5.2 *Experiments.* Experiments assessed the behavior of the information gain criterion on the Reuters news collection. The dataset was processed as follows. A text document is represented by the feature vector $d_T = (d_{T,1}, \ldots, d_{T,n})$ obtained from the text corpus of each document by applying several standard text processing techniques [Yang 1999]; stop words are first removed to eliminate redundant information, and rare words are also removed to avoid over-fitting [Joachims 1998]. After this, the Porter stemmer [Porter 1980] reduces words to their morphological root, which we call *term*. Finally, we discard the term sequence information and use a bag-of-words approach. These text preprocessing techniques result in a feature vector $d_T = (d_{T,1}, \ldots, d_{T,n})$, where each $d_{T,1}$ is the number of occurrences of term t_i in document d.

4.5.3 *Results and Discussion.* The evolution of the model likelihood and complexity with an increasing number of terms is again the most important characteristic that we wish to study. Figure 5 illustrates the model likelihood (dotted line) versus the model complexity (dashed line) and the minimum description length criterion as a measure of their trade-off. Note that the graph is actually showing the–log-likelihood for easier visualization and comparison.

Figure 5 illustrates the improving likelihood as new terms are added to the feature space. The curve smoothness observed in this graph is due to the scale of the x-axis (100 times greater than in the images case) and to the fact that neighbouring terms have similar information value. The problem of selecting the dimensionality of the optimal feature space is again answered by the minimum description length criterion that selects a feature space with 972 dimensions. It is interesting to notice that the MDL selects a low dimensionality reflecting a model with lower complexity than others with better likelihood but higher complexity. Note that if we had more samples (in this dataset the number of samples is limited to 7,770) we would be able to select a more complex model (remember that the MDL criterion assumes an infinite number of samples). Moreover, information gain is a feature selection method that ranks terms by their discriminative characteristics and does not actually try to faithfully replicate the data characteristics. This is in contrast with the hierarchical EM method used for the dense feature spaces that is a pure generative approach.

5. KEYWORD MODELS

Keywords are present in multimedia documents according to complex patterns that reflect their dependence and correlations. Different probability distributions can be applied to capture this information. Also Bayesian networks can be used to define complex distributions that try to represent complex keyword interactions. Section 3 discussed why the assumption of keyword independence is a good choice in multimedia information retrieval, and we define keywords as Bernoulli random variables, with:

$$p\left(y_t = 1\right) = 1 - p\left(y_t = 0\right) = \frac{\left|\mathcal{D}_{w_t}\right|}{|\mathcal{D}|}, \tag{26}$$

where y_t is a particular keyword, $|\mathcal{D}|$ is the size of the training collection, and $|\mathcal{D}_{wt}|$ is the number of documents in the training collection containing keyword w_t. In the previous section we proposed a probabilistic framework $p(y_t \mid \mathrm{F}(d), \beta_t)$, where $F(d)$ is a visual and text data transformation that creates a unified multimodal feature space, and a keyword w_t is represented in that feature space by a model β_t. We will ignore the feature type and use a plain vector to represent the low-level features of a document as:

$$\mathrm{F}(d^j) = \left(\mathrm{F}_T(d_T),\ \mathrm{F}_V(d_V)\right) = \left(f_1^j, \ldots, f_M^j\right). \tag{27}$$

One of the goals of the proposed F(d) transformation is the creation of an optimal feature space, where simple and scalable keyword models β_t can be used. This section will propose the application of linear models to address this particular problem. The setting is a typical supervised learning problem, where documents are labelled with the keywords that are present in that document. Thus, we define:

$$y^j = \left(y_1^j, \ldots, y_L^j\right), \tag{28}$$

as the binary vector of keyword annotations of document j, where each y_t^j indicates the presence of keyword w_t in document j if $y_t^j = 1$. Note that a perfect classifier would have $(y - d_w) = 0$ on a new document. The annotations vector y^j is used to estimate keyword models and to test the effectiveness of the computed models.

5.1 Keywords as Logistic Regression Models

Logistic regression is a statistical learning technique that has been applied to a great variety of fields, for example, natural language processing [Berger et al. 1996], text classification [Nigam et al. 1999], and image annotation [Jeon and Manmatha 2004]. In this section we employ a binomial logistic model to represent keywords in the multimodal feature space. The expression of the binomial logistic regression is:

$$p\left(y_t = 1 \mid \mathrm{F}\left(d\right), \beta_t\right) = \frac{1}{1 + \exp\left(\beta_t \cdot \mathrm{F}\left(d\right)\right)}. \tag{29}$$

The logistic regression model is also a linear model, which makes it a scalable and efficient solution for modelling keywords. The theory of Generalized Linear Models shows how to derive the logistic regression expression from the point of view of pure linear models [McCullagh and Nelder 1989].

5.1.1 *Regularization.* As discussed by Nigan et al. [1999] and Chen and Rosenfeld [1999], logistic regression may suffer from over-fitting. This usually occurs because features are high-dimensional and sparse, meaning that the regression coefficients can easily push the model density towards some particular training data points. Zhang and Oles [2001] have also presented a study on the effect of different types of regularization on logistic regression. Their results indicate that with the adequate cost function (regularization), precision results are comparable to SVMs with the advantage of rendering a probabilistic density model. An efficient and well known method of tackling over-fitting is to set a prior on the regression coefficients. As suggested by Nigan et al. [1999] and Chen and Rosenfeld [1999] we use a Gaussian prior, $\mathcal{N}_\xi$, for the regression coefficients, with mean $\mu_\xi = 0$ and σ_ξ^2 variance. The Gaussian prior imposes a cost on models β_* with large norms, thus preventing optimization procedures from creating models that depend too much on a single feature space dimension.

5.1.2 *Maximum Likelihood Estimation.* The log-likelihood function computes the sum of the log of the errors of each document in the collection $\mathcal{D}$. For each keyword model the likelihood function tells us how well the model and those parameters represent the data. The model is estimated by finding the minimum of the likelihood function by taking the regression coefficients as variables:

$$\beta_t = \min_{\beta} l\left(\beta \mid \mathcal{D}\right) = \min_{\beta} \sum_{j \in \mathcal{D}} \log\left(p(y_t^j \mid \mathrm{F}(d^j), \beta_t) p(\beta_t | \sigma_\xi^2)\right). \tag{30}$$

For models where the solution can be found analytically, the computation of the regression coefficients is straightforward. In cases where the analytical solution is not available, typical numerical optimization algorithms are adequate.

The regression coefficients need to be found by a numerical optimization algorithm that iteratively approaches a solution corresponding to a local minimum of the log-likelihood function. To find the minimum of the log-likelihood function $l(\beta)$ with respect to β, we use the Newton-Raphson algorithm:

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 l\left(\beta^{old}\right)}{\partial\beta\partial\beta^T}\right)^{-1}\left(\frac{\partial l\left(\beta^{old}\right)}{\partial\beta}\right). \tag{31}$$

The first-order derivative matrix is a vector with M elements corresponding to the dimension of the space resulting from the application of $\mathrm{F}(d)$ to the original data. The second-order derivative, the Hessian matrix, is a square-matrix with $M \times M$ components. The Hessian matrix imposes a high computational complexity (both in time and space) on the parameter estimation algorithm. In multimedia information retrieval we use feature spaces with thousands of dimensions, meaning that the processing of the Hessian matrix is computationally too costly. For these reasons, we must use algorithms that are more suitable for such a large-scale problem.

5.1.3 *Large-Scale Model Computation.* When applying the Newton-Raphson algorithm to high-dimensional data, the Hessian matrix often cannot be computed at a reasonable cost because it is too large and dense. Large scale Quasi-Newton methods are an adequate solution for our problem. Instead of storing and computing the full Hessian matrix, these methods store a few vectors that represent approximations implicitly made in previous iterations of the algorithm. The L-BFGS algorithm (limited-memory Broyden-Fletcher-Goldfarb-Shanno) is one of such algorithms; see Liu and Nocedal [1989a] for details: “The main idea of this method is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations, which is less likely to be relevant to the actual behavior of the Hessian at the current iteration, is discarded in the interest of saving storage.” The L-BFGS algorithm iteratively evaluates the log-likelihood function and its gradient, and updates the regression coefficients and the Hessian approximation. For the binomial logistic regression the log-likelihood function is:

$$\begin{aligned} l\left(\beta_t\right) &= \sum_{d^j \in \mathcal{D}} \left(y_t^j \beta_t \mathrm{F}\left(d^j\right) - \log\left(1 + \exp\left(\beta_t \mathrm{F}\left(d^j\right)\right)\right)\right) - \lambda\beta_t^2, \\ \lambda &= \frac{1}{2\sigma_\xi^2}, \end{aligned} \tag{32}$$

where for each example d^j the variable y_t^j is 1 if the example contains the keyword w_t and 0 otherwise. $\mathrm{F}(d^j)$ is the nonlinear space transformation of the document features. To minimize the log-likelihood we need to use the gradient information to find the β_t where the log-likelihood gradient is zero:

$$\frac{\partial l\left(\beta_t\right)}{\partial \beta_t} = 0 = \sum_{d^j \in \mathcal{D}} \mathrm{F}\left(d^j\right)\left(y_t^j - p\left(y_t^j = 1 | \beta_t, \mathrm{F}\left(d^j\right)\right)\right) - \lambda\beta_t. \tag{33}$$

These two last equations are the binomial logistic regression functions that the L-BFGS algorithm evaluates on each iteration to compute the β_t regression coefficients.

We use the implementation provided by Liu and Nocedal [1989b] to estimate the parameters of both linear logistic models and log-linear models. It has been shown that L-BFGS is the best optimization procedure for both maximum entropy [Malouf 2002] and conditional random fields models [Sha and Pereira 2003]. For more details on the limited-memory BFGS algorithm see [Nocedal and Wright 1999].

5.2 Keyword Baseline Models

The linear models that we present in this section are simple but effective models that can be applied in the multimodal feature space. The advantage of both Rocchio classifier and naïve Bayes classifier is that they can be computed analytically.

5.2.1 *Rocchio Classifier.* The Rocchio classifier was initially proposed as a relevance feedback algorithm to compute a query vector from a small set of positive and negative examples [Rocchio 1971]. It can also be used for categorization tasks, for example Joachims [1997] a keyword w_t is represented as a vector β_t in the multimodal space, and the closer a document is to this vector the higher the similarity between the document and the keyword. A keyword vector β_t is computed as the average of the vectors of both relevant documents $\{\mathcal{D}_{wt}\}$ and nonrelevant documents $\{\mathcal{D}\backslash\mathcal{D}_{w_t}\}$ (see Magalhães [2008] for details). The Rocchio classifier is a simple classifier that has been widely used in the area of text information retrieval and, as we have shown, can also be applied to semantic-multimedia information retrieval. Moreover, this classifier is particularly useful for online learning scenarios and other interactive applications where the models need to be updated on-the-fly or the number of training examples is limited.

5.2.2 *Naïve Bayes Model.* The naïve Bayes classifier assumes independence between feature dimensions and is the result of the direct application of Bayes' law to classification tasks,

$$p\left(y_t = 1 \mid d\right) = \frac{p\left(y_t = 1\right) p\left(d = f_1, \ldots, f_M | y_t = 1\right)}{p\left(d\right)}. \tag{34}$$

The assumption that features f_i are independent of each other in a document can be modelled by several different independent probability distributions. A distribution is chosen according to some constraints that we put on the independence assumptions. For example, if we assume that features f_i can be modelled as simple presence or absence in a document then we consider a binomial distribution. If we assume that features f_i can be modelled as discrete values to indicate the presence confidence in a document then we consider a multinomial distribution see McCallum and Nigam [1998]. The binomial distribution over features f_i would be too limiting; the multinomial distribution over features f_i offers greater granularity to represent a feature value. One can compute the log-odds and classify a document with the keywords that have a value greater than zero:

$$\log \frac{p\left(w_j = 1 \,|\, d\right)}{p\left(w_j = 0 \,|\, d\right)} = \log \frac{p(y_t = 1)}{p(y_t = 0)} + M \sum_{i=1}^{M} p(f_i \,|\, d) \log \frac{p(f_i \,|\, y_t = 1)}{p(f_i \,|\, y_t = 0)}. \qquad (35)$$

Formulating naïve Bayes in log-odds space has two advantages: it shows that naïve Bayes is a linear model and avoids decision thresholds in multicategorization problems. In this case the keyword models become:

$$\beta_{t,i} = \log \frac{p(f_i \,|\, y_t = 1)}{p(f_i \,|\, y_t = 0)}, \quad i = 1, ..., M. \qquad (36)$$

6. EVALUATION

The presented algorithms were evaluated with a retrieval setting on the Reuters-21578 collection, on a subset of the Corel Stock Photo CDs [Duygulu et al. 2002], and on a subset of the TRECVID2006 development data.

6.1 Collections

6.1.1 *Reuters-21578.* This is a widely used text dataset that allows comparison of our results with others in the literature. Each document is composed of a text corpus, a title (which we ignore) and labelled categories. This dataset has several possible splits and we have used the *ModApte* split, which contains 9603 training documents and 3299 test documents. This is the same evaluation setup used in several other experiments [Joachims 1998; Nigam et al. 1999; McCallum and Nigam 1998; Zhang and Oles 2001]. Terms appearing less than 3 times were removed. Only labels with at least 1 document on the training set and the test set were considered, leaving us with 90 labels. After these steps we ended with 7770 labelled documents for training.

6.1.2 *Corel Images.* This dataset was compiled by Duygulu et al. [2002] from a set of COREL Stock Photo CDs. The dataset has some visually similar keywords (jet, plane, Boeing), and some keywords have a limited number examples (10 or less). In their seminal paper, the authors acknowledge that fact and ignored the classes with these problems. In this article we use the same setup as in Yavlinsky et al. [2005], Carneiro and Vasconcelos [2005], Jeon et al. [2003], Lavrenko et al. [2003], and Feng et al. [2004], which differs slightly from the one used in the dataset in the original paper, [Duygulu et al. 2002]. The retrieval evaluation scenario consists of a training set of 4500 images and a test set of 500 images. Each image is annotated with 1–5 keywords from a vocabulary of 371 keywords. Only keywords with at least 2 images in the test set and training set each were evaluated, which reduced the amount of vocabulary to 179 keywords. Retrieval lists have the same length as the test set: 500 items.

Table I. MAP results for Reuters and Corel collections

	Rocchio	NBayes	LogReg
Reuter	0.497	0.245	0.490
Corel	0.219	0.243	0.279

6.1.3 *TRECVID2006.* To test the similarity ranking on multimodal data, we used the TRECVID2006 data. Since only the training set is completely labelled, we randomly split the training English videos into 23,709 training documents and 12,054 test documents. We considered each document to be a key-frame plus the ASR text within a window of 6 seconds around that key-frame. Key-frames are annotated with the standard vocabulary of 39 keywords provided by NIST.

6.2 Experiment Design

To evaluate the proposed framework we designed a retrieval experiment for all collections listed in the previous section. The experimental methodology was as follows.

For a given algorithm and a given a multimodal feature space:

(1) For each keyword in the considered collection,
  (a) estimate the keyword model on the training set by applying a cross-validation with 5 folds and 10 value iterations, as suggested in Kohavi [1995], to determine the ideal Gaussian prior variance σ_ξ^2.
  (b) Compute the relevance of each test document.
  (c) Rank all test documents by their relevance for the considered keyword.
(2) Use the collection relevance judgements to measure the retrieval effectiveness of the considered rank,
  (a) repeat step 1(a) for all keywords.
  (b) Compute the mean average precision.
(3) Repeat for a different algorithm or a for different multimodal feature space.

This methodology was repeated for all linear models that we presented in this section and for different multimodal feature spaces. We considered the Reuters-21578 collection, the Corel5000 collection, the ASR part of the TRECVID2006, the key-frames of the TRECVID2006, and both key-frames and text of the TRECVID2006 development data, which makes five collections.

6.3 Text-Only Models

6.3.1 *Retrieval Effectiveness.* Experiments in the Reuters dataset were evaluated with mean average precision, Table I, and precision-recall curves, Figure 6. All results were obtained with a 972-dimensional multimodal feature space selected by the minimum description length criterion.

When comparing the naïve Bayes model to the logistic regression model, results show that naïve Bayes performs much worse than logistic regression (24.5% MAP versus 49.0% MAP). However, it is a surprise to see that the Rocchio classifier is actually comparable to logistic regression—it obtained

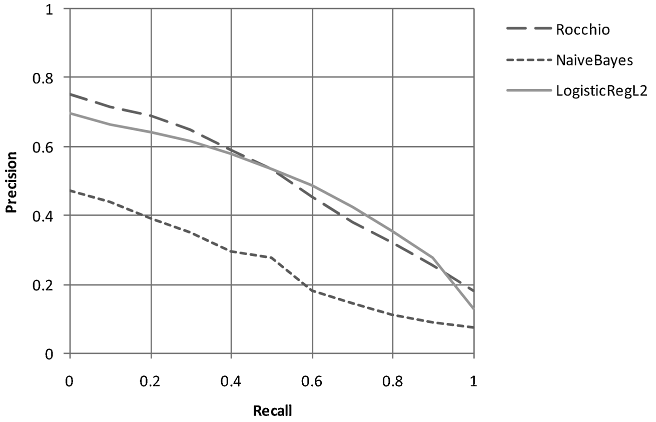

Fig. 6. Precision-recall curve evaluation on the Reuters-21578.

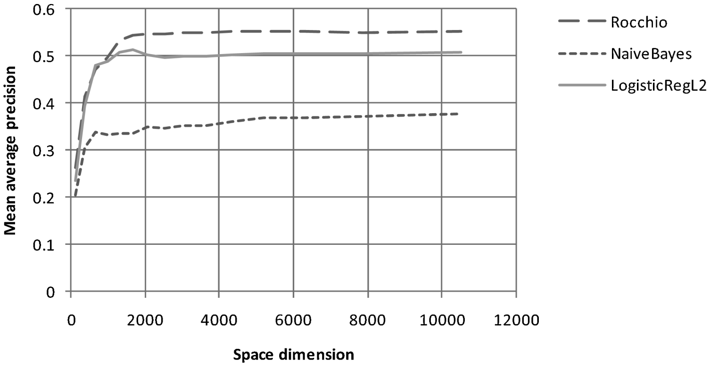

Fig. 7. Retrieval precision for different space dimensions (text-only models).

49.7%. This supports the hypothesis that the Reuters data is structured in a single cluster shape. Another reason why the Rocchio classifier performs so well on this dataset is that of all three classifiers it is the one that uses the simplest assumptions about data (organized as a high-dimensional sphere). The implications are that it is less prone to over-fit on classes with few training examples, unlike logistic regression. The precision-recall curves in Figure 6, offer a more detailed comparison of the models and confirm that logistic regression and Rocchio are very similar.

6.3.2 *Model Complexity Analysis.* We also studied the effect of the optimal space dimensionality by measuring the MAP on different spaces. The different multimodal feature spaces were obtained by progressively adding new terms according to the average mutual information criterion. Figure 7 shows that after some number of terms (space dimension), precision does not increase because information carried by new terms is already present in the previous ones. The graph confirms that Rocchio is consistently better than logistic regression. Note that the MDL point (972 terms) achieves a good trade-off between the model complexity and the model retrieval effectiveness.

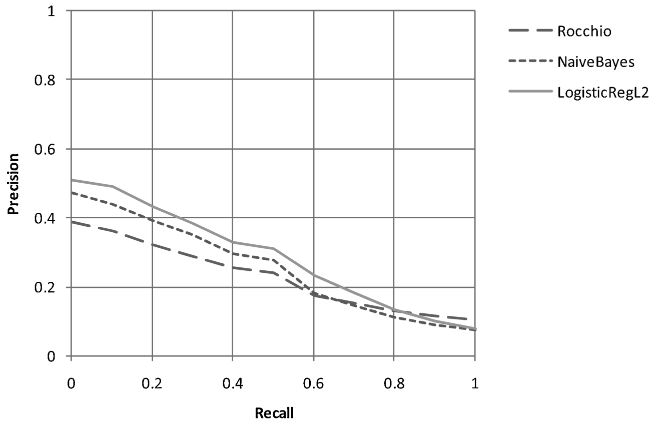

Fig. 8. Precision-recall curves for different keyword models.

Table II. MAP Comparison with other Algorithms (Corel)

Algorithm	MAP	L
Cross-Media Relevance Model [Jeon et al. 2003]	16.9%	179
Continuous-space Relevance Model [Lavrenko et al. 2003]	23.5%	179
Naïve Bayes	24.3%	179
LogisticRegL2	27.9%	179
Nonparametric Density Distribution [Yavlinsky et al. 2005]	28.9%	179
Multiple-Bernoulli Relevance Model [Feng et al. 2004]	30.0%	260
Mixture of Hierarchies [Carneiro and Vasconcelos 2005]	31.0%	260

6.4 Image-Only Models

6.4.1 *Retrieval Effectiveness.* We first used the MDL criterion to select a multimodal feature space and then ran the retrieval experiments for all linear models. The space selected by the MDL criterion has 2989 dimensions. The MAP measures shown in Table I shows that the best performance is achieved by the logistic regression models with 27.9%, followed by naïve Bayes with 24.3% and Rocchio with 21.9%. Contrary to the Reuters collection, the more complex structure of the Corel Images dataset has affected the performance of the Rocchio classifier. Thus, both naïve Bayes, and more specifically, logistic regression can better capture the structure of this data. The precision-recall curves in Figure 8 show that logistic regression is better than Rocchio and naïve Bayes across most of the recall area. Results on this collection are more in agreement with what one would expect from the complexity of each model. Naïve Bayes applies a Gaussian on each dimension of the feature space, which demonstrates a more accurate assumption than the single cluster assumption made by the Rocchio classifier. Finally, logistic regression can better capture the non-Gaussian patterns of the data and achieve a better performance. Table II compares some of the published algorithms' MAPs on the Corel collection. Note that some algorithms consider keywords with only training 1 example and 1 test example, thus resulting in 260 keywords instead of the 179 keywords. Methods that used the 260 keywords are some types of nonparametric density distributions that can easily model classes with a small number

Table III. Annotation Performance for an Image with 192 × 128 Pixels

Task	Time(ms)
margHSV (9 tiles)	30
Tamura (9 tiles)	54
Gabor (9 tiles)	378
Annotation of 179 Keywords	9

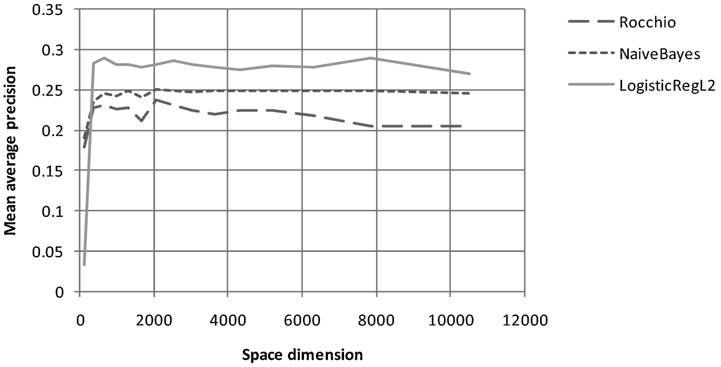

Fig. 9. Retrieval precision for different space dimensions.

of examples. This table also shows how the proposed algorithm achieves a retrieval effectiveness that is in the same range as other state-of-the-art algorithms.

6.4.2 *Time Complexity.* The time complexity of the proposed framework is a crucial characteristic for multimedia indexing tasks. For this reason we carefully chose algorithms that can handle multimedia semantics with little computational complexity. Table III illustrates the times required to extract the visual-features and to run the semantic-multimedia analysis algorithm. Measures were done on an AMD Athlon 64 running at 3.7GHz. Note that these values are for the inference phase and not for the learning phase.

6.4.3 *Model Complexity Analysis.* Figure 9 depicts the evolution of the mean average precision with respect to the dimensionality of the multimodal feature space. Each point on the curve reflects the different levels of model complexity of the output of the hierarchical EM. Remember that the multimodal feature space is the concatenation of the hierarchical EM Gaussian mixture models of the different feature subspaces. We concatenate subspaces with a similar number of levels of complexity, for example, GMMs with the same number of components per feature subspace. For low dimensional multimodal spaces, the MAP values for all models are quite low. Only when the dimensionality increases does the MAP achieve more stable values. The MAP stabilizes because the more complex GMM models do not allow better discrimination between the relevant and nonrelevant examples. The same phenomenon was observed for the Reuters collection.

Table IV. MAP Results for TRECVID Collection

Model	Text	Images	Multimodal
Rocchio	0.148	0.234	0.240
NBayes	0.174	0.257	0.273
LogReg	0.203	0.273	0.295

Table V. MAP Comparison with other Algorithms (TRECVID)

Algorithm	MAP	L	Modalities	Videos
LogisticRegL2	27.3%	39	V	English
Non-parametric Density Distribution [Yavlinsky, Schofield and Rüger, 2005]	21.8%	10	V	All
LogisticRegL2	29.5%	39	V+T	English
SVM [Chang et al., 2005]	26.6	10	V+T	All

6.5 Multimodal Models

6.5.1 *Retrieval Effectiveness.* We first applied the MDL criterion to select a multimodal feature space and then ran the retrieval experiments for all linear models. The space selected by the MDL criterion has 5670 dimensions for the visual modality, 10,576 for the text modality, and the multimodal space has a total of 16,247 dimensions. For the text modality, the MDL selects the maximum number of terms because some of the key-frames have no ASR.

Table IV presents a summary of the retrieval effectiveness evaluation in terms of MAP. All types of keyword models show the same variation with respect to each modality: text-based models are always much lower than the image-based models, and the difference between image-based models and multimodal models is always small. Moreover, logistic regression models are always better than naïve Bayes and Rocchio. This confirms previous knowledge that the TRECVID collection is more difficult and its data exhibits a more complex structure, which is why logistic regression can exploit the non-Gaussian patterns of data. It achieves 20.2% MAP on the text-only experiment, 27.3% on the image-only experiment, and 29.5% on the multimodal experiment. Multimodal models, Figure 10, show that naïve Bayes models better exploit the higher number of information sources than the Rocchio classifier. This is not a surprise as naïve Bayes considers individual dimensions, and the data structure is more complex than the spherical structure assumed by Rocchio. Also related to this phenomenon is the retrieval effectiveness obtained by the logistic regression model.

Table V compares the proposed algorithm to two TRECVID submissions that attained an MAP above the median and all keywords are modeled with the same algorithm (some TRECVID systems employ a different algorithm for each keyword). Note that our results were obtained for more keywords (39 instead of 10) and less training data (just English), so results are a rough indication of how our method compares to others. We limited the amount of training data due to computational reasons. However, as we can see from the table, the proposed approach is competitive with approaches that were trained in more advantageous conditions (fewer keywords).

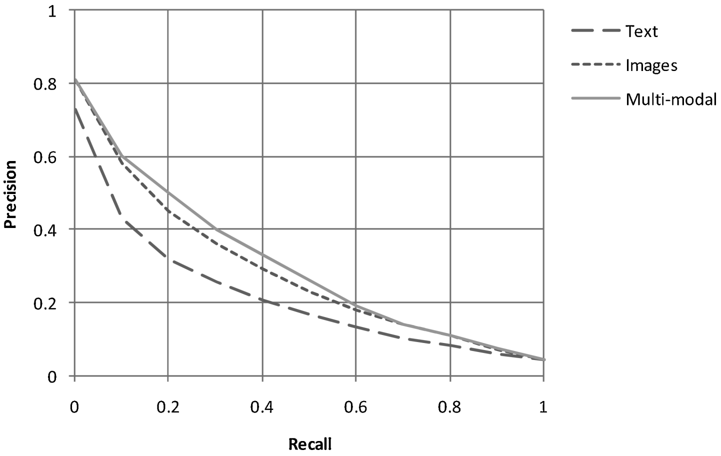

Fig. 10. Precision-recall curve for multimodal models.

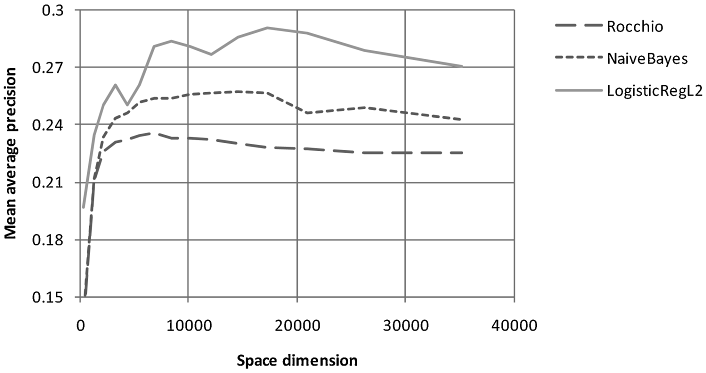

Fig. 11. MAP vs space dimension for multimodal models.

6.5.2 *Model Complexity Analysis.* For the second experiment, we studied the effect of the complexity of the feature space transformations—the number of dimensions of the optimal feature space. Multimodal-based models, Figure 11, exhibit a more irregular trend than the single-media models. The higher dimensionality and feature heterogeneity might be the cause for this phenomenon. The differences between the three models are related to the respective modelling capabilities. Rocchio assumes a spherical structure, which has been shown to be too simplistic for this data; naïve Bayes assumed independent dimensions, which is also not the best model for this data; finally, logistic regression further exploits feature dimension interactions and linear combinations of them. Logistic regression, with an adequate cross-validation procedure, appears to achieve the best retrieval effectiveness.

7. CONCLUSIONS

The creation of the multimodal feature space is a generalization procedure that results in a trade-off between accuracy and computational complexity. Thus, the described algorithm offers an appealing solution for applications that require an information extraction algorithm with good precision, scalability, flexibility, and robustness. The novelty of the proposed framework resides in the simplicity of the linear combination of the heterogeneous sources of information that were selected by the minimum description length criterion.

—*Retrieval Effectiveness*. The experiments show that our framework offers performance in the same range as other state-of-the-art algorithms. Text and image results are quite good while multimodal experiments were affected by the noise present on the speech text and by the higher number of parameters to estimate. It was not surprising to see that logistic regression attains better results than naïve Bayes at the expense of a higher learning cost.

—*Model Selection*. The algorithm's immunity to overfitting is illustrated by the MAP curve stability as the model complexity increases. Logistic regression can be interpreted as an ensemble method (additive model) if we consider each dimension as a weak learner, and the final model as a linear combination of those weak learners. This means that our model has some of the characteristics of additive models, namely the observed immunity to overfitting. It is interesting to note that the simple naïve Bayes model appears to be more immune to overfitting than the logistic regression model. This occurs because the optimization procedure fits the model tightly to the training data favoring large regression coefficients, while the naïve Bayes avoids overfitting by computing the weighted average of all codewords (dimensions). Note that when fitting the model we are minimizing a measure of the model log-likelihood (the average classification residual error) and not a measure of how documents are ranked in a list (average precision). The mean average precision is the mean of the accumulated precision over a ranked list. Thus, we believe that if we trained our models with average precision as our goal metric, the retrieval results on the test set would improve.

—*Computational Scalability*. Since the optimal feature space is common to all keywords the transformation must be computed only once for all keywords. Thus the resources required to evaluate the relevance of a multimedia document for each keyword are relatively small. During classification, both time and space complexity of the data representation algorithms are given by the number of Gaussians (clusters) selected by the model selection criteria. The computational complexity of linear models during the classification phase is negligible, resulting in a very low computational complexity for annotating multimedia content and making it quickly searchable. The computational complexity during the learning phase is dominated by the hierarchical EM algorithm of a mixture of Gaussians and the cross-validation method. The worst-case space complexity during learning is proportional to the maximum number of clusters, the number of samples, the dimension of each feature, and the total number of cross-validation iterations and folds. we consider this cost to be less important because the learning can be done offline. Apart from the mixture of hierarchies [Carneiro and Vasconcelos 2005] all other methods are kinds of kernel density distributions. It is well known [Hastie et al. 2001] that the nature of these methods makes the task of running these models on new data computationally demanding. The model corresponds to the entire training set, meaning that the demand on CPU time and memory increases with the training data. Results show that such a low complexity approach compares competitively with much more complex approaches. It has a bearing on the design of image search engines, where scalability and response time are as important as the actual mean average precision of the returned results.

—*Semantic Scalability*. Assuming that the set of keywords used is a faithful sample of a larger keyword vocabulary it is expected that one can use the same optimal feature space to learn the linear model of new keywords and preserve the same models. Note that the optimal feature space is a representation of the data feature space—it is selected based on the entire data and independently of the number keywords. The consequence of this design is that systems can be semantically scalable in the sense that new keywords can be added to the system without affecting previous annotations.

REFERENCES

Amir, A., Argillander, J. O., Campbell, M., Haubold, A., Iyengar, G., Ebabdollahi, S., Kang, F., Naphade, M., Natsev, A., Smith, J. R., Tesic, J., and Volkmer, T. 2005. IBM Research TRECVID-2005 video retrieval system. In *Proceedings of the TREC Video Retrieval Evaluation Workshop*.

Argillander, J., Iyengar, G., and Nock, H. 2005. Semantic annotation of multimedia using maximum entropy models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Proceessing*.

Barnard, K., and Forsyth, D. A. 2001. Learning the semantics of words and pictures. In *Proceedings of the International Conference on Computer Vision*.

Barron, A., and Cover, T. 1991. Minimum complexity density estimation. *IEEE Trans. Inf. Theory 37*, 4, 1034–1054.

Berger, A., Pietra, S., and Pietra, V. 1996. A maximum entropy approach to natural language processing. *Computational Ling. 37*, 39–71.

Blei, D., and Jordan, M. 2003. Modeling annotated data. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Carneiro, G., and Vasconcelos, N. 2005. Formulating semantic image annotation as a supervised learning problem. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*.

Chang, S.-F., Hsu, W., Kennedy, L., Xie, L., Yanagawa, A., Zavesky, E., and Zhang, D.-Q. 2005. Columbia University TRECVID-2005 video search and high-level feature extraction. In *Proceedings of TRECVID*.

Chen, S. F., and Rosenfeld, R. 1999. A Gaussian prior for smoothing maximum entropy models. Tech. rep. Carnegie Mellon University, Pittsburg. PA.

Cover, T. M., and Thomas, J. A. 1991. *Elements of Information Theory*. John Wiley & Sons.

Duygulu, P., Barnard, K., de Freita, N., and Forsyth, D. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the European Conference on Computer Vision*.

Feng, S. L., Lavrenko, V., and Manmatha, R. 2004. Multiple Bernoulli relevance models for image and video annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Figueiredo, M., and Jain, A. K. 2002. Unsupervised learning of finite mixture models. *IEEE Trans. Patt. Anal. Mach. Intell. 24*, 3, 381–396.

Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *Mach. Learn. Res.* 1289–1305.

Hastie, T., Tibshirani, R., and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer Series in Statistics: Springer.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jeon, J., Lavrenko, V., and Manmatha, R. 2003. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jeon, J., and Manmatha, R. 2004. Using maximum entropy for automatic image annotation. In *Proceedings of the International Conference on Image and Video Retrieval*.

Joachims, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the International Conference on Machine Learning*.

Joachims, T. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *European Conference on Machine Learning*.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Lavrenko, V., Manmatha, R., and Jeon, J. 2003. A model for learning the semantics of pictures. In *Proceedings of the Neural Information Processing System Conference*.

Lazebnik, S., Schmid, C., and Ponce, J. 2005. A maximum entropy framework for part-based texture and object recognition. In *Proceedings of the International Conference on Computer Vision*.

Liu, D. C., and Nocedal, J. 1989a. On the limited memory BFGS method for large scale optimization. *Math. Prog. 45*, 503–528.

Liu, D. C., and Nocedal, J. 1989b. On the limited memory method for large scale optimization. *Mathe. Prog. B 45*, 3, 503–528.

MacKay, D. J C. 2004. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Magalhães, J. 2008. *Statistical Models for Semantic-Multimedia Information Retrieval*. PhD Thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London.

Magalhães, J., and Rüger, S. 2007. Information-theoretic semantic multimedia indexing. In *Proceedings of the ACM Conference on Image and Video Retrieval*.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning.* 49–55.

McCallum, A., and Nigam, K. 1998. A comparison of event models for naive Bayes text classification. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*.

McCullagh, P., and Nelder, J. A. 1989. *Generalized Linear Models*. 2nd Ed: Chapman and Hall.

Naphade, M. R., and Huang, T. S. 2001. A probabilistic framework for semantic video indexing filtering and retrieval. *IEEE Trans. Multimedia. 3,* 1, 141–151.

Nigam, K., Lafferty, J., and McCallum, A. 1999. Using maximum entropy for text classification. In *Proceedings of the IJCAI - Workshop on Machine Learning for Information Filtering*.

Nocedal, J., and Wright, S. J. 1999. *Numerical Optimization*. Springer-Verlag.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program. 14*, 3, 130–137.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica. 14*, 465–471.

Rocchio, J. 1971. Relevance feedback in information retrieval. In G. Salton (Ed) *The SMART Retrieval System: Experiments in Automatic Text Retrieval*, Prentice-Hall.

Sha, F. and Pereira, F. 2003. Shallow parsing with conditional random fields. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Snoek, C. G. M., v. Gemert, J. C., Gevers, T., Huurnink, B., Koelma, D. C., v. Liemp, M., d. Rooij, O., d. Sande, Seinstra, F. J., Smeulder, A. W. M., Thean, A. H. C., Veenman, C. J., and Worring, M. 2006. The MediaMill TRECVID 2006 semantic video search engine. In *Proceedings of the TREC Video Retrieval Evaluation Workshop*.

Snoek, C. G. M., Worring, M., Geusebroe, J.-M., Koelma, D. C., Seinstra, F. J., and Smeulders, A. W. M. 2006b. The semantic pathfinder: using an authoring metaphor for generic multimedia indexing. *IEEE Trans. Patt. Anal. Mach. Intell. 28*, 10, 1678–1689.

Vailaya, A., Figueiredo, M., Jain, A. K., and Zhang, H. J. 2001. Image classification for content-based indexing. *IEEE Trans. Image Process. 10* 1, 117–130.

Westerveld, T., and de Vrie, A. P. 2003. Experimental result analysis for a generative probabilistic image retrieval model. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Westerveld, T., de Vrie, A. P., Ianeva, T., Boldareva, L., and Hiemstra, D. 2003. Combining information sources for video retrieval. In *Proceedings of the TREC Video Retrieval Evaluation Workshop*.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Inform. Retriev. 1*, 1/2, 67–88.

Yang, Y., and Chute, C. G. 1994. An example-based mapping method for text categorization and retrieval. *ACM Trans. Inform. Syst. 13*, 3, 252–277.

Yang, Y., and Liu, X. 1999. A re-examination of text categorization methods. In *Proceedings of the ACM Special Interest Group on Information Retrieval* (SIGIR).

Yang, Y., and Pedersen, J. O. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the Conference on Machine Learning*.

Yavlinsky, A., Schofield, E., and Rüger, S. 2005. Automated image annotation using global features and robust nonparametric density estimation. In *Proceedings of the International Conference on Image and Video Retrieval*.

Zhang, T., and Oles, F. J. 2001. Text categorization based on regularized linear classification methods. *Inform. Retrieval. 4*, 5–31.

Received December 2008; revised October 2009; accepted March 2010