Weighted Pseudometric Discriminatory Power Improvement Using a Bayesian Logistic Regression Model Based on a Variational Method

Riadh Ksantini, Djemel Ziou, Bernard Colin, and Francois Dubeau

Abstract—In this paper, we investigate the effectiveness of a Bayesian logistic regression model to compute the weights of a pseudometric in order to improve its discriminatory capacity and thereby increase image retrieval accuracy. In the proposed Bayesian model, the prior knowledge of the observations is incorporated and the posterior distribution is approximated by a tractable Gaussian form using variational transformation and Jensen's inequality, which allow a fast and straightforward computation of the weights. The pseudometric makes use of the compressed and quantized versions of wavelet decomposed feature vectors, and in our previous work, the weights were adjusted by the classical logistic regression model. A comparative evaluation of the Bayesian and classical logistic regression models is performed for content-based image retrieval, as well as for other classification tasks, in a decontextualized evaluation framework. In this same framework, we compare the Bayesian logistic regression model to some relevant state-of-the-art classification algorithms. Experimental results show that the Bayesian logistic regression model outperforms these linear classification algorithms and is a significantly better tool than the classical logistic regression model to compute the pseudometric weights and improve retrieval and classification performance. Finally, we perform a comparison with results obtained by other retrieval methods.

Index Terms—Image retrieval, logistic regression, variational method, weighted pseudometric.

1 Introduction

The rapid expansion of the Internet and the wide use of digital data in many real-world applications in the fields of medicine, weather prediction, communications, commerce, and academic research has increased the need for efficient image database creation and retrieval procedures. The content-based image retrieval (CBIR) approach was proposed to meet this need [1], [2]. In this approach, the first step is to compute, for each database image, a feature vector that captures certain visual features of the image, such as color, texture, and shape. This feature vector is stored in a featurebase, and then, given a query image chosen by a user, its feature vector is computed and compared to the featurebase feature vectors by a similarity measure, and finally, the database images most similar to the query image are returned to the user. Distance measures like the nearest neighbor rule distance and the euclidean distance have been widely used for feature vector comparison in CBIR systems. However, these similarity measures are based only on the distances between feature vectors in the feature space, and they blindly assume that features have the same relevance by giving them the same weight. Moreover, they do not capitalize on any statistical regularities in the data that might be estimated from a large training set of relevance and irrelevance classes. Therefore, distance measures can fail, and irrelevant features may hurt retrieval performance. Statistical approaches are a promising solution to this CBIR problem [3], [27], and they can lead to a significant gain in retrieval accuracy. In fact, these approaches are capable of generating probabilistic similarity measures and highly customized metrics (learned metrics) for computing image similarity based on the consideration of and distinction among feature relevances. This literature is too wide to survey here, but in this section, we review some relevant work based on these statistical approaches. For work using probabilistic similarity measures, we review these relevant examples: Caenen and Pauwels [6] use the classical quadratic logistic regression model in order to classify database image feature vectors as relevant or irrelevant. Based on this classification, a total relevance probability is generated for each image in the database. This total relevance probability is a linear combination of weights used to fine-tune the influence of each individual feature with the natural logarithms of the logistic relevance probabilities of the feature vector components. Database images are ranked according to their total relevance probabilities. Aksoy and Haralick [5] investigate the effectiveness of five different normalization methods in combination with two different likelihood-based similarity measures that compute the likelihood of two images being similar or dissimilar, one being the query image and the other one being an image in the database. First, two classes are defined, the

- *R. Ksantini and D. Ziou are with the Département d'Informatique, Faculté des Sciences, 2500 Bl. Université, Université de Sherbrooke, J1K2R1, Sherbrooke, Québec, Canada.*
E-mail: {riadh.ksantini, djemel.ziou}@usherbrooke.ca.
- *B. Colin and F. Dubeau are with the Département de Mathématiques, Faculté des Sciences, 2500 Bl. Université, Université de Sherbrooke, J1K2R1, Sherbrooke, Québec, Canada.*
E-mail: {bernard.colin, francois.dubeau}@usherbrooke.ca.

Manuscript received 8 Mar. 2006; revised 3 Nov. 2006; accepted 21 Feb. 2007; published online 20 June 2007.
Recommended for acceptance by M. Figueiredo.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0214-0306.
Digital Object Identifier no. 10.1109/TPAMI.2007.1165.

0162-8828/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

relevance class and the irrelevance class, and then, the likelihood values are derived from the Bayesian classifier. Two different methods are used to estimate the conditional probabilities used in the classifier. The first method uses multivariate normal assumption, and the second one uses independently fitted distributions for each feature. The degree of similarity between a query image and a database image is measured by the likelihood ratio. Vasconcelos [22] adopts the minimum probability of error (MPE) as the optimality criterion and formulates retrieval as a problem of statistical classification. He shows that the Bayesian classifier is the optimal similarity function for MPE retrieval systems, as it minimizes the probability of retrieval error. Also, he proposes a new algorithm for MPE feature design that scales to problems containing a large number of classes. Westerveld and de Vries [23] present the use of generative probabilistic models for image retrieval. They estimate Gaussian mixture models to describe the visual content of images and explore two different approaches for using them for retrieval. These two approaches are called query generation (How likely is the query given the document (image) model?) and document generation (How likely is the document given the query model?) and are fitted in a common probabilistic framework. In each approach a variant is computed using the Gaussian mixture models and then used for image ranking. The query generation variant is shown to be more appropriate for ranking than the document generation variant. Lavrenko et al. [24] apply a continuous relevance model (CRM) to the problem of directly retrieving the visual content of videos using text queries. The approach computes a joint probability model for image features and words using a training set of annotated images. This joint probability allows the computation of the conditional probability of words given image vector features. Once the annotation and feature components of the joint probability are modeled respectively by multinomial distribution and Gaussian kernels, images are ranked according to the conditional probability. Ghebreab et al. [25] conceive of a concept as an incremental and interactive formalization of the user's conception of an object in an image. They describe an object in terms of multiple-continuous boundary features and represent an object concept by the stochastic characteristics of an object population. The probability that a database object is an instance of a given object concept is computed on the basis of a Mahalanobis distance model. Objects that are an instance of the concept the user has in mind have high probability.

Several authors have used learned metrics to improve CBIR methods and classification algorithms that can be used for CBIR purposes. We will now review some relevant examples of this work. Peng et al. [4] use a binary classification to classify the database color image feature vectors as relevant or irrelevant. The classified feature vectors and the query image feature vectors constitute the training data from which relevance weights for different features are computed. The components of the weight vector represent the local relevance of each feature. They are adjusted to the location of the query image feature vector in the feature space. After the feature relevance has been determined, a weighted similarity metric is selected using reinforcement learning, which is based on classical logistic regression. Three different metrics are chosen: a weighted euclidean metric, a weighted city-block metric, and a weighted dominance metric. Aksoy et al. [7] use weighted L_1 and L_2 distances to measure the degree of similarity between two images, where the weights are the ratios of the standard deviations of the feature values both for the whole database and among the images selected as relevant. Each component of the weight vector represents the local relevance of a specific feature, and more importance is assigned to features that are relevant. Hastie and Tibshirani [29] propose an algorithm that starts with the euclidean distance and, for each test object, iteratively changes the weights of attributes. At each iteration, it selects a neighborhood of a test object and applies local discriminant analysis to shrink the distance in the direction parallel to the boundary between decision classes. Finally, it selects the k nearest neighbors according to the locally transformed metric. Domeniconi et al. [30] pursue the idea presented in [29] but use a support vector machine (SVM) instead of local discriminant analysis to determine class boundaries using margin maximization and to shrink the distance. Support vectors can be computed during the learning phase, which makes this approach much more efficient in comparison to local discriminant analysis. Chopra et al. [32] recently proposed a framework for similarity metric learning in which the metrics are parameterized by pairs of identical convolutional neural nets. Their cost function penalizes large distances between similarly labeled inputs and small distances between differently labeled inputs, with penalties that incorporate the idea of a margin.

Much work on metric learning has indeed focused on Mahalanobis distance learning. In these studies, the classification setting is based on a natural equivalence relation, namely, whether two points are in the same class or not. One classical statistical method that uses this Mahalanobis distance idea is Fisher's Linear Discriminant Analysis (LDA) (see, for example, [26]). Other recent methods seek to minimize various separation criteria between the classes by posing Mahalanobis distance learning as an optimization problem. One relevant example of these recent studies is that of Xing et al. [21], who use semidefinite programming to learn the Mahalanobis metric for clustering. Their algorithm aims to minimize the sum of squared distances between similarly labeled inputs while maintaining a lower bound on the sum of distances between differently labeled inputs. Goldberger et al. [27] propose the neighborhood component analysis (NCA), a novel algorithm for learning a Mahalanobis distance, designed to improve the KNN classification algorithm. The algorithm maximizes a nonconvex stochastic variant of the leave-one-out KNN score on the training set using gradient descent. It can also learn a low-dimensional linear embedding of labeled data that can be used for data visualization and fast classification. Other examples are those of Weinberger [28] and Globerson and Roweis [31], which pursue essentially the same goals as NCA but differ in their construction of convex objective functions.

Our retrieval approach consists of learning a weighted pseudometric using a Bayesian logistic regression model based on a variational method, and it has several advantages. First, the pseudometric is constructed in such a way that it can handle decomposed and compressed feature vectors via any kind of wavelet transform. Wavelet decomposition and compression allow a very good feature vector approximation with just few coefficients. This has the advantage of accelerating the search for a query feature vector and reducing storage for the featurebase. Second, the pseudometric is low rank as it considers only the resolution levels of the decomposed feature vectors instead of the totality of their coefficients, using a bucketing function. Therefore, the dimensionality of the transformed feature space is significantly reduced. Third, the adopted Bayesian logistic regression model is based on a variational method that allows the training to have low computational complexity while preserving a good classification performance. In our previous work [8], the pseudometric was learned using classical logistic regression. We will show that the Bayesian logistic regression model is a significantly better tool than the classical logistic regression model for learning the pseudometric and improving the classification performance and query results. The classification performance of both models is evaluated and compared for CBIR and other classification tasks in a decontextualized evaluation framework. In this same framework, we compare the Bayesian logistic regression model to some relevant state-of-the-art linear classification algorithms. Experiments show that the Bayesian logistic regression model outperforms these linear classification algorithms and is a significantly better tool than the classical logistic regression model for improving retrieval and classification performance. Finally, we perform a comparison with results for other retrieval methods.

In the next section, we briefly define the pseudometric and explain the fast feature vector querying algorithm. In Section 3, we explain the data training process and describe the adjustment of the pseudometric weights using the classical logistic regression model while showing its limitations and demonstrating that the Bayesian logistic regression model based on a variational method is more appropriate for the pseudometric weight computation. Then, we give a detailed description of the Bayesian logistic regression model based on a variational method and present the weight computation algorithm. The color image retrieval method is briefly presented in Section 4. In Section 5, a decontextualized evaluation is performed to compare the Bayesian logistic regression model with the classical version and some relevant state-of-the-art linear classification algorithms. Then, the feature vectors that we use to represent the database color images are summarized. Finally, a contextualized comparative evaluation of the Bayesian and classical logistic regression model s is performed for CBIR, and a comparison with results for different retrieval methods is provided.

2 The Pseudometric and the Fast Feature Vector Querying Algorithm

2.1 The Pseudometric

Let us consider Q and T as the query and the target feature vectors, respectively, with 2^J components each. The vectors Q and T are mapped from the feature space to a wavelet space using any kind of wavelet transform. Then, they are compressed to m coefficients each. Finally, each of their largest positive and negative wavelet coefficients are quantized to $+1$ and -1, respectively. The pseudometric is given by the following expression:

$$\|Q,T\| = \tilde{w}_0|\tilde{Q}[0]-\tilde{T}[0]| + \sum_{i:\tilde{Q}_q^c[i]\neq 0} w_{bin(i)}\left(\tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i]\right), \quad (1)$$

where

$$\left(\tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i]\right) = \begin{cases} 1 & \text{if } \tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i] \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$\tilde{Q}[0]$ and $\tilde{T}[0]$ are the scaling function coefficients; $\tilde{Q}_q^c[i]$ and $\tilde{T}_q^c[i]$ represent the ith coefficients of their wavelet decomposed versions, compressed and quantized; $\tilde{w}_0$ and the $w_{bin(i)}$ are the positive weights to compute; and the bucketing function $bin()$ groups these weights according to the J resolution levels such that

$$bin(i) = \lfloor log_2(i) \rfloor \quad \text{with} \quad i = 1, \dots, 2^J - 1. \quad (3)$$

To compute the pseudometric over a database of feature vectors, it is generally quicker to count the number of matching coefficients of $\tilde{Q}_q^c$ and $\tilde{T}_q^c$ than the mismatching coefficients. For this reason, we rewrite

$$\sum_{i:\tilde{Q}_q^c[i]\neq 0} w_{bin(i)}\left(\tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i]\right) = \sum_{i:\tilde{Q}_q^c[i]\neq 0} w_{bin(i)} - \sum_{i:\tilde{Q}_q^c[i]\neq 0} w_{bin(i)}\left(\tilde{Q}_q^c[i] = \tilde{T}_q^c[i]\right), \quad (4)$$

where

$$\left(\tilde{Q}_q^c[i] = \tilde{T}_q^c[i]\right) = \begin{cases} 1 & \text{if } \tilde{Q}_q^c[i] = \tilde{T}_q^c[i] \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Since the term $\sum_{i:\tilde{Q}_q^c[i]\neq 0} w_{bin(i)}$ is independent from the vectors $\tilde{T}_q^c$ and $\tilde{Q}_q^c$, we can discard it. Therefore, our pseudometric becomes

$$\|Q,T\| = \tilde{w}_0|\tilde{Q}[0]-\tilde{T}[0]| - \sum_{i:\tilde{Q}_q^c[i]\neq 0} w_{bin(i)}(\tilde{Q}_q^c[i] = \tilde{T}_q^c[i]). \quad (6)$$

2.2 The Fast Feature Vector Querying Algorithm

In order to optimize the metric computation process, we introduce two arrays called search arrays: Θ_+ for the coefficients quantized to $+1$ and Θ_- for those that are quantized to -1. Each array contains $2^J - 1$ elements, and each element contains a list. For example, the element $\Theta_+[i]$ points to the list of all database feature vectors with a large positive wavelet coefficient at the ith position after compression. In the same way, the element $\Theta_-[i]$ points to the list of all database feature vectors with a large negative wavelet coefficient at the ith position. Thanks to these arrays and the compression, during the querying process, we need only go through the m lists associated to the m coefficients retained for the query instead of $2^J - 1$ coefficients. Given the search arrays and the weights $\tilde{w}_0$ and $\{w_j\}_{j=0}^{J-1}$, the retrieval procedure for a query feature vector Q in the featurebase of feature vectors $T_k (k = 1, \dots, |DB|)$, where $|DB|$ denotes the featurebase size, is defined as follows:

```
Procedure Retrieval ($Q$: array $[1..2^J]$ of reals,
$m$: integer,$\Theta_-, \Theta_+$)
  $\tilde{Q} \leftarrow$ WaveletsDecomposition($Q$)
  Initialize $Score[k] = 0$, for each $k \in \{1, \dots, |DB|\}$
  For each $k \in \{1, \dots, |DB|\}$ do

      $Score$[position of $T_k$ in the (DB)] $= \tilde{w}_0 * |\tilde{Q}[0] - \tilde{T}_k[0]|$
  end for
  $\tilde{Q}^c \leftarrow$ Compress($\tilde{Q}, m$)
  $\tilde{Q}^c_q \leftarrow$ Quantify($\tilde{Q}^c$)
  For each $\tilde{Q}^c_q[i] \neq 0$ do
      If $\tilde{Q}^c_q[i] > 0$ then
          List $\leftarrow \Theta_+[i]$
      Else
          List $\leftarrow \Theta_-[i]$
      End if
      for each $l$ of List do
          $Score$[position of $l$ in the(DB)] = $Score$[position of $l$ in the (DB)] $- w_{bin(i)}$
   End for
  End for
  Return $Score$
End procedure
```

This procedure returns an array $Score$ such that $Score[k] = \|Q, T_k\|$ for each $k \in \{1, \ldots, |DB|\}$. The elements of $Score$, which are the degrees of similarity between the query Q and the feature vectors $T_k(k \in \{1, \ldots, |DB|\})$, can be negative or positive. The most negative similarity degree corresponds to the closest target to the query Q.

3 Pseudometric Weight Adjustment

The weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ are adjusted in such a way that the pseudometric should be effective enough to match similar feature vectors, as well as discriminate dissimilar ones. We define two classes, a relevance class Ω_0 and an irrelevance class Ω_1, in order to classify the feature vector pairs as similar or dissimilar. We suppose that Ω_0 contains n_0 explanatory vectors $\{X^r_i\}_{i=1}^{n_0}$ to represent the pairs of similar feature vectors, and Ω_1 contains n_1 explanatory vectors $\{X^{ir}_j\}_{j=1}^{n_1}$ to represent the pairs of dissimilar feature vectors. Given an explanatory vector $X^r_i = (\tilde{X}^r_{0,i}, X^r_{0,i}, \ldots, X^r_{J-1,i}, 1) \in \Omega_0$ representing a pair of similar feature vectors that are wavelet decomposed, compressed, and quantized, $\tilde{X}^r_{0,i}$ is the absolute value of the difference between their scaling factors, and $\{X^r_{k,i}\}_{k=0}^{J-1}$ are the numbers of mismatches between their J resolution level coefficients. The components of an explanatory vector $X^{ir}_j = (\tilde{X}^{ir}_{0,j}, X^{ir}_{0,j}, \ldots, X^{ir}_{J-1,j}, 1) \in \Omega_1$ are computed for a pair of dissimilar feature vectors in the same way. The basic aim of using the Bayesian and classical logistic regression model s is to allow a good separation between Ω_0 and Ω_1 by hyperplane and to compute the weights that represent the local relevances of the pseudometric components. The classes Ω_0 and Ω_1 are experimentally created in the data training phase explained below.

3.1 Data Training

Let us consider a color image database in which the images are clustered beforehand in a number of semantic clusters. Each cluster contains color images that are perceptually close to each other in terms of visual features such as color, texture, and shape. The purpose of weighting the pseudometric is to make it efficient enough to match images that belong to the same cluster and discriminate between images that belong to different clusters. For this reason, to create Ω_0, we draw all possible pairs of feature vectors representing color images belonging to the same cluster in the database, and for each pair, we compute an explanatory vector. Similarly, to create Ω_1, we draw all possible pairs of feature vectors representing color images belonging to different clusters in the database, and for each pair, we compute an explanatory vector.

3.2 The Classical Logistic Regression Model

In this model, each explanatory vector X^r_i of Ω_0 is associated with a binary target variable $S^r_i = 0$ for similarity, and each explanatory vector X^{ir}_j of Ω_1 is associated with a binary target variable $S^{ir}_j = 1$ for dissimilarity. Given two explanatory vectors X^r_i and X^{ir}_j, we model their associated binary target variables S^r_i and S^{ir}_j, respectively, by a relevance probability p^r_i and an irrelevance probability p^{ir}_j defined as follows:

$$p^{ir}_j = P\left(S^{ir}_j = 1|X^{ir}_j\right) = F\left(\tilde{w}_0\tilde{X}^{ir}_{0,j} + \sum_{k=0}^{J-1} w_k X^{ir}_{k,j} + v\right), \quad (7)$$

$$p^r_i = P\left(S^r_i = 0|X^r_i\right) = F\left(-\tilde{w}_0\tilde{X}^r_{0,i} - \sum_{k=0}^{J-1} w_k X^r_{k,i} - v\right), \quad (8)$$

where $F(x) = \frac{e^x}{1+e^x}$ is the logistic function, and v is an unknown intercept that will be computed with the pseudometric weights but will not be considered when using the pseudometric for feature vector comparison, as it is a constant for all query-target pairs and $F(-x)$ is a decreasing function. The weights and the intercept are determined using maximum likelihood estimation; that is, such that they optimize the probability of the actual configuration occurring. More precisely, if we look up the relevance and irrelevance class explanatory vectors and their associated binary variable values and use (7) and (8) to compute the probabilities p^{ir}_j and p^r_i, then the weights and the intercept are chosen to maximize the following conditional log likelihood:

$$\log\left(L(W = (\tilde{w}_0, w_0, \ldots, w_{J-1}, v))\right) = \sum_{i=1}^{n_0} log(p^r_i) + \sum_{j=1}^{n_1} log(p^{ir}_j). \quad (9)$$

The log-likelihood function is globally concave (there is only one solution, which is the maximum) [34]. Many numerical methods can be used to estimate the weights and the intercept. The methods most often used are the Gradient ascent and Fisher scoring algorithms [33]. The Fisher scoring method has the advantage of adding a direction matrix that assesses how quickly the log-likelihood function is changing [33]. This direction matrix is the Hessian matrix of the log-likelihood function. The Fisher scoring algorithm proceeds according to the equation

$$W_{new} = W_{old} - \alpha H^{-1}\frac{\partial log(L)}{\partial W},$$

where H and $\frac{\partial log(L)}{\partial W}$ are respectively the Hessian matrix and the gradient of $log(L)$, and α is a step-size parameter optimized via a line search to give the largest downhill step subject to $\tilde{w}_0 \geq 0$ and $w_j \geq 0 \forall j \in \{0, \ldots, J-1\}$ [40]. Once the Fisher scoring and line-search algorithms have been used to compute positive weights, an active set algorithm is applied to correct the false zero weight solutions [41], when converging with $\alpha = 0$. The inverse of the Hessian matrix approximates the variance-covariance matrix of the maximum log-likelihood estimators [35]. Therefore, the flatter the log-likelihood function, the smaller the Hessian matrix coefficients and the larger the variances of the estimators. This corresponds to the intuition that the flatter the log-likelihood function, the harder it will be to find the maximum of the function despite its concavity [33]. Also, when there are too many observations or explanatory vectors, the Fisher scoring algorithm has high computational complexity and takes a long time to converge; sometimes, it diverges because of the exponential term in the log-likelihood function [36]. Moreover, according to Weiss et al. [11], in the case where there are many zero explanatory vectors, maximum likelihood can fail and estimates of the parameters of interest (weights and intercept) may not exist or may be on the boundary of the parameter space. The most severe problems that can occur when fitting a logistic regression model are multicollinearity among the explanatory variables and cases where the data is completely or quasicompletely separable. Multicollinearity in the logistic regression model is a result of strong correlations between some or all of the explanatory variables. It generally occurs when the logistic regression model is large (contains many explanatory variables), and it greatly inflates the variances of the maximum log-likelihood estimators and can cause wrong signs and magnitudes of these estimators [37]. In the case of completely and quasicompletely separable data, the log-likelihood function is strictly monotonic, almost completely flat in the region of the parameter estimators, and reaches its maximum at infinity (maximum log likelihood does not exist) [38]. Since the classes Ω_0 and Ω_1 are intended to be large (training performed over all database images), high-dimensional (large J in case of feature vectors having a great number of components), and composed of real data, all of the problems mentioned above must be faced when fitting our classical logistic regression model. The problems related to the inflation and nonexistence of the log-likelihood estimators can be solved by regularizing the likelihood function by a prior distribution over the weights and intercept that smooths their estimates and reduces their space. The problems related to the high complexity caused by large and high-dimensional data sets can be solved by using variational transformations that simplify the computation of the weight and intercept estimates [9]. This motivates the adoption of a Bayesian logistic regression model based on a variational method.

3.3 The Bayesian Logistic Regression Model

In the Bayesian logistic regression framework, there are three main components: a chosen prior distribution over the parameters of interest, the likelihood function, and the posterior distribution. These three components are formally combined by Bayes' rule. The posterior distribution contains all the available knowledge about the parameters of interest in the model. In the literature, many priors with different distributional forms have been chosen for different applications based on the Bayesian logistic regression. Examples include the Dirichlet prior, Jeffrey's prior, and the Gaussian prior. The Dirichlet prior was chosen for the log-linear analysis of sparse frequency tables in [12]. In fact, in this application, the likelihood function is a multinomial density function that is a conjugate of the Dirichlet prior and, therefore, the posterior distribution has an analytically tractable Dirichlet form. This has the effect of smoothing the estimates to a specific model [10]. Jeffrey's prior is based on a structural rule and has a good theoretical justification [12]. However, in larger problems where the number of explanatory variables is large, it is difficult to apply because of its computational complexity. The Gaussian prior has become popular in logit modeling [12], [13], [11], [14], [15]. It has the advantage of having low computational complexity and of smoothing the estimates toward a fixed mean and away from unreasonable extremes. However, when the likelihood function is not a conjugate of the Gaussian prior, the posterior distribution has no tractable form, and its mode and mean computations are usually performed respectively by the modal a posteriori (MAP) approach and high-dimensional integration algorithms [12], which have a very high computational cost [9], [12], especially when the data set is large and high-dimensional, as in our case. To avoid this sizable computational cost, some authors have used Laplace approximation to approximate the posterior distributions with a tractable Gaussian form [11], [39]. However, Laplace approximation suffers from a lack of flexibility and is inaccurate [11]. According to [9], variational transformations have been shown to have much more flexibility, which translates into improved accuracy of the approximation. In this approach, variational transformations are used in order to approximate the likelihood function with a simpler tractable exponential form. In this case, thanks to the conjugacy, by combining a Gaussian prior distribution over the parameters of interest with the likelihood approximation, we obtain a closed Gaussian form approximation to the posterior distribution. However, as the number of observations is large, the number of variational parameters that must be updated to optimize the posterior distribution approximation is also large; hence, the computational cost is high. In the Bayesian logistic regression model that we propose, we use variational transformations [9] and Jensen's inequality in order to approximate the likelihood function with a tractable exponential form. The explanatory vectors are not observed but instead are distributed according to two specific distributions. This has the advantage of incorporating their prior knowledge in the weight computation. The posterior distribution is also accurately approximated with a Gaussian that depends only on two variational parameters. The computation of the mean of the posterior distribution approximation is fast and has low computational complexity. Let us denote the random vectors whose realizations represent the explanatory vectors $\{X_i^r\}_{i=1}^{n_0}$ of the relevance class Ω_0 and the explanatory vectors $\{X_j^{ir}\}_{j=1}^{n_1}$ of the irrelevance class Ω_1 by $\underline{\mathrm{X}}_0 = (\tilde{\underline{\mathrm{X}}}_{0,0}, \underline{\mathrm{X}}_{0,0}, \ldots, \underline{\mathrm{X}}_{J-1,0}, 1)$ and $\underline{\mathrm{X}}_1 = (\tilde{\underline{\mathrm{X}}}_{0,1}, \underline{\mathrm{X}}_{0,1}, \ldots, \underline{\mathrm{X}}_{J-1,1}, 1)$, respectively. We suppose that $\underline{\mathrm{X}}_0 \sim q_0(\underline{\mathrm{X}}_0)$ and $\underline{\mathrm{X}}_1 \sim q_1(\underline{\mathrm{X}}_1)$, where q_0 and q_1 are two chosen distributions. With $\underline{\mathrm{X}}_0$, we associate a binary random variable $\underline{\mathrm{S}}_0$ whose realizations are the target variables $\{S_i^r = 0\}_{i=1}^{n_0}$, and with $\underline{\mathrm{X}}_1$, we associate a binary random variable $\underline{\mathrm{S}}_1$ whose realizations are the target variables $\{S_j^{ir} = 1\}_{j=1}^{n_1}$. We set $\underline{\mathrm{S}}_0$ equal to 0 for similarity, and we set $\underline{\mathrm{S}}_1$ equal to 1 for dissimilarity. The parameters of interest (weights and intercept) are considered as random variables and are denoted by the random vector $\underline{\mathrm{W}} = (\underline{\tilde{w}}_0, \underline{w}_0, \ldots, \underline{w}_{J-1}, \underline{v})$. We assume that $\underline{\mathrm{W}} \sim \pi(\underline{\mathrm{W}})$, where π is a Gaussian prior with prior mean μ and prior covariance matrix Σ. Using Bayes' rule, the posterior distribution over $\underline{\mathrm{W}}$ is given by

$$P(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1) = \frac{P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1|\underline{\mathrm{W}})\pi(\underline{\mathrm{W}})}{P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1)}, \tag{10}$$

where

$$\begin{aligned} P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1|\underline{\mathrm{W}}) &= \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{W}}) \\ &= \sum_{x_0\in\Omega_0, x_1\in\Omega_1} \frac{\prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i, \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}})}{(\pi(\underline{\mathrm{W}}))^2} \\ &= \sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} \left[\frac{P(\underline{\mathrm{S}}_i = i, \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}})}{P(\underline{\mathrm{W}}, \underline{\mathrm{X}}_i = x_i)\pi(\underline{\mathrm{W}})}\right] P(\underline{\mathrm{W}}, \underline{\mathrm{X}}_i = x_i). \end{aligned}$$

Since in the Bayesian approach we generally suppose that the space of unknown parameters is independent from the space of observations, we assume that $\underline{\mathrm{W}}$ and $\underline{\mathrm{X}}_i$ are independent for each $i \in \{0,1\}$ and, thus, the joint probability $P(\underline{\mathrm{W}}, \underline{\mathrm{X}}_i = x_i) = \pi(\underline{\mathrm{W}})q_i(\underline{\mathrm{X}}_i = x_i)$ for each $i \in \{0,1\}$. Therefore, we obtain

$$\begin{aligned} &P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1|\underline{\mathrm{W}}) \\ &\quad= \sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} \left[\frac{P(\underline{\mathrm{S}}_i = i, \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}})}{(\pi(\underline{\mathrm{W}}))^2 q_i(\underline{\mathrm{X}}_i = x_i)}\right] \pi(\underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i) \\ &\quad= \sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i), \end{aligned}$$

where $P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) = F((2i-1)\underline{\mathrm{W}}^t x_i)$ for each $i \in \{0,1\}$ represent logistic modelings of $\underline{\mathrm{S}}_0$ and $\underline{\mathrm{S}}_1$ given the realizations of $\underline{\mathrm{X}}_0$ and $\underline{\mathrm{X}}_1$, respectively. Therefore,

$$\begin{aligned} &P(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1) \\ &\quad= \frac{\left[\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\right]\pi(\underline{\mathrm{W}})}{P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1)}, \end{aligned} \tag{11}$$

where

$$\begin{aligned} &P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1) = \\ &\int \left[\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\right] \pi(\underline{\mathrm{W}}) d\underline{\mathrm{W}}. \end{aligned} \tag{12}$$

The computation of the posterior distribution $P(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1)$ is intractable. However, we can approximate it by a variational posterior approximation with a Gaussian form, whose mean and covariance matrix computation are feasible. To obtain this variational posterior approximation, we perform two successive approximations to the posterior distribution nominator term $\left[\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i| \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\right]$ in (11) in order to bound it by an exponential form that is a conjugate of the Gaussian prior $\pi(\underline{\mathrm{W}})$.

First approximation. This first approximation is based on a variational transformation of the sigmoid function $F(x)$ of the logistic regression. According to [9], the variational approximation of the sigmoid function in $H_i = (2i-1)\underline{\mathrm{W}}^t x_i \forall i \in \{0,1\}$ is given by

$$\begin{aligned} &P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) = F(H_i) \\ &\qquad \geq F(\epsilon_i) e^{\left[\frac{(H_i - \epsilon_i)}{2} - \varphi(\epsilon_i)\left(H_i^2 - \epsilon_i^2\right)\right]} = P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}, \epsilon_i), \end{aligned} \tag{13}$$

where $\epsilon_i > 0$ is the variational parameter, $\varphi(\epsilon_i) = \frac{tanh(\frac{\epsilon_i}{2})}{4\epsilon_i}$, and

$$\tanh\left(\frac{\epsilon_i}{2}\right) = \frac{e^{\frac{\epsilon_i}{2}} - e^{\frac{-\epsilon_i}{2}}}{e^{\frac{\epsilon_i}{2}} + e^{\frac{-\epsilon_i}{2}}}.$$

Therefore, the posterior distribution nominator in (11) can be approximated as follows:

$$\begin{aligned} &\left[\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\right] \pi(\underline{\mathrm{W}}) \\ &\geq \left[\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}, \epsilon_i) q_i(\underline{\mathrm{X}}_i = x_i)\right] \pi(\underline{\mathrm{W}}). \end{aligned} \tag{14}$$

Second approximation. The first approximation is insufficient to approximate the term $\left[\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\right]$ by an exponential form. We therefore perform a second approximation, based on Jensen's inequality, which uses the convexity of the function e^x. Using Jensen's inequality, we obtain

$$\begin{aligned} &\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \left[\prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}, \epsilon_i) q_i(\underline{\mathrm{X}}_i = x_i)\right] \\ &= \left[\prod_{i=0}^{1} F(\epsilon_i)\right] \sum_{x_0\in\Omega_0, x_1\in\Omega_1} \left[e^{\left[\sum_{i=0}^{1}\left[\frac{(H_i - \epsilon_i)}{2} - \varphi(\epsilon_i)\left(H_i^2 - \epsilon_i^2\right)\right]\right]} \right. \\ &\quad \left. \prod_{i=0}^{1} q_i(\underline{\mathrm{X}}_i = x_i)\right] \\ &\geq \left[\prod_{i=0}^{1} F(\epsilon_i)\right] e^{\left[\sum_{i=0}^{1}\left[\frac{E_{q_i}[H_i] - \epsilon_i}{2}\right] - \sum_{i=0}^{1}\left[\varphi(\epsilon_i)\left(E_{q_i}[H_i^2] - \epsilon_i^2\right)\right]\right]} \\ &= \underline{\mathrm{P}}\left(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}\right), \end{aligned}$$

where E_{q_0} and E_{q_1} are the expectations with respect to the distributions q_0 and q_1, respectively.

Finally, thanks to the two above approximations, the posterior distribution numerator in (11) can be approximated as follows:

$$\begin{aligned} &\left[\sum_{x_0\in\Omega_0, x_1\in\Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\right] \pi(\underline{\mathrm{W}}) \\ &\quad \geq \underline{\mathrm{P}}\left(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}\right) \pi(\underline{\mathrm{W}}). \end{aligned}$$

Thus, the variational posterior approximation is given by

$$\begin{aligned} &P\left(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}\right) \\ &\qquad = \frac{\underline{\mathrm{P}}\left(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}\right) \pi(\underline{\mathrm{W}})}{P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1)}. \end{aligned}$$

Since $P(\underline{S}_0 = 0, \underline{S}_1 = 1)$ is a constant that does not affect the form of the variational posterior approximation, we can ignore it. We thus obtain

$$P\big(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1\big) \propto \underline{P}\big(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1\big)\pi(\underline{W}).$$

Finally, the posterior distribution is approximated as follows:

$$\begin{aligned} &P(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1) \\ &\geq \ P\big(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1\big), \end{aligned} \qquad (15)$$

$$\propto \ \underline{P}\big(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1\big)\pi(\underline{W}). \quad (16)$$

Since $\pi(\underline{W})$ is a Gaussian that is a conjugate of $\underline{P}\big(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1\big)$, which has an exponential form, the variational posterior approximation is a Gaussian with a posterior mean μ_{post} and a posterior covariance matrix Σ_{post}. Substituting $\pi(\underline{W})$ and $P\big(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1\big)$ by their Gaussian forms in (16), we obtain

$$\begin{aligned} e^{-\frac{1}{2}(\underline{W}-\mu_{post})^t\Sigma_{post}^{-1}(\underline{W}-\mu_{post})} &\propto \underline{P}(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 \\ &= 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1)e^{-\frac{1}{2}(\underline{W}-\mu)^t\Sigma^{-1}(\underline{W}-\mu)}. \end{aligned}$$

Thus, omitting the algebra, Σ_{post} and μ_{post} are given by the following Bayesian update equations:

$$(\Sigma_{post})^{-1} \ = \ (\Sigma)^{-1} + 2\sum_{i=0}^{1}\Big[\varphi(\epsilon_i)E_{q_i}[x_i(x_i)^t]\Big], \qquad (17)$$

$$\mu_{post} \ = \ \Sigma_{post}\left[(\Sigma)^{-1}\mu + \sum_{i=0}^{1}\left[\Big(i - \frac{1}{2}\Big)E_{q_i}[x_i]\right]\right]. \qquad (18)$$

According to (17), Σ_{post} depends on the variational parameters $\{\epsilon_i\}_{i=0}^1$, so we must specify these. We have to find the values of $\{\epsilon_i\}_{i=0}^1$ that yield a tight lower bound in (15) and, then, an optimal approximation to the posterior distribution. This can be done by an expectation-maximization (EM) algorithm, which is derived in Appendix A which can be found at http://computer.org/tpami/archives.htm. The variational parameters are given by

$$\begin{aligned} \epsilon_i^2 \ &= E_{P\left(\underline{W}|\underline{S}_0=0,\underline{S}_1=1,\left(\{\epsilon_i\}_{i=0}^1\right)^{old},\{q_i\}_{i=0}^1\right)}\left[E_{q_i}[(\underline{W}^t x_i)^2]\right] \\ &= E_{q_i}[(x_i)^t\Sigma_{post}x_i] + (\mu_{post})^t\Big[E_{q_i}[x_i(x_i)^t]\Big]\mu_{post}, \forall i \in \{0,1\}, \end{aligned} \qquad (19)$$

where $P\big(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \big(\{\epsilon_i\}_{i=0}^1\big)^{old}, \{q_i\}_{i=0}^1\big)$ is the variational posterior approximation based on the previous values of $\{\epsilon_i\}_{i=0}^1$. The weight and intercept computation algorithm has two phases. The first phase is the initialization; the second is iterative and allows the computation of Σ_{post} and μ_{post} through the Bayesian update equations (17) and (18), respectively, while using (19) to find the variational parameters at each iteration. In the second phase, we use a line-search algorithm to optimize a step-size parameter θ to give the largest downhill step, subject to $\mu_{post,i} \geq 0 \forall i \in \{0,\ldots,J\}$. The values of the μ_{post} components are the desired estimates of the pseudometric weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ and the intercept v.

```
Initialization.
1. Compute the parameters of the distributions $q_0$ and $q_1$ that model the relevance class $\Omega_0$ and irrelevance class $\Omega_1$ explanatory vectors, respectively.
2. Initialize the covariance matrix $\Sigma^{old}$ to the identity matrix and the mean $\mu^{old}$ to a vector with components equal to 1.
3. Initialize the variational parameters as follows:
   For each $i \in \{0,1\}$ do
      $(\epsilon_i^{old})^2 \leftarrow E_{q_i}[(x_i)^t\Sigma^{old}x_i] + (\mu^{old})^t\Big[E_{q_i}[x_i(x_i)^t]\Big]\mu^{old}$
   End for

Computation of $\Sigma_{post}$ and $\mu_{post}$:
1. Do
      $(\Sigma_{post}^{try})^{-1} \leftarrow (\Sigma^{old})^{-1} + 2\sum_{i=0}^{1}\big[\varphi(\epsilon_i^{old})E_{q_i}[x_i(x_i)^t]\big]$
      $\mu_{post}^{try} \leftarrow \Sigma_{post}^{try}\Big[(\Sigma^{old})^{-1}\mu^{old} + \sum_{i=0}^{1}\big[(i - \frac{1}{2})E_{q_i}[x_i]\big]\Big]$
      For each $i \in \{0,1\}$ do
         $(\epsilon_i^{try})^2 \leftarrow E_{q_i}[(x_i)^t\Sigma_{post}^{try}x_i] + (\mu_{post}^{try})^t\Big[E_{q_i}[x_i(x_i)^t]\Big]\mu_{post}^{try}$
      End for
      $\theta \leftarrow \min_{j\in\{0,\ldots,J\}}\left\{\frac{-\mu_{post,j}^{old}}{(\mu_{post,j}^{try} - \mu_{post,j}^{old})}, 1/(\mu_{post,j}^{try} - \mu_{post,j}^{old}) < 0\right\}$
      $\begin{pmatrix}\mu_{post}^{new}\\ \Sigma_{post}^{new}\\ \epsilon_0^{new}\\ \epsilon_1^{new}\end{pmatrix} \leftarrow \begin{pmatrix}\mu_{post}^{old}\\ \Sigma_{post}^{old}\\ \epsilon_0^{old}\\ \epsilon_1^{old}\end{pmatrix} + \theta\left[\begin{pmatrix}\mu_{post}^{try}\\ \Sigma_{post}^{try}\\ \epsilon_0^{try}\\ \epsilon_1^{try}\end{pmatrix} - \begin{pmatrix}\mu_{post}^{old}\\ \Sigma_{post}^{old}\\ \epsilon_0^{old}\\ \epsilon_1^{old}\end{pmatrix}\right]$
   While ($|\Sigma_{post}^{old} - \Sigma_{post}^{new}| >$ threshold or $|\mu_{post}^{old} - \mu_{post}^{new}| >$ threshold)
   Return $\mu_{post}^{new}$
2. Apply an active set algorithm to correct the false zero solutions of $\mu_{post,i}(i \in \{0,\ldots,J\})$ [41], when exiting the iterative phase with $\theta = 0$.
3. Assign the $\mu_{post}$ component values to the pseudometric weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ and the intercept $v$.
```

The iterative phase of the above algorithm scales with the dimension of $\underline{W}$. In fact, it is dominated by the inversion of the variance-covariance matrix, which requires $\mathcal{O}((J+2)^3)$ operations at each iteration.

4 Color Image Retrieval Method

The querying method has two phases. The first is a preprocessing phase, executed once for the entire database containing $|DB|$ color images. The second is the querying phase.

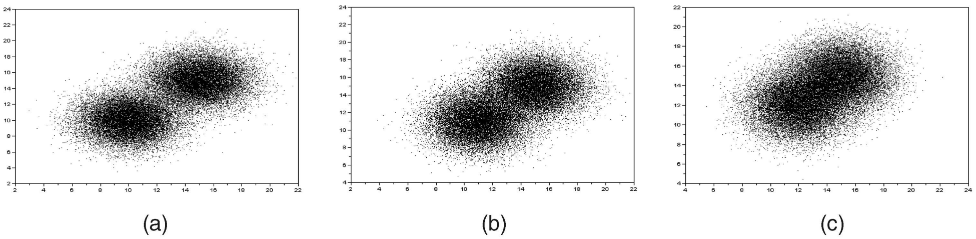


Fig. 1. Synthetic data: (a) slightly overlapped clusters $(\mathrm{OLR} = 0.1)$, (b) overlapped clusters $(\mathrm{OLR} = 0.3)$, and (c) highly overlapped clusters $(\mathrm{OLR} = 0.5)$.

4.1 Color Image Database Preprocessing

In the general case, the preprocessing phase (executed once for all the database color images before the querying phase) can be broken down into the following steps:

1. Choose N feature vectors for comparison.
2. Compute the N feature vectors $T_{li}(l \in \{1, \dots, N\})$ for each $i\mathrm{th}$ color image of the database, where $i \in \{1, \dots, |DB|\}$.
3. Wavelet decompose, compress to m coefficients each, and quantize the feature vectors representing the database color images.
4. Organize the decomposed, compressed, and quantized feature vectors into search arrays Θ^l_+ and $\Theta^l_-(l = 1, \dots, N)$.
5. Adjust the metric weights $\tilde{w}^l_0$ and $\{w^l_k\}^{J-1}_{k=0}$ for each featurebase $T_{li}(i = 1, \dots, |DB|)$ representing the database color images, where $l \in \{1, \dots, N\}$.

4.2 The Querying Algorithm

We describe the querying algorithm in the general case by the following steps:

1. Given a query color image, denote the feature vectors representing the query image by $Q_l(l = 1, \dots, N)$.
2. Wavelet decompose the feature vectors representing the query image, compress them to m coefficients each, and quantize them.
3. Represent the degrees of similarity between $Q_l(l = 1, \dots, N)$ and the database color image feature vectors $T_{li}(l = 1, \dots, N)(i = 1, \dots, |DB|)$ by the arrays $Score_l$ $(l = 1, \dots, N)$ such that $Score_l[i] = \|Q_l, T_{li}\|$ for each $i \in \{1, \dots, |DB|\}$. These arrays are returned by the procedures $\mathrm{Retrieval}(Q_l, m, \Theta^l_+, \Theta^l_-)(l = 1, \dots, N)$, respectively.
4. Represent the degrees of similarity between the query color image and the database color images by a resultant array $TotalScore$ such that $TotalScore[i] = \sum^N_{l=1} \gamma_l Score_l[i]$ for each $i \in \{1, \dots, |DB|\}$, where $\{\gamma_l\}^N_{l=1}$ are weight factors used to fine-tune the influence of each individual feature.
5. Organize the database color images in order of increasing resultant similarity degrees in the array $TotalScore$. The most negative resultant similarity degrees correspond to the closest target images to the query image. Finally, return to the RI target color images closest to the query color image, where RI is the number of images returned, chosen by the user.

5 Experimental Results

In this section, we will present a decontextualized comparison of the Bayesian logisti regression model (BLRM) to the Classical logistic regression model (CLRM) and some relevant state-of-the-art linear classification algorithms that learn classifiers that are constructed as weighted linear combinations of features. Then, we will perform an evaluation and comparison of the BLRM and the CLRM in the image retrieval context.

5.1 Decontextualized Evaluation and Comparison

In this section, we use synthetic data and a collection of benchmark real data sets to evaluate the BLRM and to compare it to the CLRM, the Support Vector Machine (SVM) [42], the Relevance Vector Machine (RVM) [43], and the Informative Vector Machine (IVM) [44] in terms of classification performance and training runtime. Since the aim of the decontextualized evaluation is to tease out the performance of the BLRM in a general context, the chosen data sets are not related to wavelet representation. The classification performance evaluations and comparisons are performed on the synthetic data using the following error measures: classifier error, bias, and variance, proposed and described in [46]. For the real data, these evaluations and comparisons are performed using the following error measures: classification accuracy [18] and the B index measure of predictive accuracy (its values are on the interval [0, 1], where 1 indicates perfect prediction) [47]. Because we are especially interested in two-class linear classification problems with large numbers of features or training samples, the synthetic and real data sets were selected to vary widely in training set size and the number of features. We implemented our own C++ code for the BLRM, CLRM, RVM (based on the block-wise algorithm in [43]), and IVM, but for the SVM, we adopted the widely used SVM-light program, which uses a highly optimized C code [45]. The synthetic data is a collection of three 10-dimensional $(M = 10)$ data sets. Each set has two clusters with a total of $N = 40,000$ points and is generated from two Gaussians, 20,000 points per Gaussian. The two clusters in the first set are slightly overlapped, those in the second set are overlapped, and those in the third set are highly overlapped. The overlap between two clusters is measured using the overlap rate (OLR) (it lies between 0 and 1, where 1 indicates perfect overlap) [48] and controlled by moving a cluster toward the other after translating its mean. For ease of representation, the synthetic data is reduced from its original dimensionality to two dimensions and then shown in Fig. 1. Table 1 describes the eight real data sets chosen.

Image, *Waveform*, *German*, and *Breast Cancer* were extracted from the famous University of California, Irvine (UCI) collection. More details concerning the original provenance

TABLE 1
Description of the Real Data Sets

Data set name	Number of training samples N	Number of test samples	Number of classes	Number of features (dimension M)
Image	1300	1010	2	18
Waveform	400	4600	2	21
German	700	300	2	20
Breast Cancer	200	77	2	9
$0-6$ (*MNIST*)	11841	1938	2	256
$7-9$ (*MNIST*)	12214	2037	2	256
$d-t$ (*TIMIT*)	6380	300	2	118
$iy-ih$ (*TIMIT*)	8874	446	2	118

TABLE 2
Evaluation and Comparison on the Synthetic Data

	BLRM	SVM	RVM	IVM	CLRM
Slightly overlapped clusters	1.7/7.5/19.5/38	1.8/7.6/19.7/1510	2.1/7.8/20.2/380	2.3/8.3/20.9/710	3.1/10.2/23.6/340
Overlapped clusters	3.2/10.1/33.8/42	3.3/10.3/34.1/1480	3.5/10.7/34.7/362	3.7/10.8/34.9/740	4.9/12.5/37.9/310
Highly overlapped clusters	5.3/12.8/56.2/35	5.4/13.1/56.6/1540	6.3/13.2/57.6/325	6.5/13.6/58.2/680	8.1/16.2/62.4/285

The four entries (left to right) are the training bias (percent), variance (percent), classifier error (percent), and training runtime (seconds). Note that the classifier $\mathrm{error} = \mathrm{bias} + \mathrm{variance} + \mathrm{Bayes}$ *error, where the Bayes error is the misclassification rate [46].*

TABLE 3
Evaluation and Comparison on the Eight Real Data Sets

	BLRM	SVM	RVM	IVM	CLRM
Image	93.7/0.95/51	93.2/0.94/180	91.2/0.92/82	90.8/0.9/106	82.2/0.88/91
Waveform	80.2/0.87/57	79.9/0.85/110	78.5/0.82/65	76.8/0.79/85	69.5/0.75/76
German	70.2/0.72/54	69.4/0.68/130	69.8/0.71/63	69.7/0.69/81	58.5/0.61/69
Breast Cancer	64.8/0.66/39	64/0.65/83	62.3/0.63/50	63.2/0.61/62	54.7/0.51/55
$0-6$ (*MNIST*)	96.3/0.98/212	96/0.97/1080	93.7/0.95/350	92/0.94/510	75.3/0.72/323
$7-9$ (*MNIST*)	95.2/0.96/267	95/0.95/1123	94.3/0.93/389	94.6/0.94/540	74.2/0.68/410
$d-t$ (*TIMIT*)	77.5/0.76/142	76.9/0.74/830	75.7/0.73/280	74/0.71/523	51/0.48/250
$iy-ih$ (*TIMIT*)	91.2/0.89/125	90/0.88/910	87/0.85/310	88.8/0.86/415	65/0.62/273

The three entries (left to right) are the test classification accuracy (percent), test B index measure ($\in [0,1]$), and training runtime (seconds).

of these data sets are available in a highly comprehensive online repository [49]. The 0-6 and 7-9 data sets were extracted from the $MNIST$ data set of handwritten digits, and the $d-t$ and $iy-ih$ data sets were extracted from the $TIMIT$ speech data set. More details concerning the original provenance of these data sets are available in [50]. For each synthetic and real data set, we did 10 independent SVM runs with regularization parameters $C \in \{1, \ldots, 10\}$, respectively; then, we initialized the active set size for IVM with the average number (over 10 runs) of support vectors [42]. When fitting the BLRM and the CLRM on the synthetic and real data sets, we did not use the line-search algorithm to restrict the weights to be positive. Moreover, for each data set of the synthetic data, the distributions q_0 and q_1 of the BLRM were chosen as 10-dimensional Gaussians whose parameters were computed directly from the cluster points, and for each real data set, q_0 and q_1 were chosen as empirical distributions, since we do not have prior knowledge about the classes of the real data set. Table 2 illustrates the computed training classifier errors, biases, and variances and the training runtimes for the various classifiers on the synthetic data sets. Table 3 illustrates the computed test classification accuracies and test B index measures and the training runtimes for the various classifiers on the real data sets.

In terms of classification performance and training runtime, we can see that the BLRM outperforms the other four classifiers on the synthetic data sets and on all eight real data sets. In Tables 2 and 3, it can be seen that the BLRM slightly outperforms the SVM while achieving a significantly lower training runtime. Generally speaking, in terms of classification performance (except for *German*), there is a greater difference between the BLRM and the RVM and IVM than there is between these two and the SVM, but in terms of training runtime, the RVM and IVM are closer to the BLRM than to the SVM. We also notice that the BLRM significantly outperforms the CLRM in terms of classification performance and training runtime, especially on the largest and most high-dimensional data sets 0-6, 7-9, $d-t$, and $iy-ih$. In fact, the high dimensionality makes the CLRM suffer from various side effects of multicollinearity, which strongly affect the precision of the maximum likelihood estimates. The BLRM outperforms the other classifiers in terms of classification performance thanks to its incorporation of the prior knowledge of the data set clusters and its robust approximation of the posterior distribution. The variational approximation adopted here was shown to be more flexible and accurate than the Laplace quadratic approximation adopted in the RVM [43] and the approximate method adopted in the IVM [44]. In terms of time complexity, the BLRM training time scales with only $\mathcal{O}(M^3)$ (dominated by the inversion of the posterior covariance matrix), whereas for SVM and IVM, it scales with $\mathcal{O}(N^2)$ and $\mathcal{O}(NN_s^2)$ [50], respectively, where N_s is the number of support vectors. As for the RVM and CLRM, their training times both scale with $\mathcal{O}(M^3) + \mathcal{O}(NM^2)$, where $\mathcal{O}(M^3)$ is the complexity of the Hessian inversion for the CLRM and the inversion of the posterior covariance matrix for the RVM, apart from the computations of these matrices that require $\mathcal{O}(NM^2)$ each. We can notice that as $N >> M$ and $N_s >> M$, which is the case of the used data sets, the BLRM has much lower computational complexity than the other classifiers. Note that the computed training runtimes given in Tables 2 and 3 above are also dominated by the number of iterations of each classifier. We noticed in our experiments that for all data sets, the BLRM requires fewer iterations to converge than do the other classifiers. This is thanks to the simple EM algorithm adopted, which iterates over only two variational parameters.

5.2 Contextualized Evaluation and Comparison

In this section, we briefly present the feature vectors used for color image representation. Then, we discuss the choices of the distributions q_0 and q_1 in order to validate the BLRM in the image retrieval context. Finally, we evaluate the querying method using the CLRM and BLRM separately, and we perform a comparison with results for different retrieval methods. The choices of the distributions q_0 and q_1 and the querying evaluation were conducted on the WANG, Zurich Building Image Database (ZuBuD), University of Washington (UW), and California Institute of Technology (Caltech) color image databases proposed in [17]. The WANG database contains $|DB| = 1,000$ color images that were selected manually to form 10 sets (for example, Africa, beach, ruins, and food) of 100 images each. ZuBuD contains $|DB| = 1,005$ color images of buildings selected to form 201 image classes, where each class contains five color images of the same building taken from different positions. The UW database contains $|DB| = 1,109$ color images. No class information is available for the images, but they are annotated. We clustered the images in different classes according to their annotations (for example, barcelona, springflowers, and swissmountains): that is, two images belong to the same class iff their annotations contain identical words. The Caltech database contains $|DB| = 2,000$ color images that we selected from the Caltech collection categories (for example, motorbikes, airplanes, and faces) to form 100 classes of 20 images each. Before feature vector extraction, we represented the WANG, ZuBuD, UW, and Caltech database color images in the perceptually uniform LAB color space.

5.2.1 Feature Vectors Used

The luminance histogram and the weighted histograms described in detail in [8] are used for image color and contrast description in this paper; image texture description is performed using kurtosis and skewness histograms [19]. Given an $M \times N$ pixel LAB color image, its luminance histogram is denoted by h_L and plots the number of pixels of luminance L. The weighted histograms are the color histogram constructed after edge region elimination and the multispectral gradient module mean histogram. The former is denoted by h_k^h, and the latter is denoted by $\bar{h}_k^e (k = a, b)$, where a and b are the chrominances red/green and yellow/blue, respectively. The LAB color image kurtosis and skewness histograms are given by

$$h_k^\kappa(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k^\kappa(i,j) - c) \tag{20}$$

and

$$h_k^s(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k^s(i,j) - c), \tag{21}$$

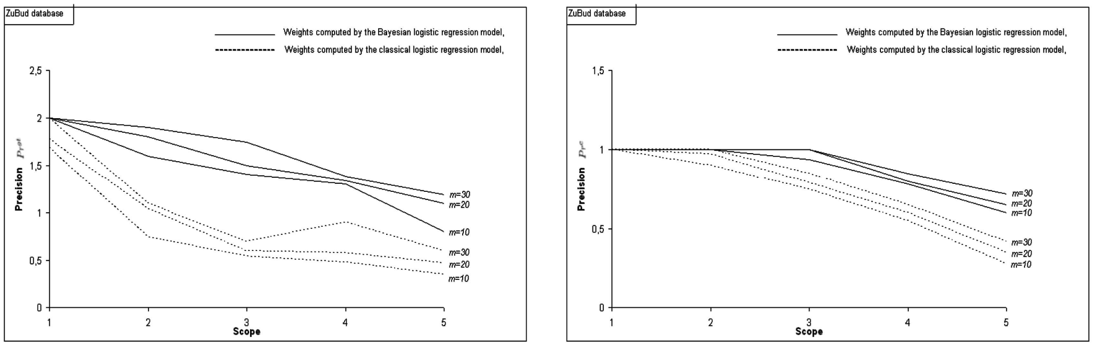
respectively, for each $c \in \{0, \ldots, 255\}$ and $k = L, a, b$, where I_L^κ, I_a^κ, and I_b^κ are the kurtosis images of the luminance L and the chrominances a and b, respectively, and I_L^s, I_a^s, and I_b^s are their skewness images. They are obtained by a local computation of the kurtosis and skewness values at the luminance and chrominance image pixels. Thus, each color image of the WANG, ZuBuD, UW, and Caltech databases is represented by $N = 11$ feature vectors, which are the histograms h_L, h_a^h, h_b^h, $\bar{h}_a^e$, $\bar{h}_b^e$, h_L^κ, h_a^κ, h_b^κ, h_L^s, h_a^s, and h_b^s. Then, all of these histograms are Daubechies-8 wavelet decomposed, compressed to m coefficients each, and quantized. Therefore, each database is represented by eleven featurebases of transformed histograms. We chose Daubechies-8 wavelets as they have been proven to have good frequency properties and to be good for one-dimensional signal synthesis. Moreover, they are a good compromise between computational time and performance [51]. Since we discretized each histogram extracted into 256 components, we set J equal to 8 in the following sections.

5.2.2 The Choice of q_0 and q_1

The choice of q_0 and q_1 are performed separately for each of the featurebases representing the WANG, ZuBuD, UW, and Caltech databases. For simplicity, we assume that $\underline{\tilde{\mathrm{X}}}_{0,0}$ and $(\underline{\mathrm{X}}_{0,0}, \ldots, \underline{\mathrm{X}}_{J-1,0})$ are independent. Analogously, for the same reason, we made the same assumption for $\underline{\tilde{\mathrm{X}}}_{0,1}$ and $(\underline{\mathrm{X}}_{0,1}, \ldots, \underline{\mathrm{X}}_{J-1,1})$. For each histogram featurebase, we suppose that the random vector $(\underline{\mathrm{X}}_{0,0}, \ldots, \underline{\mathrm{X}}_{J-1,0})$ random variables whose realizations are positive integers are independent, and each one of them follows a Poisson distribution. Analogously, we made the same choice for $(\underline{\mathrm{X}}_{0,1}, \ldots, \underline{\mathrm{X}}_{J-1,1})$. We are aware that these modelings are approximations, especially when the realizations of the random variables are very small integers, but we claim that they have very negligible effect on the querying results. For each histogram featurebase, the realizations of the random variable $\underline{\tilde{\mathrm{X}}}_{0,0}$ are positive real numbers. We modeled them by a Gaussian mixture distribution whose parameters were estimated by the EM algorithm and whose component number was selected using the minimum message length (MML) validity function, as it has been shown to give good results in [16]. Similarly, the realizations $\underline{\tilde{\mathrm{X}}}_{0,1}$ were modeled by a Gaussian mixture distribution. Note that we also chose the distributions q_0 and q_1 as empirical distributions to validate the BLRM, but in adopting this choice, we noticed that the querying results differ slightly from the ones found after choosing q_0 and q_1 as joint distributions of Gaussian mixtures and Poisson distributions. Moreover, this latter choice remains better as the Poisson distribution parameters are obtained by computing simple arithmetic means of the integer realizations, whereas since q_0 and q_1 are empirical distributions, a higher computational complexity is involved in precomputing the expectations in the equations of the variational parameter initialization and Bayesian update before the iterative phase of the BLRM.

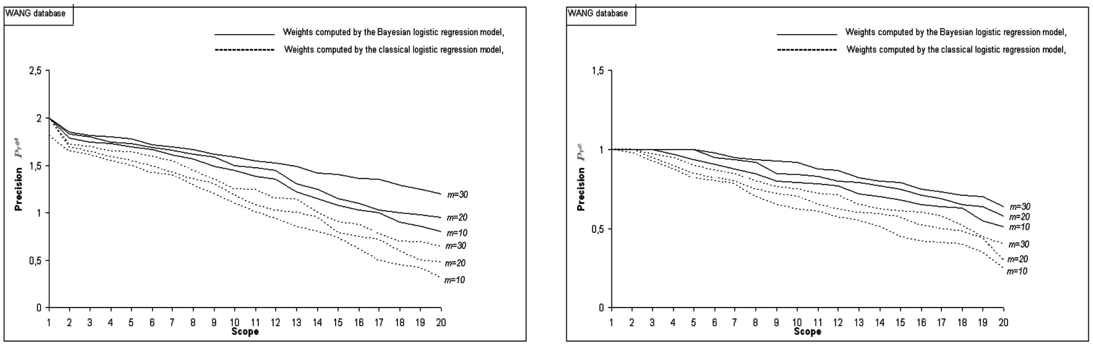
5.2.3 Comparative Evaluation of the Querying Procedure

In order to evaluate our querying method, two principal issues are required: A ground truth and an objective performance evaluation of the adopted classification method. These two issues are represented by precision-scope curves $Pr = f(RI)$ [20], where the scope RI is the number of images



(a) (b)

Fig. 2. Evaluation (ZuBuD database): (a) ground-truth precision-scope curves and (b) classification precision-scope curves for retrieval using weights computed by the CLRM and weights computed by the BLRM for the compression orders $m \in \{30, 20, 10\}$.

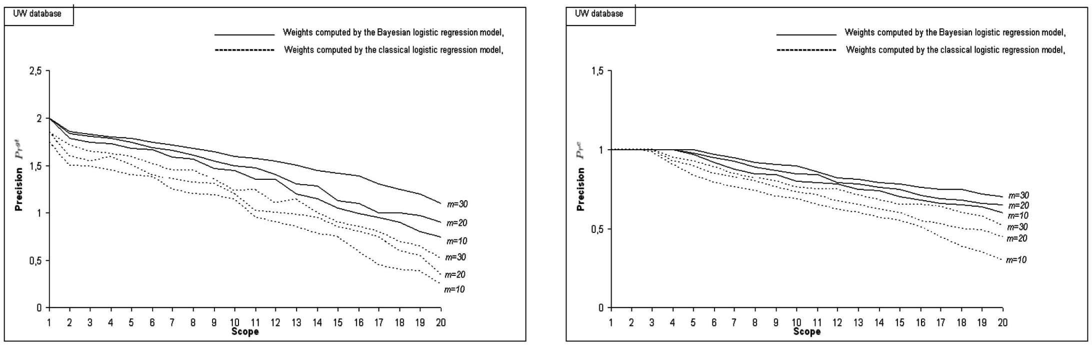


(a) (b)

Fig. 3. Evaluation (WANG database): (a) ground-truth precision-scope curves and (b) classification precision-scope curves for retrieval using weights computed by the CLRM and weights computed by the BLRM for the compression orders $m \in \{30, 20, 10\}$.

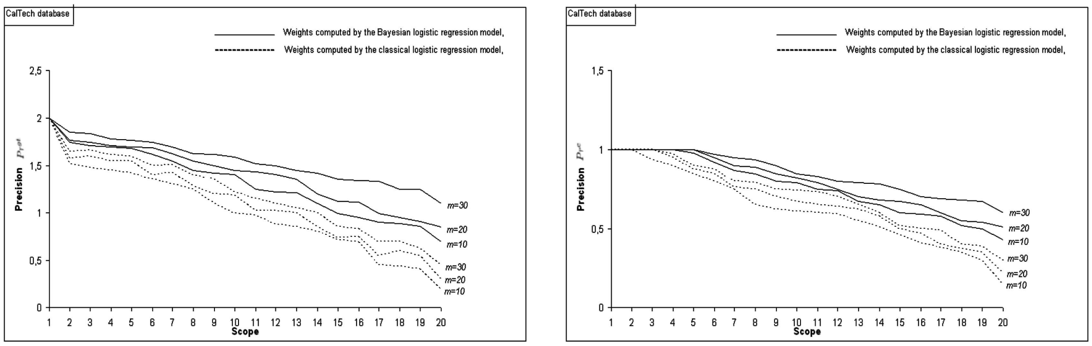
returned to the user. For ground truth, we use human observations and judgments. In fact, eight external persons participated in the evaluation described below. In the objective evaluation of the adopted classification method, the querying results are presented with reference to the prior labeling of images into classes. In each query performed in the evaluation experiment, each human subject is asked to assign a goodness score and a labeling score to each retrieved image. The goodness score is 2 if the retrieved image is almost the same as the query, 1 if the retrieved image is fairly similar to the query, and 0 if there is no similarity between the retrieved image and the query. The labeling score is 1 if the query image and the retrieved image belong to the same class and 0 otherwise. Therefore, the ground truth and classification precisions are thus computed as follows: $Pr^{gt} = \text{the}$ sum of goodness scores for retrieved images$/RI$ and $Pr^{c} = \text{the}$ sum of labeling scores for retrieved images$/RI$. The curves $Pr^{gt} = f(RI)$ and $Pr^{c} = f(RI)$ give the precisions for different values of RI, which lie between 1 and 20 when we perform the querying evaluation on the WANG, UW, and Caltech databases and between 1 and 5 when we perform the querying evaluation on the ZuBuD database. When the human subjects perform different queries in the evaluation experiment, we average the computed Pr^{gt} values and the computed Pr^{c} values for each value of RI, and then, we construct the classification and ground-truth precision-scope curves. In order to evaluate the querying procedure on the WANG database, each human subject is asked to formulate a query from the database, execute the querying procedure using weights computed by the CLRM, and assign goodness and labeling scores to each retrieved image, and then to reformulate a query from the database, execute the querying procedure using weights computed by the BLRM, and assign goodness and labeling scores to each retrieved image. Each human subject repeats the querying process 50 times, choosing a new query from the database each time. We repeat this experiment for different orders of compression $m \in \{30, 20, 10\}$, keeping the weight factors $\{\gamma_l\}_{l=1}^{3}$ equal to $\frac{1}{2}$ and $\{\gamma_l\}_{l=4}^{11}$ equal to 1 to give more importance to the edge region and texture features. Similarly, to evaluate the querying procedure on the ZuBuD, UW, and Caltech databases, each human subject is asked to follow the preceding steps. The resulting ground-truth and classification precision-scope curves for each compression order are shown in Figs. 2, 3, 4, and 5 for the ZuBuD, WANG, UW, and Caltech databases.

Thanks to the above precision-scope curves, we can notice that the BLRM is a significantly better tool than the CLRM to improve retrieval ground-truth and classification precisions. This is because of the problems related to the CLRM and mentioned in Section 3.2. In order to compare our image retrieval method when using the BLRM to others proposed in [17], [53], and [52], we chose the error rate ER as retrieval performance measure, as it has been shown in [17] to be well established for classification tasks and strongly correlated to several state-of-the-art measures. The ER is given as



(a) (b)

Fig. 4. Evaluation (UW database): (a) ground-truth precision-scope curves and (b) classification precision-scope curves for retrieval using weights computed by the CLRM and weights computed by the BLRM for the compression orders $m \in \{30, 20, 10\}$.



(a) (b)

Fig. 5. Evaluation (Caltech database): (a) ground-truth precision-scope curves and (b) classification precision-scope curves for retrieval using weights computed by the CLRM and weights computed by the BLRM for the compression orders $m \in \{30, 20, 10\}$.

TABLE 4
Comparison: ER (in Percent) Averages for Our Retrieval Method When Using the BLRM and Other Retrieval Methods

Image collection	T. Deselaers *et al.* [17]	R. Fergus *et al.* [53]	H. Shao *et al.* [52]	our retrieval method $(m = 20)$
WANG	12.7 %	-	-	8 %
UW	12.2 %	-	-	8.29 %
ZuBud	15.7 %	-	13.9 %	6.9 %
CalTech airplanes	0.8 %	9.8 %	-	1.25 %
CalTech faces	1.6 %	3.6 %	-	1.3 %
CalTech motorbikes	7.4 %	7.5 %	-	5.5 %

$1 - Pr^c(1)$, where $Pr^c(1)$ is the classification precision of the first image retrieved. If $Pr^c(1)$ is averaged over a set of queries, ER is equivalent to the percentage of incorrect images retrieved in the first rank. In [17], the four image databases were used, whereas in [53] and [52], Caltech and ZuBuD were used, respectively. To enable comparison with the results obtained in these works, we set the weight factors $\{\gamma_l\}_{l=1}^{11}$ equal to 1 to give all features the same importance, and we selected the query images as follows: for the WANG, UW, and Caltech databases, no separate train/test corpus is available; thereby, queries were selected in a leave-one-out manner. All images of WANG and UW were selected as queries, whereas for Caltech, only images of the categories motorbikes, airplanes, and faces were selected as queries. For the ZuBuD database, a separate test set of 115 query images is provided [17]. Table 4 illustrates the computed ER averages for our retrieval method and retrieval methods in [17], [53], and [52].

6 Conclusion

We have proposed an effective BLRM with a Gaussian prior distribution over the parameters of interest. This model is based on a variational approximation and on the Jensen's inequality. Thanks to these two approximations, the computation of the parameters of interest is straightforward and fast. The incorporation of the prior knowledge of the explanatory vectors in the model also optimizes the computation of the parameters of interest. Moreover, the consideration of a Gaussian prior distribution over these parameters smooths their estimates toward a fixed mean and away from the unreasonable extremes caused by the maximum likelihood routine used in the CLRM. We performed a decontextualized comparison of the BLRM to the CLRM and to some relevant state-of-the-art linear classification algorithms. Experiments showed that the BLRM outperforms these algorithms and the CLRM in terms of classification performance and training runtime. Also, we performed an evaluation and comparison of the BLRM and CLRM in the image retrieval context.

Experiments showed that the BLRM is a significantly better tool than the classical one for improving retrieval performance. Finally, we showed that our retrieval method turns out to be competitive with other retrieval methods that use the same image databases.

REFERENCES

[1] Y. Rui, T.S. Huang, and S.-F. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," *J. Visual Comm. and Image Representation,* vol. 10, pp. 39-62, 1999.

[2] A. Smeulder, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, pp. 1349-1380, 2000.

[3] N. Vasconcelos, "On the Efficient Evaluation of Probabilistic Similarity Functions for Image Retrieval," *IEEE Trans. Information Theory,* vol. 50, pp. 1482-1496, 2004.

[4] J. Peng, B. Bhanu, and S. Qing, "Learning Feature Relevance and Similarity Metrics in Image Databases," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* pp. 14-18, 1998.

[5] S. Aksoy and R.M. Haralick, "Feature Normalization and Likelihood-Based Similarity Measures for Image Retrieval," *Pattern Recognition Letters,* vol. 22, no. 5, pp. 563-582, 2001.

[6] G. Caenen and E.J. Pauwels, "Logistic Regression Models for Relevance Feedback in Content-Based Image Retrieval," *Proc. SPIE Storage and Retrieval for Media Databases,* , vol. 4676, pp. 49-58, 2002.

[7] S. Aksoy, R.M. Haralick, F.A. Cheikh, and M. Gabbouj, "A Weighted Distance Approach to Relevance Feedback," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 4, pp. 812-815, 2000.

[8] R. Ksantini, D. Ziou, and F. Dubeau, "Image Retrieval Based on Region Separation and Multiresolution Analysis," *Int'l J. Wavelets, Multiresolution and Information Processing,* vol. 4, no. 1, pp. 147-175, 2006.

[9] T.S. Jaakkola and M.I. Jordan, "Bayesian Parameter Estimation via Variational Methods," *Statistics and Computing,* vol. 10, no. 1, pp. 25-37, 2000.

[10] C.C. Clogg, D.B. Rubin, N. Schenker, B. Schultz, and L. Widman, "Multiple Imputation of Industry and Occupation Codes in Census Public-Use Samples Using Bayesian Logistic Regression," *J. Am. Statistical Assoc.,* vol. 86, pp. 68-78, 1991.

[11] R. Weiss, R. Berk, W. Li, and M. Farrell-Ross, "Death Penalty Charging in Los Angeles County: An Illustrative Data Analysis Using Skeptical Priors," *Sociological Methods and Research,* vol. 28, pp. 91-115, 1999.

[12] F. Galindo-Garre, J.K. Vermunt, and W.P. Bergsma, "Bayesian Posterior Estimation of Logit Parameters with Small Samples," *Sociological Methods and Research,* vol. 33, pp. 1-30, 2004.

[13] P. Congdon, *Bayesian Statistical Modelling.* John Wiley & Sons, 2001.

[14] G. Koop and D. Poirier, "An Empirical Investigation of Wagner's Hypothesis by Using a Model Occurrence Framework," *J. Royal Statistical Soc., Series A,* vol. 158, no. 1, pp. 123-141, 1995.

[15] R. Gerlach, R. Bird, and A.D. Hall, "A Bayesian Approach to Variable Selection in Logistic Regression with Application to Predicting Earnings Direction from Accounting Information," *Australian and New Zealand J. Statistics,* vol. 44, no. 2, pp. 155-168, 2002.

[16] S.J. Roberts, D. Husmeier, I. Rezek, and W.D. Penny, "Bayesian Approaches to Gaussian Mixture Modeling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1133-1142, Nov. 1998.

[17] T. Deselaers, D. Keysers, and H. Ney, "Classification Error Rate for Quantitative Evaluation of Content-Based Image Retrieval Systems," *Proc. 17th Int'l Conf. Pattern Recognition,* vol. 2, pp. 505-508, 2004.

[18] M.L. Yiu and N. Mamoulis, "Iterative Projected Clustering by Subspace Mining," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 2, pp. 176-189, Feb. 2005.

[19] H.A. Murthy and S. Haykin, "Bayesian Classification of Surface-Based Ice-Radar Images," *IEEE J. Oceanic Eng.,* vol. 12, no. 3, pp. 493-501, 1987.

[20] M.L. Kherfi and D. Ziou, "Relevance Feedback for CBIR: A New Approach Based on Probabilistic Feature Weighting with Positive and Negative Examples," *IEEE Trans. Image Processing,* vol. 15, no. 4, pp. 1017-1030, 2006.

[21] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance Metric Learning, with Application to Clustering with Side-Information," *Advances in Neural Information Processing Systems,* vol. 15, pp. 505-512, 2003.

[22] N. Vasconcelos, "Minimum Probability of Error Image Retrieval," *IEEE Trans. Signal Processing,* vol. 52, pp. 2322-2336, 2004.

[23] T. Westerveld and A.P. de Vries, "Generative Probabilistic Models for Multimedia Retrieval: Query Generation against Document Generation," *IEE Proc. Vision, Image, and Signal Processing,* vol. 152, no. 6, pp. 852-858, 2005.

[24] V. Lavrenko, S.L. Feng, and R. Manmatha, "Statistical Models for Automatic Video Annotation and Retrieval," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* vol. 3, pp. 17-21, 2003.

[25] S. Ghebreab, C.C. Jaffe, and A.W.M. Smeulders, "Population-Based Incremental Interactive Concept Learning for Image Retrieval by Stochastic String Segmentations," *IEEE Trans. Medical Imaging,* vol. 23, no. 6, pp. 676-689, 2004.

[26] T. Hastie, R. Tibshirani, and J.H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2001.

[27] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood Components Analysis," *Advances in Neural Information Processing Systems,* vol. 17, pp. 513-520, 2005.

[28] K. Weinberger, J. Blitzer, and L. Saul, "Neighbourhood Components Analysis," *Advances in Neural Information Processing Systems,* vol. 18, pp. 1473-1480, 2006.

[29] T. Hastie and R. Tibshirani, "Discriminant Adaptive Nearest Neighbor Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, pp. 607-616, 1996.

[30] C. Domeniconi, D. Gunopulos, and J. Peng, "Large Margin Nearest Neighbor Classifiers," *IEEE Trans. Neural Networks,* vol. 16, no. 4, pp. 899-909, 2005.

[31] A. Globerson and S. Roweis, "Metric Learning by Collapsing Classes," *Advances in Neural Information Processing Systems,* vol. 18, pp. 451-458, 2006.

[32] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, With Application to Face Verification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 539-546, 2005.

[33] J.S. Long, "Regression Models for Categorical and Limited Dependent Variables," *Advanced Quantitative Techniques in the Social Sciences.* Sage Publications, 1997.

[34] D. Hosmer and S. Lemeshow, *Applied Logistic Regression.* John Wiley & Sons, 1989.

[35] J.S. Cramer, *Econometric Applications of Maximum Likelihood Methods.* Cambridge Univ. Press, 1986.

[36] P. Komarek, "Logistic Regression for Data Mining and High-Dimensional Classification," PhD dissertation, School of Computer Science, Carnegie Mellon Univ., 2004.

[37] R.J. Freund and P.D. Minton, *Regression Methods: A Tool for Data Analysis.* Marcel Dekker, 1979.

[38] A. Albert and J.A. Anderson, "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models," *Biometrika,* vol. 71, pp. 1-10, 1984.

[39] D.J. Spiegelhalter and S.L. Lauritzen, "Sequential Updating of Conditional Probabilities on Directed Graphical Structures," *Networks,* vol. 20, pp. 579-605, 1990.

[40] M.S. Bazaraa and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms.* John Wiley & Sons, 1979.

[41] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization.* Academic Press, 1989.

[42] M. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines.* Cambridge Univ. Press, 2000.

[43] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *J. Machine Learning Research,* vol. 1, pp. 211-244, 2001.

[44] N.D. Lawrence, M. Seeger, and R. Herbrich, "Fast Sparse Gaussian Process Methods: The Informative Vector Machine," *Advances in Neural Information Processing Systems,* vol. 15, pp. 609-616, 2003.

[45] www.svmlight.joachims.org, 2007.

[46] L. Breiman, "Bias, Variance and Arcing Classifiers," Technical Report 460, Dept. Statistics, Univ. of California, 1996.

[47] M. Pohar, M. Blas, and S. Turk, "Comparison of Logistic Regression and Linear Discriminant Analysis: A Simulation Study," *Metodoloski Zvezki,* vol. 1, no. 1, pp. 143-161, 2004.

[48] S. Wang and H. Sun, "Measuring Overlap-Rate for Cluster Merging in a Hierarchical Approach to Color Image Segmentation," *Int'l J. Fuzzy Systems,* vol. 6, no. 3, pp. 147-156, 2004.

[49] http://ida.first.gmd.de/~raetsch/, 1999.

[50] A. Klautau, "Discriminative Gaussian Mixture Models: A Comparison with Kernel Classifiers," *Proc. 20th Int'l Conf. Machine Learning,* pp. 353-360, 2003.

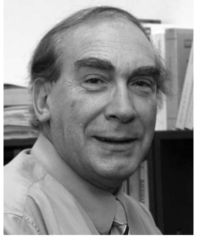
[51] I. Daubechies, *Ten Lectures on Wavelets.* SIAM, 1992.

[52] H. Shao, T. Svoboda, T. Tuytelaars, and L.V. Gool, "HPAT Indexing for Fast Object/Scene Recognition Based on Local Appearance," *Proc. Int'l Conf. Image and Video Retrieval,* pp. 71-80, 2003.

[53] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 264-271, 2003.

Riadh Ksantini received the bachelor's degree in applied mathematics from the Faculté des Sciences, Monastir, Tunisia, in 2000 and the MSc degree in computer science from the Université de Sherbrooke, Sherbrooke, Québec, Canada, in 2003. He is currently pursuing the PhD degree at the Université de Sherbrooke. His research interests include image processing, image retrieval, computer vision, data mining, and pattern recognition.

Djemel Ziou received the BEng degree in computer science from the University of Annaba, Algeria, in 1984 and the PhD degree in computer science from the Institut National Polytechnique de Lorraine (INPL), France, in 1991. From 1987 to 1993, he served as a lecturer in several universities in France. During the same period, he was a researcher at the Centre de Recherche en Informatique de Nancy (CRIN) and the Institut National de Recherche en Informatique et Automatique (INRIA) in France. Presently, he is a full professor at the Department of Computer Science, Université de Sherbrooke, Québec, Canada. He is holder of the Natural Sciences and Engineering Research Council (NSERC)/Bell Canada Research Chair in personal imaging. He has served on numerous conference committees as a member or chair. He heads the laboratory MOdélisation en Imagerie, Vision et REseaux de neurones (MOIVRE) and the consortium CoRIMedia, which he founded. His research interests include image processing, information retrieval, computer vision, and pattern recognition.



Bernard Colin received the MSc degree from Ecole Nationale Supérieure des Arts et Métiers, Paris, in 1966 and the PhD degree in mathematical statistics from Université Pierre et Marie Curie (Paris VI) in 1971. Since 1969, he has been with the Departement of Mathematics, Sherbrooke University, Québec, Canada. His research interests include mathematical statistics, Bayesian inference, information theory and coding, functional data analysis, and nonparametric statistics. He is also interested with applications in medical diagnosis, data analysis, statistical inference, and data processing.

Francois Dubeau received the BScA degree in engineering physics in 1971 and the MScA degree in industrial engineering in 1973 from École Polytechnique de Montréal and the PhD degree in mathematics from the University of Montréal in 1981. He is a professor in the Department of Mathematics, University of Sherbrooke, Québec, Canada, since 1992. He taught at the Collège Militaire Royal de St-Jean from 1982 to 1992. His research interests include applied mathematics, operational research, numerical analysis, and digital image processing.