RECHERCHE D'IMAGES PAR LE CONTENU, ANALYSE MULTIRÉSOLUTION ET MODÈLES DE RÉGRESSION LOGISTIQUE

par

Riadh Ksantini

Thèse présentée au Département d'informatique
en vue de l'obtention du grade de philosophiae doctor (Ph.D.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, octobre 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-42627-2
Our file Notre référence
ISBN: 978-0-494-42627-2

Le 5 octobre 2007

le jury a accepté la thèse de M. Riadh Ksantini dans sa version finale.

Membres du jury

M. François Dubeau
Directeur
Département de mathématiques

M. Djemel Ziou
Codirecteur
Département d'informatique

M. Richard Egli
Membre
Département d'informatique

M. Hamid Krim
Membre externe
Electrical and Computer Engineering Department - NC State University

M. Bernard Colin
Président-rapporteur
Département de mathématiques

À ma chère mère Najet, mon frère Mehdi et ma femme Hanane.

SOMMAIRE

Cette thèse, présente l'ensemble de nos contributions relatives à la recherche d'images par le contenu à l'aide de l'analyse multirésolution ainsi qu'à la classification linéaire et nonlinéaire. Dans la première partie, nous proposons une méthode simple et rapide de recherche d'images par le contenu. Pour représenter les images couleurs, nous introduisons de nouveaux descripteurs de caractéristiques qui sont des histogrammes pondérés par le gradient multispectral. Afin de mesurer le degré de similarité entre deux images d'une façon rapide et efficace, nous utilisons une pseudo-métrique pondérée qui utilise la décomposition en ondelettes et la compression des histogrammes extraits des images. Les poids de la pseudo-métrique sont ajustés à l'aide du modèle classique de régression logistique afin d'améliorer sa capacité à discriminer et la précision de la recherche. Dans la deuxième partie, nous proposons un nouveau modèle bayésien de régression logistique fondé sur une méthode variationnelle. Une comparaison de ce nouveau modèle au modèle classique de régression logistique est effectuée dans le cadre de la recherche d'images. Nous illustrons par la suite que le modèle bayésien permet par rapport au modèle classique une amélioration notoire de la capacité à discriminer de la pseudo-métrique et de la précision de recherche. Dans la troisième partie, nous détaillons la dérivation du nouveau modèle bayésien de régression logistique fondé sur une méthode variationnelle et nous comparons ce modèle au modèle classique de régression logistique ainsi qu'à d'autres classificateurs linéaires présents dans la littérature. Nous comparons par la suite, notre méthode de recherche, utilisant le modèle bayésien de régression logistique, à d'autres méthodes de recherches déjà publiées. Dans la quatrième partie, nous introduisons la sélection des caractéristiques pour améliorer notre méthode de recherche utilisant le modèle introduit ci-dessus. En effet, la sélection des caractéristiques permet de donner automatiquement plus d'importance aux caractéristiques qui discriminent le plus et moins d'importance aux caractéristiques qui discriminent le moins. Finalement, dans la cinquième partie, nous proposons un nouveau modèle bayésien d'analyse discriminante logistique construit à l'aide de noyaux permettant ainsi une classification nonlinéaire flexible.

REMERCIEMENTS

Je tiens en premier lieu à exprimer toute ma reconnaissance envers mon directeur de recherche, le Professeur François Dubeau de l'Université de Sherbrooke pour avoir accepté de diriger mes travaux. Son encadrement, ses conseils avisés et sa disponibilité furent pour moi des éléments importants quant à la bonne conduite de mes projets de recherche.

Je voudrais également remercier mon codirecteur, le Professeur Djemel Ziou de l'Université de Sherbrooke pour les discussions profitables que j'ai eues avec lui, ainsi que pour ses judicieuses critiques et ses suggestions pertinentes.

Je voudrais également remercier le Professeur Bernard Colin de l'Université de Sherbrooke pour les discussions profitables que j'ai eues avec lui, ainsi que pour ses judicieuses critiques et ses suggestions pertinentes.

Mes remerciements vont aux Laboratoires Universitaires Bell, à Patrimoine Canada, à l'ISM, au Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG) ainsi qu'à la Faculté des sciences et à mes directeur et codirecteur de recherche pour leur soutien financier.

Une pensée spéciale va à mes collègues du groupe MOIVRE de l'Université de Sherbrooke pour leur soutien scientifique et l'excellente ambiance de travail. Je ne peux passer sous silence l'apport du professionnel de recherche Karim Hamel qui m'a aidé dans la mise en œuvre de l'un de mes travaux.

Je tiens finalement à remercier tous les membres de ma famille ainsi que tous ceux qui me sont proches pour leur encouragement et leur soutien. Un remerciement spécial va à ma mère pour son aide et son soutien éternel sans lequel je n'aurais pas pu continuer mes études.

Riadh Ksantini
Sherbrooke, juillet 2007

TABLE DES MATIÈRES

INTRODUCTION

Avec le développement réalisé récemment dans les techniques de production, de transmission, et de traitement des données, il y a eu une explosion dans la quantité et la complexité des données générées chaque jour. Ces données prennent différents formats incluant le texte, les images, le son, la vidéo et le multimédia. Elles peuvent être trouvées sous forme de bases de données, de collections non organisées, ou encore sur le World Wide Web. À titre d'exemple, le nombre de pages Web référencées par *Google* s'élève à plus de 4 milliards en ce moment. Par contre, cette abondance d'information n'a pas que des impacts positifs. Le grand paradoxe auquel sont confrontés les gens actuellement est qu'il y a de plus en plus de données disponibles à propos d'un sujet spécifique, et qu'il est de plus en plus difficile de localiser l'information pertinente dans des délais raisonnables. Ceci a donné naissance à un nouveau besoin qui est celui d'inventer des outils qui aident les gens à localiser l'information voulue, ces outils sont les moteurs de recherche d'information. Nous pouvons donc définir un moteur de recherche comme étant un outil auquel l'utilisateur soumet une requête, textuelle ou autre, et qui se charge de chercher dans une collection de données tous les articles qui lui correspondent. Les premiers moteurs de recherche à avoir vu le jour et à avoir suscité beaucoup d'intérêt à la fois parmi les chercheurs et parmi les utilisateurs sont les moteurs de recherche de texte. Le fait que les pages Web telles que celles de *Google* ou de *Yahoo!* figurent parmi les pages les plus visitées sur Internet illustre bien l'importance et l'utilité de tels outils. Cependant, en dépit de la quantité d'information visuelle dans les bases de données et le Web, peu de gens se sont intéressés au problème de la recherche d'images, et la plupart des moteurs existants sont primitifs et leur performance reste assez limitée.

En recherche d'images, la première approche à avoir vu le jour trouve ses origines dans les algorithmes de recherche de texte. Pour pouvoir rechercher les images, on commence par les annoter avec du texte et ensuite les techniques de recherche de texte sont appliquées pour retrouver des images. Cette approche, connue sous le nom de "recherche d'images basée sur le texte", date des années 70 et est due à la communauté de gestion des bases de données. Même si quelques systèmes commerciaux tels que *Google image search* et *AltaVista photo finder* l'ont adoptée, cette technique souffre de plusieurs limitations. Premièrement, plusieurs collections d'images ne sont pas annotées avec du texte et leur annotation manuelle peut s'avérer fastidieuse et très coûteuse. Deuxièmement, même quand une collection est annotée, son annotation est généralement faite par des humains et peut par conséquent être subjective : deux personnes différentes peuvent utiliser des termes différents pour annoter la même image. En plus de cela, se baser exclusivement sur le texte est souvent insuffisant surtout quand les usagers sont intéressés par les composantes visuelles de l'image qui peuvent difficilement être décrites par des mots. En effet, une image peut contenir plusieurs objets et chaque objet peut posséder une longue liste d'attributs, ce qui défie la description avec les mots.

Ces limitations ont poussé les gens à réfléchir à une autre solution consistant à "laisser les images se décrire par elles-mêmes". Ceci a donné naissance à une seconde approche basée sur les caractéristiques visuelles des images telles que la couleur et la texture. Cette approche, connue sous le nom de "recherche d'images basée sur le contenu" (CBIR), a été proposée au début des années 90 et vient de la communauté de vision par ordinateur. Plusieurs moteurs de recherche récents l'ont adoptée tels que

QBIC, SIMPLIcity et Cires. Les premiers moteurs de recherche basés sur le contenu exigeaient de l'usager de sélectionner les caractéristiques visuelles qui l'intéressent et de fournir des valeurs numériques à chacune de ces caractéristiques. Cependant pour différentes raisons, il est généralement difficile pour l'usager de spécifier explicitement les valeurs des caractéristiques visuelles. Tout d'abord, un usager peut ignorer les détails de l'imagerie et son jargon. Pour s'en convaincre, imaginons un usager auquel on demande de choisir entre le "filtre de Gabor" et les "Ondelettes" par exemple ! Ensuite, il est difficile, même pour un spécialiste en imagerie, de traduire les images qu'il a en tête en une combinaison de caractéristiques et de valeurs numériques. Dès lors, les gens se sont mis à réfléchir à une autre alternative, et la solution qu'ils ont adoptée consiste à permettre à l'usager de spécifier implicitement les caractéristiques qui l'intéressent à travers un paradigme connu sous le nom de "requête par l'exemple" (QBE). En utilisant l'interface que le moteur lui offre, l'usager choisit une image requête et ensuite le moteur parcourt la collection de données en extrayant toutes les images qui ressemblent à cette requête. Précisément, la requête choisie par l'usager et les images de la base de données sont initialement représentées par des vecteurs de caractéristiques. Ensuite, la similarité entre l'image requête et une image cible est mesurée par une métrique qui est calculée entre leurs vecteurs de caractéristiques. Enfin, les images cibles les plus proches de l'image requête, au sens de la métrique, sont retournées à l'usager. La métrique doit être assez flexible, pour tenir compte des distorsions de la requête par rapport à la cible, et aussi assez rapide d'exécution, pour pouvoir être utilisée sur de grandes bases de données.

Dans cette thèse, notre objectif est de développer une méthode de recherche d'images basée sur le contenu qui soit à la fois efficace et rapide. Pour ce faire, nous avons utilisé plusieurs outils comme les ondelettes, de nouveaux descripteurs ou vecteurs des caractéristiques, une structure de données spécifique et la classification linéaire et nonlinéaire. Par conséquent, nous avons fait des contributions relatives à la recherche d'image par le contenu, la description des caractéristique et aussi la classification linéaire et nonlinéaire. Dans le Chapitre 1 de la thèse, nous proposons une méthode simple et rapide de recherche d'images par le contenu. Pour représenter les images couleurs, nous introduisons de nouveaux descripteurs de caractéristiques qui sont des histogrammes pondérés par le gradient multispectral. Afin de mesurer le degré de similarité entre deux images d'une façon rapide et efficace, nous utilisons une pseudo-métrique pondérée qui utilise la décomposition en ondelettes et la compression des histogrammes extraits des images. Les poids de la pseudo-métrique sont ajustés à l'aide du modèle classique de régression logistique afin d'améliorer sa capacité à discriminer et la précision de la recherche. Dans le Chapitre 2, nous proposons un nouveau modèle bayésien de régression logistique fondé sur une méthode variationnelle. Une comparaison de ce nouveau modèle au modèle classique de régression logistique est effectuée dans le cadre de la recherche d'images. Nous illustrons par la suite que le modèle bayésien permet par rapport au modèle classique une amélioration notoire de la capacité à discriminer de la pseudo-métrique et de la précision de recherche. Dans le Chapitre 3, nous détaillons la dérivation du nouveau modèle bayésien de régression logistique fondé sur une méthode variationnelle et nous comparons ce modèle au modèle classique de régression logistique ainsi qu'à d'autres classificateurs linéaires présents dans la littérature. Nous comparons par la suite, notre méthode de recherche, utilisant le modèle bayésien de régression logistique, à d'autres méthodes de recherches déjà publiées. Dans le Chapitre 4, nous introduisons la sélection des caractéristiques pour améliorer notre méthode de recherche utilisant le modèle introduit ci-dessus. En effet, la sélection des caractéristiques permet de donner automatiquement plus d'importance aux caractéristiques qui discriminent le plus et moins d'importance aux caractéristiques qui discriminent le moins. Finalement, dans le Chapitre 5, nous proposons un nouveau modèle bayésien d'analyse discriminante logistique construit à l'aide de noyaux permettant ainsi une classification nonlinéaire flexible.

Dans l'ensemble des articles qui suivent, la mise en oeuvre des contributions omniprésentes a été faite par l'auteur principal Riadh Ksantini avec l'aide précieuse des professeurs François Dubeau, Djemel Ziou et Bernard Colin.

CHAPITRE 1

Recherche d'images fondée sur la séparation des régions et l'analyse multirésolution

Dans ce chapitre, nous proposons une méthode simple et rapide de recherche d'images par le contenu. En utilisant le gradient multispectral, une image couleur est coupée en deux parties disjointes : les régions homogènes de couleur et les régions de contours. Les régions homogènes sont représentées par les histogrammes traditionnels de couleur et les régions de contours sont représentées par les histogrammes des moyennes des modules du gradient multispectral calculées sur chaque pixel de l'image couleur. Afin de mesurer le degré de similarité entre deux images couleurs rapidement et efficacement, nous utilisons une pseudo-métrique pondérée qui se sert de la décomposition en ondelettes Daubechies-8 et de la compression des histogrammes extraits. Les poids de la pseudo-métrique sont ajustés par le modèle classique de régression logistique pour améliorer sa capacité à discriminer et la précision de la recherche. Notre méthode de recherche est invariante aux translations des objets et aux intensités de couleur dans les images. Les expérimentations ont été effectuées sur une collection de 10000 images couleurs.

Nous présentons dans les pages qui suivent, un article intitulé **Image Retrieval Based on Region Separation and Multiresolution Analysis** qui a été publié dans le numéro de mars 2006 du **International Journal of Wavelets, Multiresolution and Information Processing (IJWMIP)**. Une version préliminaire de l'article a été présentée dans la Conférence Internationale en Recherche Opérationnelle (CIRO'05), Marrakech, Maroc, 2005.

Image Retrieval Based on Region Separation and Multiresolution Analysis

R. Ksantini[1], D. Ziou[1], and F. Dubeau[2]

(1) Département d'informatique, Faculté des sciences
Université de Sherbrooke
Sherbrooke, Qc, Canada J1K 2R1.
Email: riadh.ksantini@usherbrooke.ca
djemel.ziou@usherbrooke.ca
(2) Département de mathématiques, Faculté des sciences
Université de Sherbrooke
Sherbrooke, Qc, Canada J1K 2R1.
Email: francois.dubeau@usherbrooke.ca

Keywords: Multispectral Gradient, Region Separation, Multiresolution Analysis, Pseudo-metric, Logistic Regression, Color Image Retrieval.

Abstract

In this paper, a simple and fast querying method for content-based image retrieval is presented. Using the multispectral gradient, a color image is split into two disjoint parts which are the homogeneous color regions and the edge regions. The homogeneous regions are represented by the traditional color histograms, and the edge regions are represented by multispectral gradient module mean histograms. In order to measure the similarity degree between two color images both quickly and effectively, we use a one-dimensional pseudo-metric, which makes use of the one-dimensional Daubechies decomposition and compression of the extracted histograms. Our querying method is invariant to the query color image object translations and color intensities. The experimental results are reported on a collection of 10000 LAB color images.

1 Introduction

The rapid expansion of the Internet and the wide use of digital data in many real world applications in the field of medecine, weather prediction, communications, commerce and academia, increased the need for both efficient image database creation and retrieval procedures. For this reason, content-based image retrieval (CBIR) approach was proposed [19], [2]. In this approach, the first step is to compute for each database image a feature vector capturing 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, compared to the featurebase feature vectors by a distance metric or a similarity measure, and finally the most similar database images to the query image are returned to the user. In order to have effective characterization of local image properties, and to increase the data storage efficiency and the querying execution speed in the CBIR field, a wavelet based indexing approach was introduced. The wavelet transforms are proven to have the advantage of allowing better resolution in time and frequency. Consequently, they have received much attention as a tool for developing CBIR systems. In the following we review some of the CBIR systems based on the wavelet domain feature extractions.

Related work:

Jacob et al. [5] proposed a fast image querying algorithm in databases ranging in size from 1093 to 20,558 color images. RGB, HSV, and YIQ color spaces are chosen separately to represent the database color images before the querying. Each database color image component is Haar wavelets decomposed. Dominant coefficients of this decomposition are retained to represent the spatial information and color visual features of the color image. The similarity degree between a query and potential targets is measured by a weighted metric which compares how many significant wavelet coefficients they have in common.

Wang et al. [14] have proposed the WBIIS querying system in a database of 10,000 RGB color images. All database color images are four stage Daubechies-8 wavelets decomposed. The lower frequency bands in each database color image wavelet transform, represent the object configurations in the image and the higher frequency bands represent the texture and local color variations. The similarity degree between a query and potential targets is measured by a comparison between the variances of their lowpass band coefficients. Then, these latters are compared using an euclidean distance. Finally, a weighted euclidean distance is used to perform a comparison between the query and the remaining color images lowest resolution subimages representing the lowpass bands, horizontal bands, vertical bands, and diagonal bands. For database color images, this procedure is repeated on all three color channels.

Kuo et al. [15] have proposed the WaveGuide querying system in a database of 2127 YUV color images. Each database color image is wavelet packet decomposed and wavelet pyramid decomposed to extract texture, shape and color features, respectively. Both wavelet transforms are followed by a successive approximation quantization (SAQ) which uses a sequence of thresholds in order to indentify relevant and irrelevant wavelet coefficients and their locations in each wavelet transformed color image subbands. Texture descriptor is extracted from the significant coefficients in wavelet packet transformed image Y-component subbands. Color descriptor is extracted from the three color image components with respect to the (SAQ) twelve thresholds of the wavelet pyramid transformed image. Shape descriptor is extracted from the significant coefficients of the first three scale vertical, horizontal and diagonal subbands of the wavelet pyramid transformed color image Y-component. The texture, color and shape similarities between a query color image and the database color images are defined using the L_1 distance.

N. Khelil and A. Benazza-Benyahia [17] have suggested a method for image retrieval in a database of 2815 multispectral SPOT3 images and in a database of 2815 multispectral SPOT4 images. Each database color image components are decomposed separately according to the lifting scheme (second generation of wavelet transform) [1] through the 5/3 transform. The wavelet coefficients of a 5/3 transformed database color image, represent the salient features of the image. The wavelet coefficients related to all components at a given resolution level of each database 5/3 transformed color image, are merged into a common subband whatever the transform orientations, and then this subband is modelized by a zero-mean Generalized Gaussian Distribution (GGD). The similarity degree between the query image and the database color images is measured by a weighted metric which is a combination of the symmetrical Kullback-Leibler distance which is used to evaluate how different are two GGDs, and by a second order distance computed between their scaling coefficient variances.

Our approach is based on the use of the multispectral gradient in order to separate between the homogeneous regions and the edge regions of each database color image, and to represent each region by feature vectors which are weighted histograms. The weighted histograms representing the homogeneous regions are color histograms constructed after edge regions elimination, and the weighted histograms representing the edge regions are multispectral gradient module mean histograms. In the querying, just very few dominant coefficients of the wavelet decomposed versions of these weighted histograms are considered to have an effective querying, despite a lower querying computational complexity. Among all kinds of wavelets, Daubechies-8 wavelets are proven to have good frequency properties and to be good for 1-D signal synthesis [14]. Therefore, Daubechies-8 wavelets are chosen in our approach. In order to measure the similarity degree between two color images we use the one-dimensional version of the weighted metric proposed by Jacob et al. [5], which makes use of the compressed and quantized versions of the Daubechies-8 wavelet decomposed histograms. In order to discriminate most effectively, the metric weights are adjusted using the standard logistic regression model. Our querying method does not suffer from the query image object translation variance, and the querying is invariant to the color intensities of the query image, thanks to a modification of the multispectral gradient module mean histograms. We apply our retrieval method by representing our database color images in the LAB color space, because it's a perceptually uniform color space that describes color by just two coordinates.

The difference between the related work approaches and the approach developed in this paper, is that in order to reduce the computational complexity and to increase the data storage efficiency our approach is based on the wavelet decomposition and compression of the feature vectors themselves, instead of the database color images. A variety of heuristic histogram similarity measures has been proposed in literature in the context of image retrieval [18]. However, these similarity measures do not handle wavelet decomposed and compressed histograms. In fact, they are computed over the totality of the histogram pixels. Consequently, they are more expensive to compute, especially, when we have a large database. For these reasons, in our application we use the one-dimensional version of the weighted metric proposed by Jacob et al. [5].

In the next section, we explain how we construct homogeneous and edge region histograms. In the third section, we briefly explain the Daubechies-8 wavelet decomposition and the compression of a one-dimensional image. In section 4, we define the one-dimensional version of the metric proposed by [5], we explain the one-dimensional image or feature vector querying algorithm and we describe the logistic regression model and the training performed to adjust the metric weights. In section 5, we present the color image querying method. Finally, in section 6 we perform some experiments to evaluate our querying method and to show the querying improvement.

2 Color images and histogram decomposition

In this section, we present the LAB color space advantages and we explain how to represent our database color images in this space. Also, we explain the separation between the homogeneous color regions and edge regions, using the multispectral gradient. Then, we define the two feature vectors which are the traditional color histograms constructed without considering the edge regions, and the multispectral gradient module mean histogram.

2.1 Color images and color space

In order to extract color features from a color image, we need to choose a color space in which to represent it. We decided to use the LAB color space. In fact, LAB color space is approximately perceptually uniform [11], [7], [3], that maps equally distinct color differences into approximately equal euclidean distances in space. Also, it allows a good separation between the luminance and the colors. In this space, L defines the luminance with values from 0 for black to 100 for a perfectly white body, A denotes the red/green chrominance, with values from -200 for green to 200 for red, and B denotes the yellow/blue chrominance, with values from -200 for blue to 200 for yellow. In the LAB color space, each color image pixel is represented by a vector (L, a, b), where L is the luminance, a is the red/green and b is the yellow/blue. In our application, each LAB color image is numerically represented by three matrices I_L, I_a and I_b, containing the pixel intensities of the luminance, red/green and yellow/blue, respectively. To simplify, we use a linear interpolation to represent each LAB color image component intensities between 0 and 255.

2.2 Weighted histograms

The luminance histogram and the color histogram represent how pixels of some images are distributed in the LAB channels. Given any LAB color image, its luminance histogram h_L contains the number of pixels of the luminance L, and its color histograms h_a and h_b contain the number of pixels of the chrominances a red/green and b yellow/blue, respectively. Therefore, the three histograms of an $M \times N$ pixel LAB color image, can be written as follows

$$\begin{aligned} h_L(c) &= \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \delta(I_L(i,j) - c) \\ h_a(c) &= \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \delta(I_a(i,j) - c) \qquad \text{for each } c \in \{0, ..., 255\} \\ h_b(c) &= \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \delta(I_b(i,j) - c), \end{aligned} \tag{1}$$

where δ is the Kroeneicker symbol at 0, defined by

$$\delta(x-c) = \begin{cases} 1 & \text{if } x = c \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

LAB histograms have been used widely in many content-based image retrieval systems with some success [16]. They provide only the distribution of the luminance and the color. That's why histogram-based color retrieval techniques suffer from a lack of important spatial knowledge.

In order to overcome these drawbacks, spatial information should be integrated. Several recently proposed approaches augment the color histogram with some spatial information. Examples include the laplacian weighted histogram proposed by [6], the color coherent vector (CCV) proposed by [9] and enhanced by [10], and the color correlogram proposed by [13]. In our case we will use the multispectral gradient module mean histogram. The multispectral gradient changes from a color image to another having different edge shapes, which increases the discriminative power of the querying. The multispectral gradient module mean histogram is inspired from the laplacian weighted histogram which is proposed by [6]. This latter is a good tool to distinguish the pixels located at the neighbourhoods of a color image edges. However, it's noised because it's based on the color image component second derivatives, also it can represent a false feature when the color image contains several staircase edges [12].

In a color image, the number of the pixels belonging to the homogeneous regions is widely greater than the number of the edge pixels. Therefore, these latters have negligible influence on the color histogram shape, and then their statistical importance becomes insignificant, thus rendering their effect on the querying very negligible. A solution to this problem is to separate between the homogeneous regions and the edge regions. In our application, this separation will be only performed on the chrominance images. That's why we will preserve the luminance histogram and we will introduce weighted color histograms which combine the color distribution with its spatial properties.

Once we have identified the homogeneous and edge regions for a given color image, we consider two weighted color histograms for each region

$$h_k^l(c) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \delta(I_k(i,j)-c)W_k^l(i,j) \qquad \text{for each } c \in \{0,...,255\} \text{ and } k = a, b, \tag{3}$$

where W_k^l is the weight and

$$l = \begin{cases} h & \text{for homogeneous regions,} \\ e & \text{for edge regions.} \end{cases} \tag{4}$$

In [6], the weight $W_k^l(i,j) = \Delta I_k(i,j)^2$ for each $k = a, b$, where $\Delta I(i,j)$ is the laplacian of the image I_k at the pixel (i,j).

2.3 Separation of homogeneous and edge regions

According to [4], a color edge detection is based on finding local maxima in the first directional derivative of the vector-valued color image. The magnitude of the strongest change of the vector-valued image which represents the multispectral gradient module, coincides with the largest eigenvalue of the matrix J^TJ, denoted by λ_{max}, where J is the Jacobian matrix of the vector-valued image. In our case, the Jacobian matrix of the chrominance images at each pixel (x,y), is given by

$$J = \begin{pmatrix} \frac{\partial I_a(x,y)}{\partial x} & \frac{\partial I_a(x,y)}{\partial y} \\ \frac{\partial I_b(x,y)}{\partial x} & \frac{\partial I_b(x,y)}{\partial y} \end{pmatrix}, \tag{5}$$

where $\frac{\partial I_a(x,y)}{\partial x}$, $\frac{\partial I_a(x,y)}{\partial y}$, $\frac{\partial I_b(x,y)}{\partial x}$, and $\frac{\partial I_b(x,y)}{\partial y}$ are the first partial derivatives of a and b images, respectively. Consequently, the matrix J^TJ at each pixel (x,y), is defined by

$$J^TJ = \begin{pmatrix} a_{11}(x,y) & a_{12}(x,y) \\ a_{21}(x,y) & a_{22}(x,y) \end{pmatrix}, \tag{6}$$

where

$$\begin{aligned} a_{11}(x,y) &= \left(\frac{\partial I_a(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I_b(x,y)}{\partial x}\right)^2, \\ a_{22}(x,y) &= \left(\frac{\partial I_a(x,y)}{\partial y}\right)^2 + \left(\frac{\partial I_b(x,y)}{\partial y}\right)^2, \\ a_{12}(x,y) &= \frac{\partial I_a(x,y)}{\partial x}\frac{\partial I_a(x,y)}{\partial y} + \frac{\partial I_b(x,y)}{\partial x}\frac{\partial I_b(x,y)}{\partial y}, \\ a_{21}(x,y) &= a_{12}(x,y). \end{aligned}$$

Therefore, the largest eigenvalue λ_{max} of $J^T J$, which represents the strongest change of the vector-valued image at each pixel (x, y), is given by

$$\lambda_{max}(x,y) = \frac{1}{2}\Big((a_{11}(x,y) + a_{22}(x,y)) + \sqrt{(a_{11}(x,y) - a_{22}(x,y))^2 + 4a_{12}^2(x,y)} \Big). \tag{7}$$

A pixel is considered in an edge region if the λ_{max} computed over it is greater than a given threshold η, and is considered in homogeneous region elsewhere. In our application, we simply use a threshold defined by the mean of the largest eigenvalues computed over all pixels. Explicitly

$$\eta = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \lambda_{max}(i,j), \tag{8}$$

where M and N are respectively the length and the width of the color image. Let us remark that other strategies for thresholding are also possible.

2.4 Homogeneous regions : separation of modes

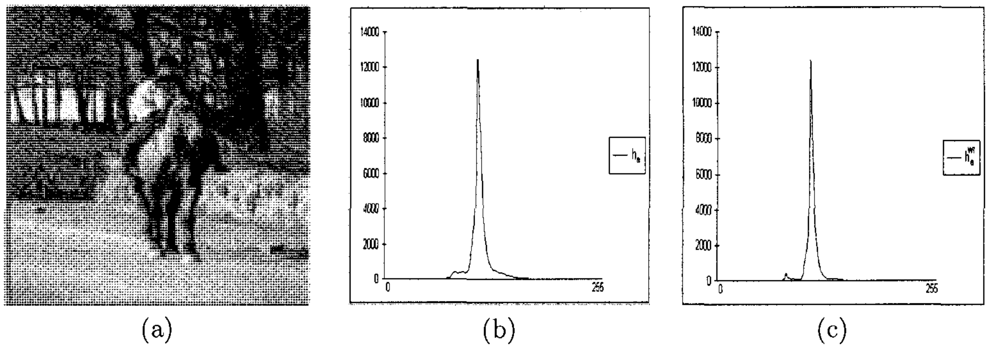
For these regions the weight $W_k^h(i,j)$ in the formula (3) is given by $W_k^h(i,j) = \chi_{[0,\eta]}\Big(\lambda_{max}(i,j)\Big)$, for each $k = a, b$. Therefore, the weighted histograms are

$$h_k^h(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k(i,j) - c)\chi_{[0,\eta]}\Big(\lambda_{max}(i,j)\Big), \tag{9}$$

for each $c \in \{0, ..., 255\}$ and $k = a, b$, and where χ_E is the characteristic function of the set E, defined by

$$\chi_E(x) = \begin{cases} 1 & \text{if } x \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

In some images, edge pixels can cause overlappings or noises between the color histogram populations. A consequence of not considering edge pixels in the formula (10) is the avoidance of these overlappings or noises. The following figure gives an example of the separation between two modes of a histogram.



(a) (b) (c)

Figure 1: Separation between two histogram modes: a) Color image, b) The a component histogram before edge region elimination and c) The a component weighted histogram.

2.5 Edge regions

For these regions the weight $W_k^e(i,j)$ in the formula (3) is given by $W_k^e(i,j) = \chi_{]\eta,+\infty[}\Big(\lambda_{max}(i,j)\Big)\lambda_{max}(i,j)$, for each $k = a, b$. Therefore, the weighted color histograms are the multispectral gradient module weighted histograms which are given by

$$h_k^e(c) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \delta(I_k(i,j) - c)\ \chi_{]\eta,+\infty[}\Big(\lambda_{max}(i,j)\Big)\lambda_{max}(i,j), \tag{11}$$

for each $c \in \{0, ..., 255\}$ and $k = a, b$. Thanks to the term $\chi_{]\eta,+\infty[}\Big(\lambda_{max}(i,j)\Big)$, the weighted histogram $h_k^e(c)$ takes into account the high values of the multispectral gradient modules. Thus, it provides information about the overall contrast in chrominances.

According to the formula (11), two color images having same colors and object shapes, but different object sizes, can have different multispectral gradient module weighted histograms. To overcome this drawback, we can consider the means of the multispectral gradient modules. In fact, the means represent the gradient module global values of a chrominance. For each chrominance, the multispectral gradient module mean histogram is given by

$$\bar{h}_k^e(c) = \frac{h_k^e(c)}{N_{p,k}(c)}, \qquad \text{for each } c \in \{0, ..., 255\} \text{ and } k = a, b, \tag{12}$$

where $N_{p,k}(c)$ is the number of the edge region pixels and is defined as

$$N_{p,k}(c) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \delta(I_k(i,j) - c)\chi_{]\eta,+\infty[}\Big(\lambda_{max}(i,j)\Big), \tag{13}$$

for $k = a, b$.

3 Wavelet decomposition, compression and quantization of an histogram

3.1 Histogram and multiresolution analysis

An histogram is a (1D) image or signal supported by 2^J pixels ($J \in \mathbb{N}$) and represented by a sequence of coefficients $\{I_i^J\}_{i=0}^{2^J-1}$. In order to analyse the histogram we use the Daubechies-8 wavelets. In fact, Daubechies-8 wavelets are continuous functions which can analyse continuous (1D) signals more efficiently. Also, Daubechies-8 wavelets are a good compromise between computational time and performances, since they have eight coefficient overlapping filters. Furthermore, they have four vanishing moments which produce as marked a contrast in wavelet coefficient sizes between smooth and non-smooth sections of the (1D) signal.

The Daubechies-8 scaling function is compactly supported by the interval $[0, 7]$ and its values can be calculated thanks to a given initial value and to a recurrence relation which is given by

$$\phi(x) = \sum_{i=0}^{i=7} h_i \phi(2x - i), \tag{14}$$

where ($h_0 = -0.014, h_1 = 0.046, h_2 = 0.043, h_3 = -0.264, h_4 = -0.039, h_5 = 0.890, h_6 = 1.009, h_7 = 0.325$) is the lowpass filter.

The Daubechies-8 scaling function ϕ serves as the basic building block for its associated Daubechies-8 wavelet function, denoted by ψ, and defined by the following recursion

$$\psi(x) = \sum_{i=-6}^{i=1} (-1)^i h_{1-i} \phi(2x - i). \tag{15}$$

In order to ensure that ϕ and ψ are compactly supported by the same interval $[0, 7]$ and are equal zero outside it, we shift the Daubechies-8 wavelet function ψ from x to $x - 3$. The scaled, translated and normalized versions of ϕ and the shifted version of ψ are denoted by $\tilde{\phi}_i^j(x) = \sqrt{2^{-(J-j)}}\phi(2^{-(J-j)}x - i)$ and $\tilde{\psi}_i^j(x - 3) = \sqrt{2^{-(J-j)}}\psi(2^{-(J-j)}x - 3 - i)$, respectively, where $2^{-(J-j)}$ is the dilatation factor and $j \in \{J, J-1, ..., 0\}$.

For the finest resolution level, we introduce a vector space V^J which is the set of all possible linear combinations of the Daubechies-8 scaling function shifted versions. With implicit periodicity considerations, we can write

$$V^J = Lin\{\tilde{\phi}_i^J : i = 0, ..., 2^J - 1\}. \tag{16}$$

By supposing that the (1D) signal $I \in V^J$, we can approximate it as follows

$$I(x) \simeq \sum_{i=0}^{2^J-1} I_i^J \tilde{\phi}(x - i). \tag{17}$$

We define the vector subspace $W^{J-1} = Lin\{\tilde{\psi}_i^{J-1} : i = 0, ..., 2^{J-1} - 1\}$ to be the orthogonal complement of V^{J-1} in V^J. Explicitly

$$V^J = V^{J-1} \oplus W^{J-1}. \tag{18}$$

The Daubechies-8 wavelet transform decomposes the 2^J pixel (1D) signal into its components for J different scales. It consists of passing successively from the space V^J to the space V^0, while generating through this decomposition the spaces $\{W^j\}_{j=0}^{J-1}$. So we obtain

$$V^J = V^0 \oplus \bigoplus_{k=0}^{J-1} W^k. \tag{19}$$

Consequently, we can rewrite the 2^J pixel (1D) signal in the Daubechies-8 basis as follows

$$I(x) \simeq \tilde{I}_0^0 \tilde{\phi}_0^0(x) + \sum_{j=0}^{J-1} \sum_{k=0}^{2^j-1} \tilde{I}_k^j \tilde{\psi}_k^j(x - 3), \tag{20}$$

where $\tilde{I}_0^0$ is the overall average of the (1D) signal, called the scaling factor. The Daubechies-8 wavelets transform decomposes an histogram having 256 pixels into its components for 8 different scales. Consequently, the smooth components and the detailed components of the histogram are readily seperated. The resulted scaling factor represents the average of the histogram Y-coordinate magnitudes. Therefore, the scaling factor of a Daubechies-8 wavelet decomposed histogram of a LAB color image component, represents the average of the overall intensity of this latter.

3.2 Signal compression and quantization

The compression is carried out on the number of the retained coefficients representing the decomposed (1D) image. The wavelet coefficients represent the local intensity variations in the image. Their magnitudes represent the importance of the variations, but their signs express the type of these variations. If we keep only the coefficients with largest magnitudes, we can obtain a good approximation of the decomposed image. Only the absolute values of the wavelet coefficients are taken into account during the compression. In the compression method we rewrite the decomposed (1D) image (20) as follows

$$I(x) \simeq \tilde{I}[0]\tilde{\phi}_0^0(x) + \sum_{i=1}^{2^J-1} \tilde{I}[i]\tilde{u}_i(x), \tag{21}$$

where $\tilde{I}[i] = \tilde{I}_k^j$ and $\tilde{u}_i(x) = \tilde{\psi}_k^j(x-3)$ for $(i = 2^j + k, k = 0, ..., 2^j - 1, j = 0, ..., J-1)$. By summing these coefficients in order of decreasing magnitude and by using a permutation σ, we obtain

$$I(x) \simeq \tilde{I}[0]\tilde{\phi}_0^0(x) + \sum_{i=1}^{2^J-1} \tilde{I}[\sigma(i)]\tilde{u}_{\sigma(i)}(x), \tag{22}$$

where

$$\mid \tilde{I}[\sigma(i_1)] \mid \geqslant \mid \tilde{I}[\sigma(i_2)] \mid \qquad \text{for all} \qquad 0 < i_1 < i_2. \tag{23}$$

Consequently, when we keep the m largest coefficients, we obtain an approximation $I^c(x)$ representing the compressed version of the decomposed (1D) image $I(x)$, defined by

$$I^c(x) \simeq \tilde{I}[0]\tilde{\phi}_0^0(x) + \sum_{i=1}^{2^J-1} \tilde{I}^c[i]\tilde{u}_i(x), \tag{24}$$

where

$$\tilde{I}^c[i] = \begin{cases} \tilde{I}[i] & \text{if } 1 \leqslant \sigma^{-1}(i) < m, \\ 0 & \text{if } \sigma^{-1}(i) \geqslant m. \end{cases} \tag{25}$$

This approximation introduces an error called the L^2-error given by

$$\parallel I - I^c \parallel_2 = \left[\sum_{i=m}^{2^J-1} (\tilde{I}[\sigma(i)])^2 \right]^{\frac{1}{2}}. \tag{26}$$

The retained coefficients after the compression, have the largest magnitudes and the most relevant data in the decomposed (1D) image. However, the storage of these coefficients requires a large space. For this reason, we use a quantization of our (1D) images which reduces the storage space. Every significant non-zero coefficient is quantified to just two levels: +1 represents the largest positive coefficients, and −1 represents the largest negative coefficients. Therefore, the quantified version I_q^c of the compressed (1D) image I^c is given by

$$I_q^c(x) \simeq \tilde{I}[0]\tilde{\phi}_0^0(x) + \sum_{i=1}^{2^J-1} \tilde{I}_q^c[i]\tilde{u}_i(x), \tag{27}$$

where

$$\tilde{I}_q^c[i] = \begin{cases} +1 & \text{if } \tilde{I}^c[i] > 0, \\ 0 & \text{if } \tilde{I}^c[i] = 0, \\ -1 & \text{if } \tilde{I}^c[i] < 0. \end{cases} \qquad i = 1, ..., 2^J - 1. \tag{28}$$

4 The metric and the querying algorithm

4.1 The metric or pseudo-metric

In the last section, we showed that the compression gives us a good approximation of the Daubechies-8 decomposed (1D) image and the quantization reduces its storage space. According to [5], if we consider only the signs of the retained and quantified coefficients after the compression in the querying, we can reduce the comparison algorithm execution time.

Let us consider Q and T as the query and the target (1D) images, respectively. The (1D) version of the metric, proposed by [5] is obtained from the following expression

$$\| Q, T \| = \omega_0 |\tilde{Q}[0] - \tilde{T}[0]| + \sum_{i=1}^{2^J-1} \omega_i |\tilde{Q}_q^c[i] - \tilde{T}_q^c[i]|, \tag{29}$$

where ω_i are the metric weights, $\tilde{Q}[0]$ and $\tilde{T}[0]$ are the scaling function coefficients of the 1D images Q and T, and $\tilde{Q}_q^c[i]$ and $\tilde{T}_q^c[i]$ represent the i-th decomposed, compressed and quantified coefficients of these latters. Since the possible values of a coefficient are +1 or −1, the term

$$|\tilde{Q}_q^c[i] - \tilde{T}_q^c[i]| \in \left\{0, 1, 2\right\}. \tag{30}$$

In the following case

$$|\tilde{Q}_q^c[i] - \tilde{T}_q^c[i]| = 2, \tag{31}$$

Q and T preserved their coefficients at the i-th position despite the decomposition and compression, but the two coefficient signs don't match. They have opposite variations. With respect to the distance between two different images, two opposite variations having the same positions don't represent more proximity than when one of them is equal to zero after the compression. Also, since we assume the vast majority of database images not to match to the query (1D) well at all, the number of mismatches is larger than the number of matches. That's why we write our metric as follows

$$\parallel Q, T \parallel = \omega_0 |\tilde{Q}[0] - \tilde{T}[0]| + \sum_{i=1}^{2^J-1} \omega_i \Big(\tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i] \Big), \tag{32}$$

where

$$\Big(\tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i] \Big) = \begin{cases} 1 & \text{if } \tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i] \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

In order to make the metric faster, we only consider terms in which the query has a non-zero wavelet coefficient. A disadvantage of this approach is that we technically disqualify our metric from being a metric because of its asymmetry. Because of this modification our metric becomes

$$\parallel Q, T \parallel = \omega_0 |\tilde{Q}[0] - \tilde{T}[0]| + \sum_{i:\tilde{Q}_q^c[i] \neq 0} \omega_i \Big(\tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i] \Big). \tag{34}$$

To compute the metric over a database of (1D) images, it's generally quicker to count the number of matching coefficients of $\tilde{Q}_q^c$ and $\tilde{T}_q^c$ than mismatching coefficients. For this reason, we rewrite

$$\sum_{i:\tilde{Q}_q^c[i] \neq 0} \omega_i \Big(\tilde{Q}_q^c[i] \neq \tilde{T}_q^c[i] \Big) = \sum_{i:\tilde{Q}_q^c[i] \neq 0} \omega_i - \sum_{i:\tilde{Q}_q^c[i] \neq 0} \omega_i \Big(\tilde{Q}_q^c[i] = \tilde{T}_q^c[i] \Big), \tag{35}$$

where

$$\Big(\tilde{Q}_q^c[i] = \tilde{T}_q^c[i] \Big) = \begin{cases} 1 & \text{if } \tilde{Q}_q^c[i] = \tilde{T}_q^c[i] \\ 0 & \text{otherwise.} \end{cases} \tag{36}$$

So our metric becomes

$$\parallel Q, T \parallel = \omega_0 |\tilde{Q}[0] - \tilde{T}[0]| + \sum_{i:\tilde{Q}_q^c[i] \neq 0} \omega_i - \sum_{i:\tilde{Q}_q^c[i] \neq 0} \omega_i \Big(\tilde{Q}_q^c[i] = \tilde{T}_q^c[i] \Big). \tag{37}$$

Since the sum of the terms ω_i is independant of every target $\tilde{T}_q^c$, we can discard it. Consequently, we obtain

$$\| Q, T \| = \omega_0 |\tilde{Q}[0] - \tilde{T}[0]| - \sum_{i:\tilde{Q}_q^c[i] \neq 0} \omega_i \Big(\tilde{Q}_q^c[i] = \tilde{T}_q^c[i] \Big). \tag{38}$$

To simplify, we suppose that the weights associated to the coefficients belonging to the same scale, are identical. Thus, we group the weights according to the resolution levels, by using a simple bucketing function $bin()$ such as $bin(i)$ represents the floor of $log_2(i)$ for $i \in \{1, ..., 2^J - 1\}$

$$bin(i) = \lfloor log_2(i) \rfloor \qquad \text{with} \qquad i = 1, ..., 2^J - 1. \tag{39}$$

Consequently, we use a set of weights $\tilde{w}_0$ and $\{w_j\}_{j=0}^{J-1}$ and define the ω_0 and ω_i's by

$$\begin{cases} \omega_0 = \tilde{w}_0 & \\ \omega_i = w_j & \text{for } i \in 2^j + \{0, ..., 2^j - 1\} \text{ and } j = 0, ..., J-1 \big(j = bin(i)\big), \end{cases}$$

where J is the maximum number of resolution levels. Finally, it suffices to compute the expression

$$\| Q, T \| = \tilde{w}_0 |\tilde{Q}[0] - \tilde{T}[0]| - \sum_{i:\tilde{Q}_q^c[i] \neq 0} w_{bin(i)} (\tilde{Q}_q^c[i] = \tilde{T}_q^c[i]). \tag{40}$$

4.2 One-dimensional image querying algorithm

In order to optimize the metric computation process, we introduce two arrays called search arrays. Let Θ_+ for the coefficients quantified to $+1$ and Θ_- for those which are quantified to -1. Each array contains $2^J - 1$ elements and each element contains a list. For example, the element $\Theta_+[i]$ points on the list of all the database (1D) images having a large positive wavelet coefficient at the i-th position, after compression. With the same way, the element $\Theta_-[i]$ points on the list of all the database (1D) images having a large negative wavelet coefficient at the i-th position. Thanks to these arrays and to the compression, during the querying process we have just to go through the m lists associated to the query m retained coefficients instead of $2^J - 1$ coefficients. After the creation of the search arrays and weights $\tilde{w}_0$ and $\{w_i\}_{i=0}^{J-1}$ computation, the retrieval procedure of a query Q in the database of (1D) images T_k ($k = 1, ..., |DB|$), where $|DB|$ denotes the database size, is defined as follows

Procedure Retrieval(Q: array $[1..2^J]$ of reals, m : integer,Θ_-,Θ_+)

```
    $\tilde{Q} \leftarrow$ FastDaubechiesWaveletsDecomposition($Q$)
    Initialize $Score$[k] $= 0$, for each $k \in \{1, ..., |DB|\}$
    **For each** $k \in \{1, ..., |DB|\}$ **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 $Score[k] = \parallel Q, T_k \parallel$ for each $k \in \{1, ..., |DB|\}$. The array *Score* elements which are the similarity degrees between the query Q and the database (1D) images T_k ($k \in \{1, ..., |DB|\}$), can be negative or positive. The most negative similarity degree corresponds to the closest target to the query Q.

4.3 Weights adjustment and logistic regression

The logistic regression is one of the most popular data mining tools. In content-based image retrieval field logistic regression was used to model the relevance feedback [8]. In our application, the logistic regression can be used to tune the weights of our metric. A weight w_k represents the corresponding relative importance of a query k-th resolution level coefficients, retained after the compression, to a target coefficients belonging to the same resolution level and having the same positions and signs. Note that if we set the weights $\tilde{w}_0$ and $\{w_j\}_{j=0}^{J-1}$ equal to 1, where J is the maximum number of resolution levels, then the target coefficients belonging to different resolution levels have the same corresponding relative importance to the query different resolution level coefficients. We devide our target set into two classes: a matching class and a mismatching class. Each class contains a set of observations extracted from the database. The logistic regression method introduces a binary target variable and a set of explanatory variables to represent the class of a given observation. Specifically

$$
\begin{aligned}
Y &: \text{ Target variable} \\
& \quad y_T = 1 \text{ case of mismatch with the target } T \\
& \quad\quad = 0 \text{ case of match with the target } T \\
\underline{t_T} &= (\tilde{t}_{0,T}, t_{0,T}, ..., t_{J-1,T}) : \text{Explanatory variables,}
\end{aligned}
$$

where $\tilde{t}_{0,T}$ is the absolute value of the difference between the scaling factors of the query and the target T, $\{t_{k,T}\}_{k=0}^{J-1}$ are the numbers of mismatches between the k-th resolution level coefficients of the query and the target T, J is the maximum number of the resolution levels and y_T is the binary target variable, it's either 0 or 1, depending on whether or not the query and the target T are intended to match. We assume a posterior probability of the mismatching with the target T, given the explanatory variables is p_T, i.e., $p_T = P(y_T|\underline{t_T})$. In our logistic regression model we assume that posterior probability is given by

$$
p_T = P\Big(y_T = 1|\underline{t_T}\Big) = F(\tilde{w}_0\tilde{t}_{0,T} + \sum_{k=0}^{J-1} w_k t_{k,T}), \tag{41}
$$

and

$$
\overline{p}_T = 1 - p_T = P\Big(y_T = 0|\underline{t_T}\Big) = F(-\tilde{w}_0\tilde{t}_{0,T} - \sum_{k=0}^{J-1} w_k t_{k,T}), \tag{42}
$$

where $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ are the weights to compute and

$$
F(x) = \frac{e^x}{1+e^x}. \tag{43}
$$

In order to tune the weights, we perform a maximum likelihood estimation. The likelihood function measures the likelihood that different $\underline{t_T}$ have given rise to the observed y_T. Provided observed target variables have independent Bernoulli distribution with the probabilities p_T for each target T, the form of the likelihood is given by

$$\begin{aligned} L(\tilde{w}_0, w_0, ..., w_{J-1}) &= \prod_{T=1}^{n} p_T^{y_T}(1-p_T)^{1-y_T} &(44)\\ &= \prod_{T=1}^{n_0}(1-p_T)\prod_{T=1}^{n_1} p_T, &(45)\end{aligned}$$

where n is the number of all observations and n_0 and n_1 are the numbers of cases with target variable value 0 and 1, respectively. We want to choose $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ so as to maximize the natural logarithm of the above likelihood. This can be done by using the software SAS 6.12 or the Fisher-scoring algorithm. The training is an important step in the logistic regression method to extract the observations from the database.

4.4 Training (for the logistic regression method)

Let us consider a color image database which consists of several color image sets such that each set contains color images which are perceptually close to each other in terms of object shapes and colors. In order to compute the metric weights $\tilde{w}_0^l$ and $\{w_k^l\}_{k=0}^{7}$ $(l \in \{1, ..., N\})$ by the logistic regression, we have to create the matching classes $(y_T^l = 0)$ $(l \in \{1, ..., N\})$ and the mismatching classes $(y_T^l = 1)$ $(l \in \{1, ..., N\})$. To create a single matching class $(y_T^l = 0)$, we draw all possible pairs of histograms or feature vectors representing color images belonging to the same database color image sets, and for each pair we compute the values of $\tilde{t}_{0,T}^l$ and $\{t_{k,T}^l\}_{k=0}^{J-1}$. Similarly, to create a single mismatching class $(y_T^l = 1)$, we draw all possible pairs of histograms or feature vectors representing color images belonging to different database color image sets, and then for each pair we compute the values of $\tilde{t}_{0,T}^l$ and $\{t_{k,T}^l\}_{k=0}^{J-1}$.

5 Color image retrieval method

The querying method is in two phases. The first phase is a pretreatment phase done once for the entire database containing $|DB|$ color images. The second phase is the querying phase.

5.1 Preprocessing (of the LAB color image database)

We detail the preprocessing phase done once for all the database color images before the querying in a general case by the following steps.

1. Choose N feature histograms for comparison.

2. Compute the N feature histograms T_{li} ($l \in \{1, ..., N\}$) for each i-th LAB color image of the database, where $i \in \{1, ..., |DB|\}$.

3. The feature histograms representing the database color images are Daubechies-8 wavelets decomposed, compressed to m coefficients each and quantified.

4. Organize the decomposed, compressed and quantified feature vectors into search arrays Θ^l_+ and Θ^l_- $(l = 1, ..., N)$.

5. Adjustment of the metric weights $\tilde{w}^l_0$ and $\{w^l_k\}^7_{k=0}$ for each set of feature histograms T_{li} $(i = 1, ..., |DB|)$ representing the database color images, where $l \in \{1, ..., N\}$.

5.2 The querying algorithm

We detail the querying algorithm in a general case by the following steps.

1. Given query LAB color image. We denote the feature vectors representing the query image by Q_l $(l = 1, ..., N)$.

2. The feature vectors representing the query image are Daubechies-8 wavelets decomposed, compressed to m coefficients each and quantified.

3. The similarity degrees between Q_l $(l = 1, ..., N)$ and the database color image feature histograms T_{li} $(l = 1, ..., N)$ $(i = 1, ..., |DB|)$ are represented by the arrays $Score_l$ $(l = 1, ..., N)$ such that $Score_l[i] = \| Q_l, T_{li} \|$ for each $i \in \{1, ..., |DB|\}$. These arrays are returned by the procedures Retrieval(Q_l, m, Θ^l_+, Θ^l_-) $(l = 1, ..., N)$, respectively.

4. The similarity degrees between the query color image and the database color images are represented by a resulted array $TotalScore$, such as, $TotalScore[i] = \sum_{l=1}^N \gamma_l Score_l[i]$ for each $i \in \{1, ..., |DB|\}$, where $\{\gamma_l\}_{l=1}^N$ are weightfactors used to fine-tune the influence of each individual feature.

5. Organize the database color images in order of increasing resulted similarity degrees of the array $TotalScore$. The most negative resulted similarity degrees correspond to the closest target images to the query image. Finally, return to the user the closest target color images to the query color image and whose number is denoted by RI and chosen by the user.

5.3 The querying method dataflow diagram

For simplicity, we describe the querying method steps by the following diagram.

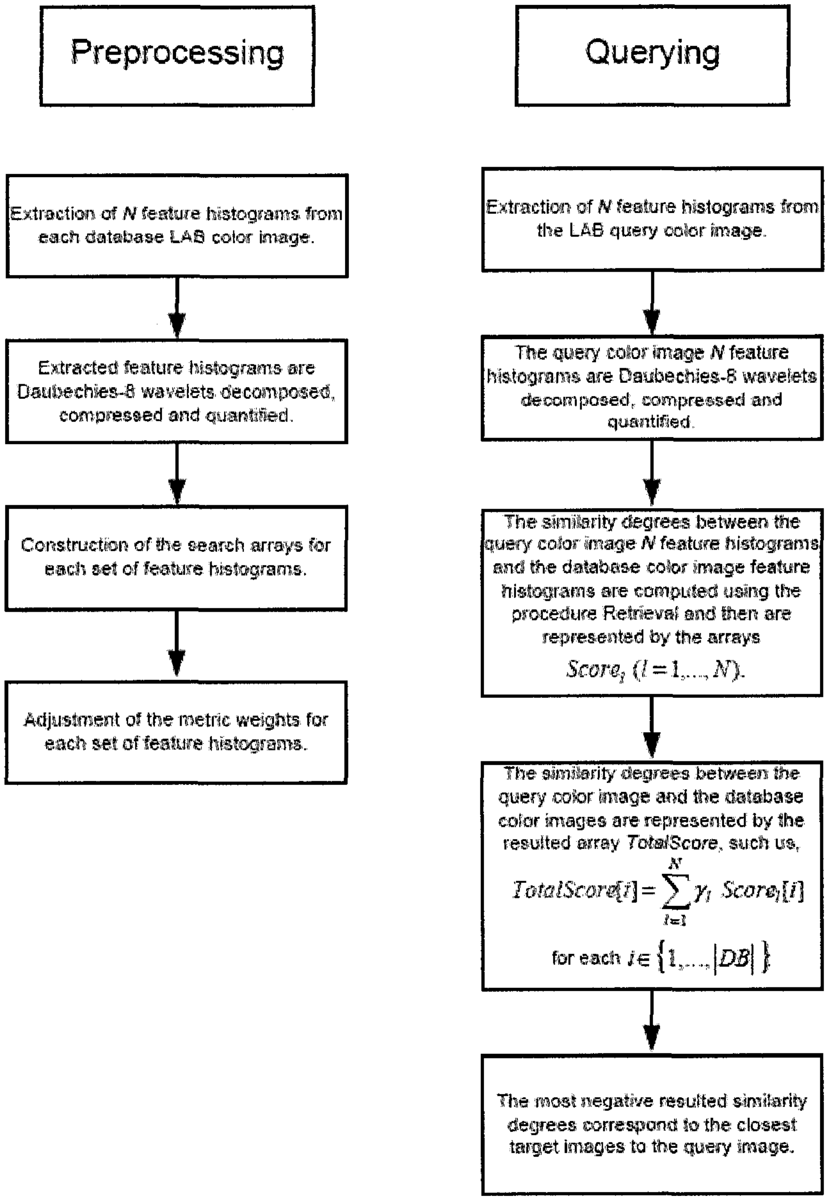

Preprocessing
Querying
Extraction of *N* feature histograms from each database LAB color image.
Extracted feature histograms are Daubechies-8 wavelets decomposed, compressed and quantified.
Construction of the search arrays for each set of feature histograms.
Adjustment of the metric weights for each set of feature histograms.
Extraction of *N* feature histograms from the LAB query color image.
The query color image *N* feature histograms are Daubechies-8 wavelets decomposed, compressed and quantified.
The similarity degrees between the query color image *N* feature histograms and the database color image feature histograms are computed using the procedure Retrieval and then are represented by the arrays $Score_l \ (l = 1, \dots, N)$.
The similarity degrees between the query color image and the database color images are represented by the resulted array *TotalScore*, such us,
$TotalScore[i] = \sum_{l=1}^{N} \gamma_l \ Score_l[i]$
for each $i \in \{1, \dots, |DB|\}$
The most negative resulted similarity degrees correspond to the closest target images to the query image.

Figure 2: Block diagram of the querying method.

6 Experimental results and evaluation

In this section we perform some experiments to validate and evaluate our querying method in a database of $|DB| = 10000$ animal, landscape, art, bridge and building LAB color images. It consists of several color image sets such that each set contains color images which are perceptually close to each other in terms of object shapes and colors.

The first evaluation will be based on a comparison between the querying after using classical color histograms h_L, h_a and h_b given by (1), and the querying after using h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ given by (1), (9) and (12), respectively, to represent the database color images. The second evaluation will be based on a comparison between the querying results when the metric weights are equal to 1 and the querying results when the metric weights are computed by the logistic regression. In this evaluation each database color image is represented by its h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$. The third evaluation will be based on a comparison between the querying after using h_L, h_a^h, h_b^h given by (1) and (9), respectively, and the laplacian weighted histograms proposed by [6], and the querying after using h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$, to represent the database color images. In the fourth evaluation we will show the importance of considering the metric term $\tilde{w}_0|\tilde{Q}[0] - \tilde{T}[0]|$ or the scaling factors in the querying. Finally, the fifth evaluation is carried out to show the improvement of the querying results thanks to the query color image $\bar{h}_a^e$ and $\bar{h}_b^e$ transformation used to make the querying invariant to the color intensities of the query color image.

Generally, to carry out an evaluation in the image retrieval field, two principal issues are required: the acquisition of ground truth and the definition of performance criteria. For ground truth, we introduce human observations. In fact, two external persons participate in the all below evaluations. Concerning performance criteria, we represent each evaluation results by the precision-scope curve $Pr = f(RI)$, where the scope RI is the number of images returned to the user. In each querying performed in an evaluation experiment, each human subject is asked to give a goodness score to each retrieved image. The goodness scores are 2 if the retrieved image is almost similar to 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. Consequently, we can compute the precision as follows: $Pr =$ the sum of goodness scores for retrieved images$/RI$. Therefore, the curve $Pr = f(RI)$ gives the precision for different values of RI which lie between 1 and 10 in our evaluation experiments. When the human subjects perform different queryings in the same evaluation experiment, we compute an average precision for each value of RI, and then we construct the precision-scope curve.

The following experiments will be carried out to extract the curve $Pr = f(RI)$ for each evaluation mentioned above.

First experiment

This experiment is carried out to show the advantage of using our weighted histograms instead of classical color histograms to represent the query color image before the querying. Each human subject is asked to formulate a query from the database and to execute a querying using $N = 3$ feature histograms which are h_L, h_a and h_b given by (1), to represent the query color image, while computing the metric weights by the logistic regression and keeping the weightfactors $\{\gamma_l\}_{l=1}^{3}$ equal to 1, and to give a goodness score to each retrieved image, then to reformulate a query from the database and to execute a querying using $N = 5$ feature histograms which are h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$, to represent the query color image, while computing the metric weights by the logistic regression and keeping the weightfactors $\{\gamma_l\}_{l=1}^{3}$ equal to $\frac{1}{2}$ and γ_4 and γ_5 equal to 1 to give more importance to the edge region features, and to give a goodness score to each retrieved image. Each querying is repeated twenty times by choosing a new query from the database each time. We repeat this experiment for different orders of compression $m \in \{30, 20, 10\}$. The resulted precision-scope curves are

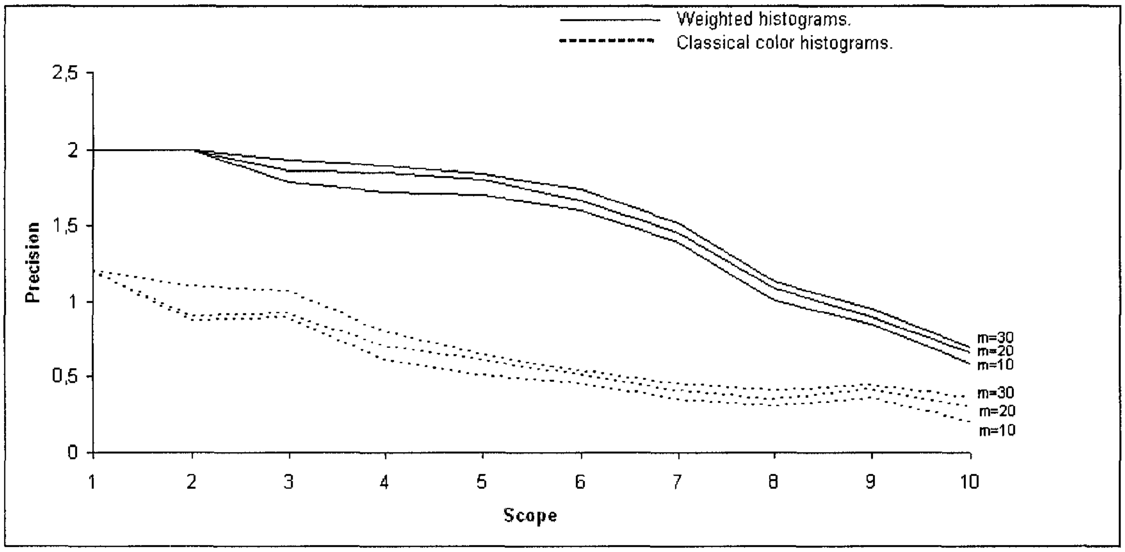
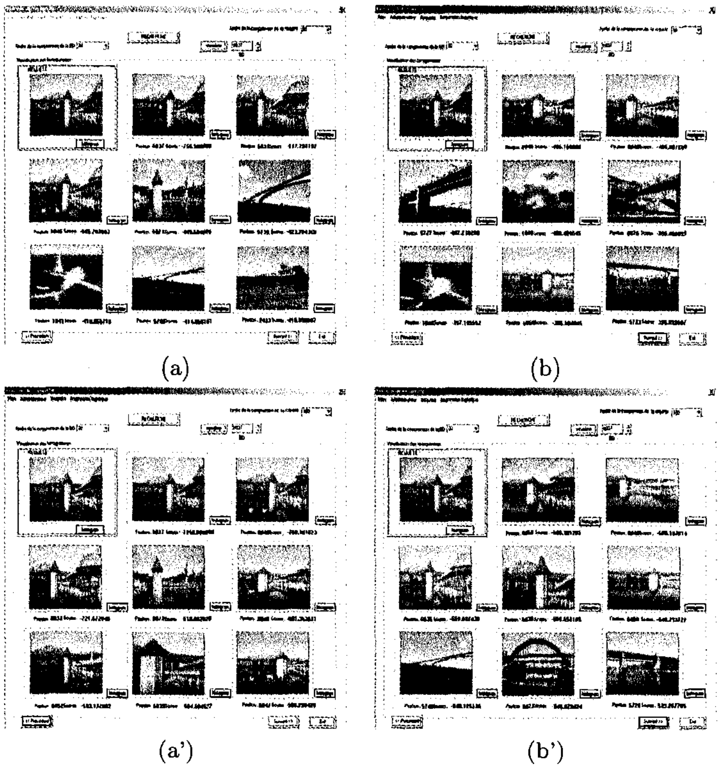


Figure 3: Evaluation: precision-scope curves for retrieval when the database LAB color images are represented by h_L, h_a and h_b each, and when the database LAB color images are represented by h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, for the compression orders $m \in \{30, 20, 10\}$.

The following three Figures give three examples of the improvements of our querying results thanks to the separation between the modes of the color histograms representing the database LAB color images and to the use of the multispectral gradient module mean histogram to consider the database color image edge regions in the querying. We choose the same query for the three examples. For each example the query is located at the top-left of the dialog box.


(a)

(b)

(a')

(b')

Figure 4: Comparison ($m = 30$): a) first 7 color images retrieved after being represented by their h_L, h_a and h_b, b) second 7 color images retrieved after being represented by their h_L, h_a and h_b, a') first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ and b') second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$.

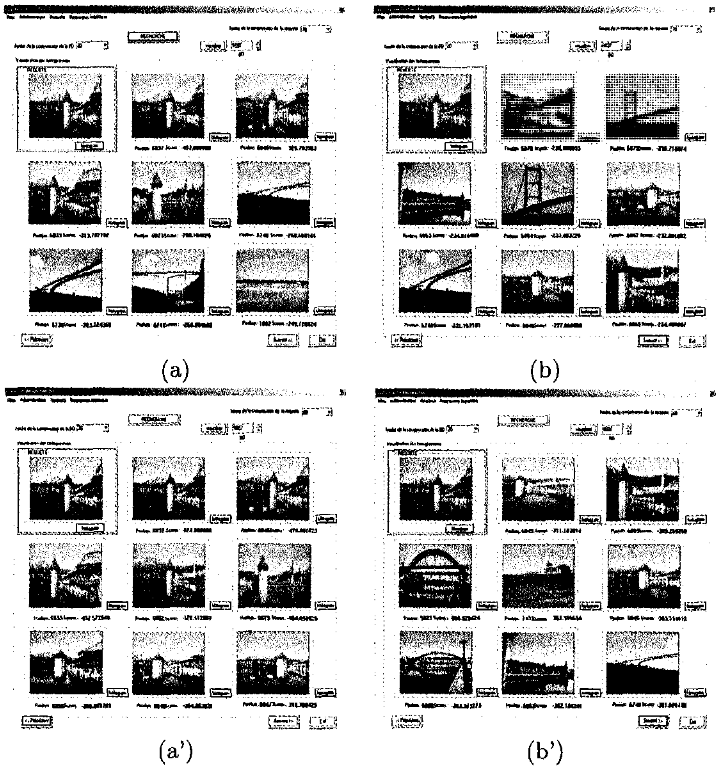
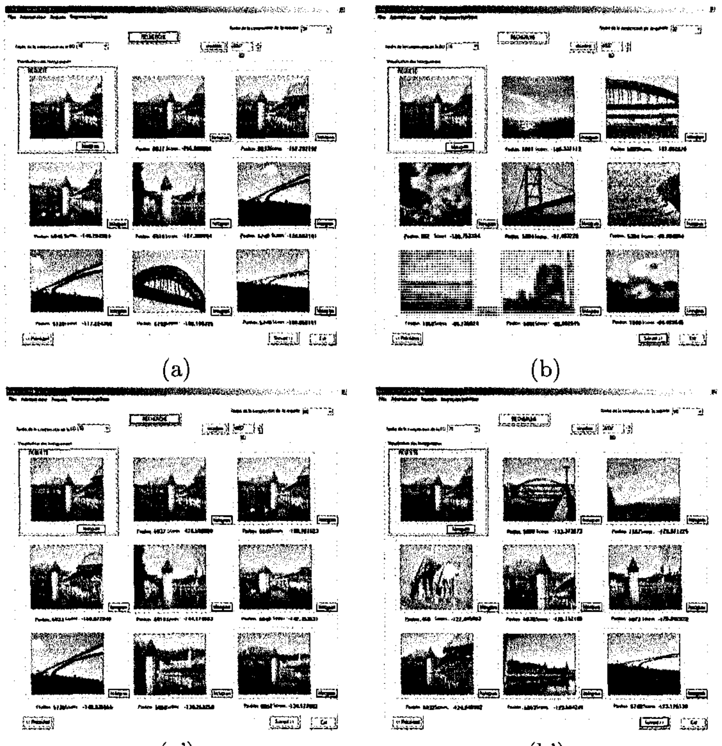


Figure 5: Comparison ($m = 20$): a) first 7 color images retrieved after being represented by their h_L, h_a and h_b, b) second 7 color images retrieved after being represented by their h_L, h_a and h_b, a') first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ and b') second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$.



(a) (b)

(a') (b')

Figure 6: Comparison ($m = 10$): a) first 7 color images retrieved after being represented by their h_L, h_a and h_b, b) second 7 color images retrieved after being represented by their h_L, h_a and h_b, a') first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ and b') second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$.

Second experiment

This experiment is carried out to show how the weights computed by the logistic regression can optimize our querying results. Each database color image is represented by h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$. Each human subject is asked to formulate a query from the database and to execute a querying, using weights computed by the logistic regression, and to give a goodness score to each retrieved image, then to reformulate a query from the database and to execute the querying, using weights equal to 1, and to give a goodness score to each retrieved image. Each querying is repeated twenty times by choosing a new query from the database each time. We repeat this experiment for different orders of compression $m \in \{30, 20, 10\}$ and we keep the weightfactors $\{\gamma_l\}_{l=1}^3$ equal to $\frac{1}{2}$ and γ_4 and γ_5 equal to 1 to give more importance to the edge region features. The following table contains the weights computed by the logistic regression, for order $m = 30$ of the compression.

	$\tilde{w}_0^l$	w_0^l	w_1^l	w_2^l	w_3^l	w_4^l	w_5^l	w_6^l	w_7^l
$h_L(l=1)$	3.31	7.23	5.34	6.85	8.51	8.37	6.24	7.25	10.14
$h_a^h(l=2)$	3.59	4.65	6.28	8.26	4.38	6.37	7.21	10.41	3.91
$h_b^h(l=3)$	5.14	8.29	3.51	4.40	8.59	5.90	8.72	7.83	6.93
$h_a^e(l=4)$	5.78	5.67	8.52	7.04	6.85	8.01	7.09	6.82	6.62
$h_b^e(l=5)$	5.63	8.08	6.18	9.37	9.47	5.05	7.57	8.43	7.94

Table 1: Weights computed by the logistic regression for $m = 30$.

The resulted precision-scope curves for each compression order are

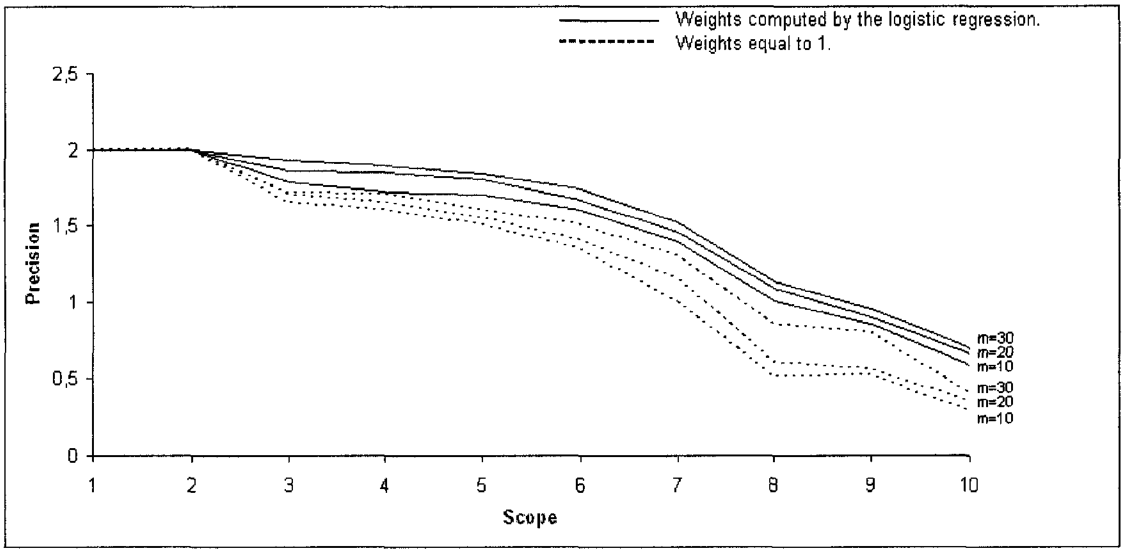
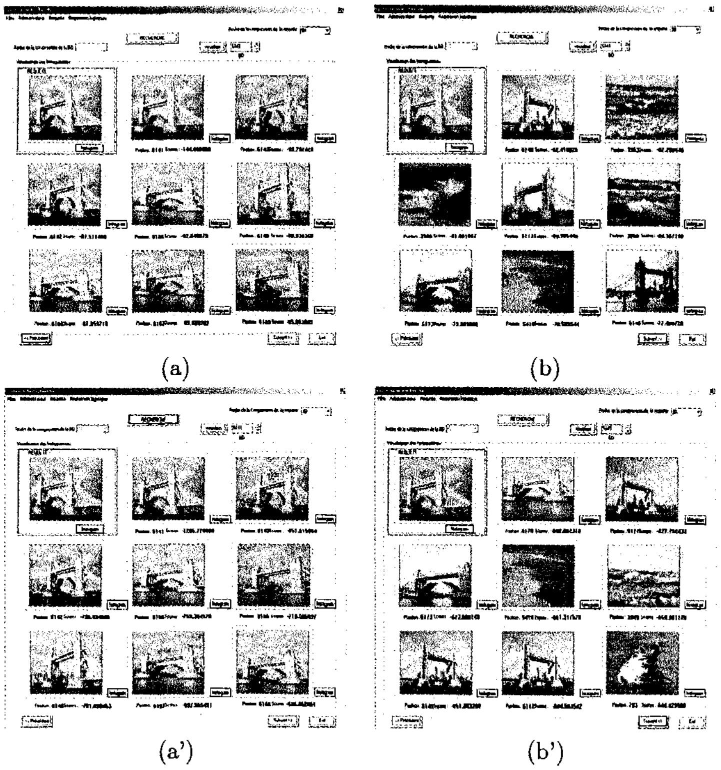


Figure 7: Evaluation: precision-scope curves for retrieval using weights equal to 1 and weights computed by the logistic regression, for the compression orders $m \in \{30, 20, 10\}$.

The following three Figures give three examples of the improvements of our querying results when we use weights computed by the logistic regression instead of weights equal to 1, to tune the metric. We choose the same query for the three examples. For each example the query is located at the top-left of the dialog box.



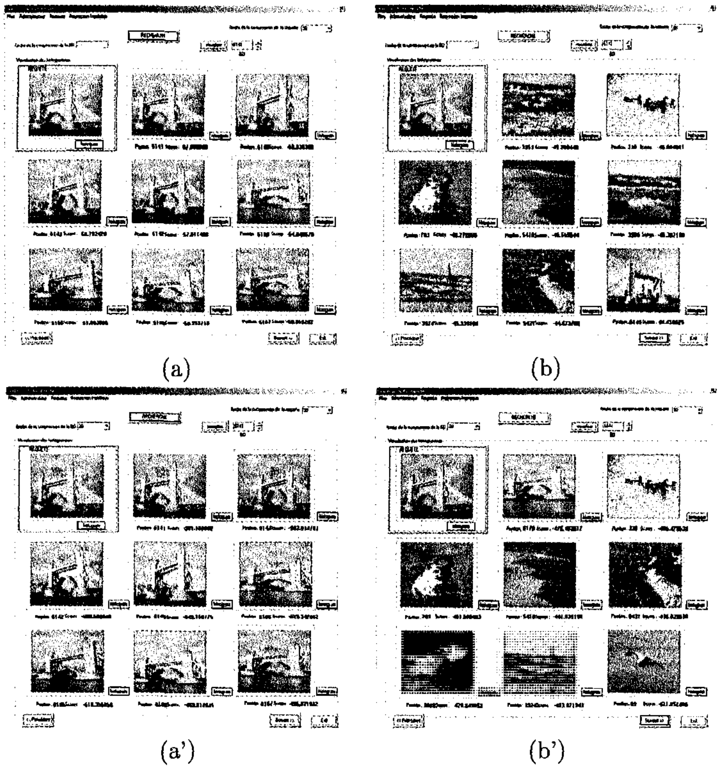
(a)

(b)

(a')

(b')

Figure 8: Comparison ($m = 30$): a) first 7 color images retrieved after using weights equal to 1, b) second 7 color images retrieved after using weights equal to 1, a') first 7 color images retrieved after using weights computed by the logistic regression and b') second 7 color images retrieved after using weights computed by the logistic regression.



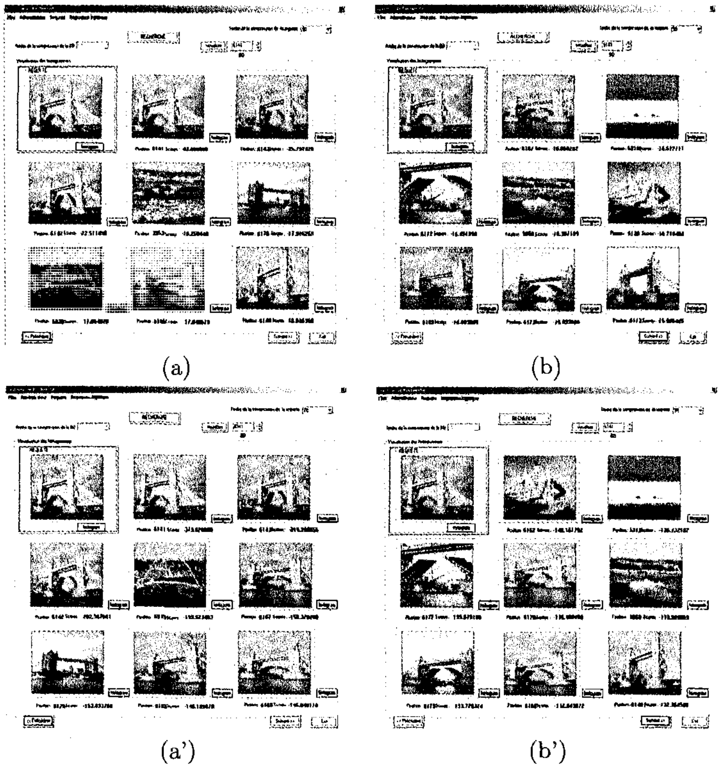
(a)

(b)

(a')

(b')

Figure 9: Comparison ($m = 20$): a) first 7 color images retrieved after using weights equal to 1, b) second 7 color images retrieved after using weights equal to 1, a') first 7 color images retrieved after using weights computed by the logistic regression and b') second 7 color images retrieved after using weights computed by the logistic regression.



(a) (b)

(a') (b')

Figure 10: Comparison ($m = 10$): a) first 7 color images retrieved after using weights equal to 1, b) second 7 color images retrieved after using weights equal to 1, a') first 7 color images retrieved after using weights computed by the logistic regression and b') second 7 color images retrieved after using weights computed by the logistic regression.

Third experiment

This experiment is carried out to show the advantage of using $\bar{h}_a^e$ and $\bar{h}_b^e$ instead of the laplacian weighted histograms to represent the query color image before the querying. Each human subject is asked to formulate a query from the database and to execute a querying using $N = 5$ feature histograms which are h_L, h_a^h, h_b^h and the laplacian weighted histograms, to represent the query color image, and to give a goodness score to each retrieved image, then to reformulate a query from the database and to execute a querying using $N = 5$ feature histograms which are h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$, to represent the query color image, and to give a goodness score to each retrieved image. Each querying is repeated twenty times by choosing a new query from the database each time. We repeat this experiment for different orders of compression $m \in \{30, 20, 10\}$, we compute the metric weights $\tilde{w}_0^l$ and $\{w_k^l\}_{k=0}^7$ by the logistic regression for each $l \in \{1, 2, 3, 4, 5\}$, and we keep the weightfactors $\{\gamma_l\}_{l=1}^3$ equal to $\frac{1}{2}$ and γ_4 and γ_5 equal to 1 to give more importance to the edge region features. The resulted precision-scope curves are

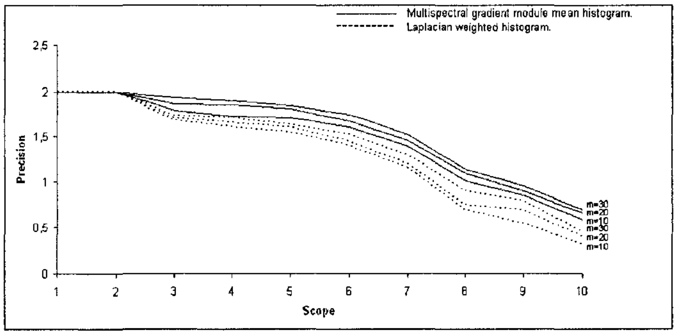
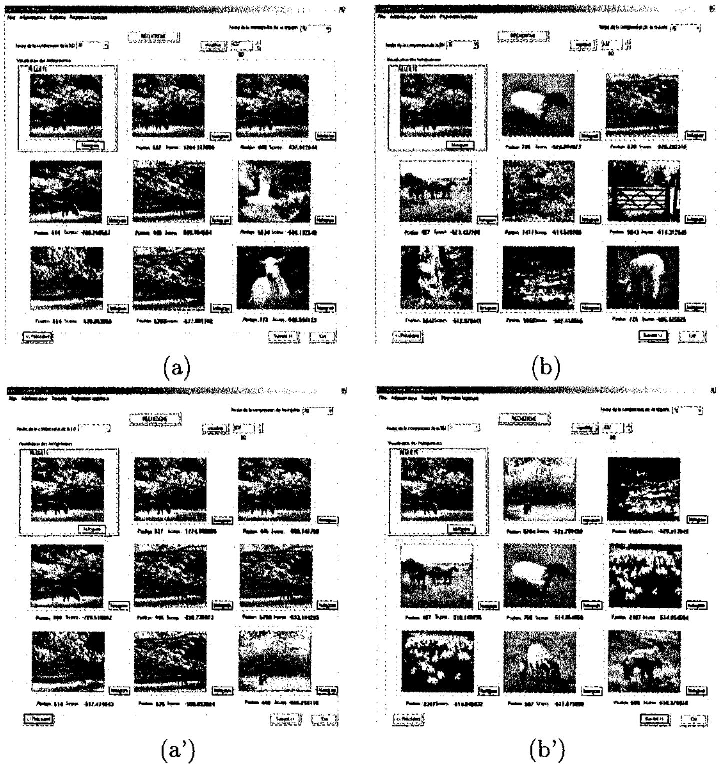


Figure 11: Evaluation: precision-scope curves for retrieval in database of LAB color images represented by h_L, h_a^h, h_b^h and the laplacian weighted histograms each, and for retrieval in the same database LAB color images represented by h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, for different compression orders $m \in \{30, 20, 10\}$.

The following three Figures give three examples of improvement of our querying results when we represent each query image by $\bar{h}_a^e$ and $\bar{h}_b^e$ instead of the laplacian weighted histograms. We choose the same query for the three examples. For each example the query is located at the top-left of the dialog box.



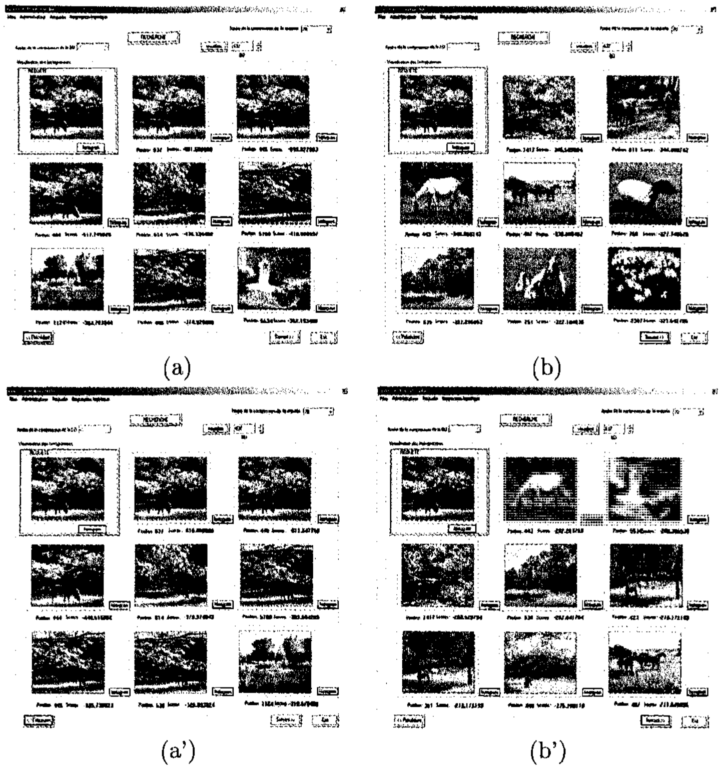
(a)

(b)

(a')

(b')

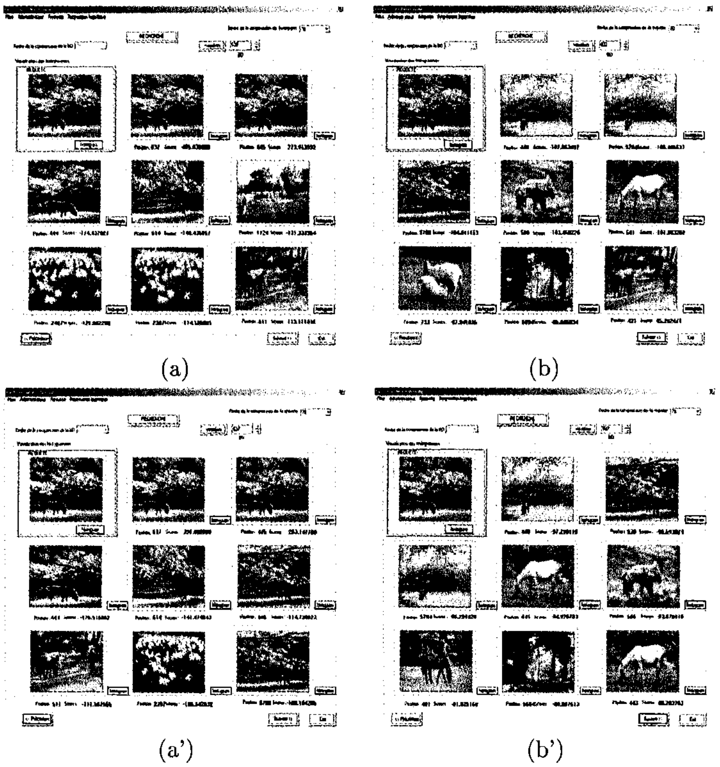
Figure 12: Comparison ($m = 30$): a) first 7 color images retrieved after being represented by their laplacian weighted histograms each, b) second 7 color images retrieved after being represented by their laplacian weighted histograms each, a') first 7 color images retrieved after being represented by their $\bar{h}_a^e$ and $\bar{h}_b^e$ each and b') second 7 color images retrieved after being represented by their $\bar{h}_a^e$ and $\bar{h}_b^e$ each.



(a) (b)

(a') (b')

Figure 13: Comparison ($m = 20$): a) first 7 color images retrieved after being represented by their laplacian weighted histograms each, b) second 7 color images retrieved after being represented by their laplacian weighted histograms each, a') first 7 color images retrieved after being represented by their $\bar{h}_a^e$ and $\bar{h}_b^e$ each and b') second 7 color images retrieved after being represented by their $\bar{h}_a^e$ and $\bar{h}_b^e$ each.



(a) (b)

(a') (b')

Figure 14: Comparison ($m = 10$): a) first 7 color images retrieved after being represented by their laplacian weighted histograms each, b) second 7 color images retrieved after being represented by their laplacian weighted histograms each, a') first 7 color images retrieved after being represented by their $\bar{h}_a^e$ and $\bar{h}_b^e$ each and b') second 7 color images retrieved after being represented by their $\bar{h}_a^e$ and $\bar{h}_b^e$ each.

Fourth experiment

This experiment is carried out to show the importance of considering the scaling factors of the Daubechies-8 decomposed versions of the histograms $\bar{h}_a^e$ and $\bar{h}_b^e$ in the querying. Each human subject is asked to formulate a query from the database and to execute a querying using $N = 5$ feature histograms which are h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ to represent the query color image, while computing the metric weights $\tilde{w}_0^l$ and $\{w_k^l\}_{k=0}^7$ by the logistic regression for each $l \in \{1, 2, 3, 4, 5\}$, and to give a goodness score to each retrieved image, then to reformulate a query from the database and to execute a querying using the same feature histograms, while keeping the metric terms $\tilde{w}_0|\tilde{Q}_4[0] - \tilde{T}_4[0]|$ and $\tilde{w}_0|\tilde{Q}_5[0] - \tilde{T}_5[0]|$ equal to zero by affecting a zero to $\tilde{w}_0^4$ and $\tilde{w}_0^5$, and to give a goodness score to each retrieved image. Each querying is repeated twenty times by choosing a new query from the database each time. We repeat this experiment for different orders of compression $m \in \{30, 20, 10\}$ we keep the weightfactors $\{\gamma_l\}_{l=1}^3$ equal to $\frac{1}{2}$ and γ_4 and γ_5 equal to 1 to give more importance to the edge region features. The resulted precision-scope curves for each compression order are

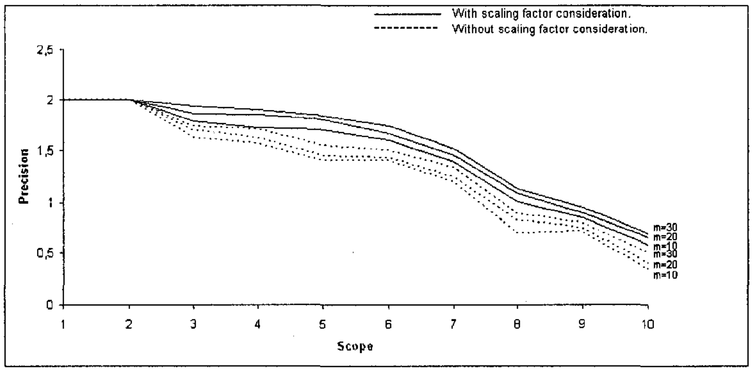
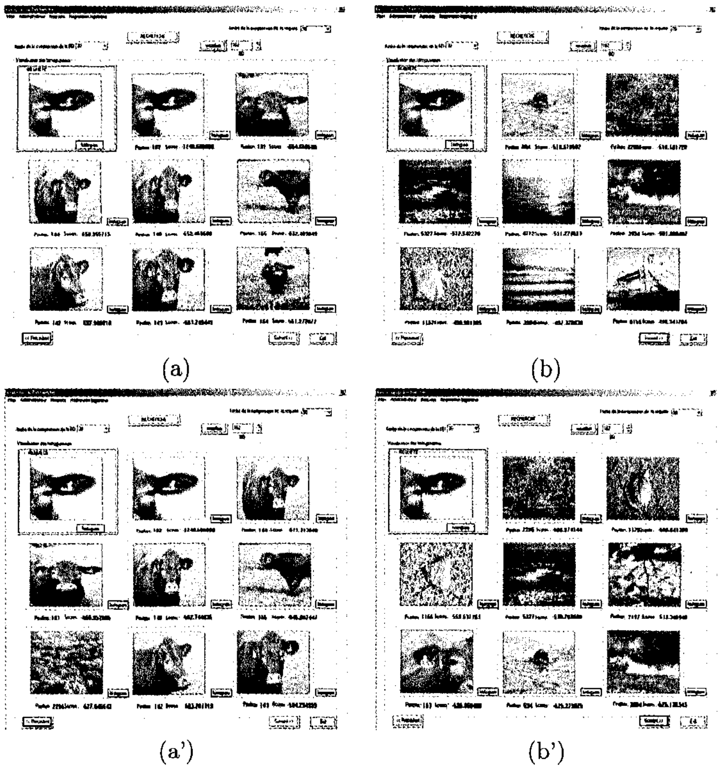


Figure 15: Evaluation: precision-scope curves for retrieval after considering the scaling factors and after neglecting these latters , for the compression orders $m \in \{30, 20, 10\}$.

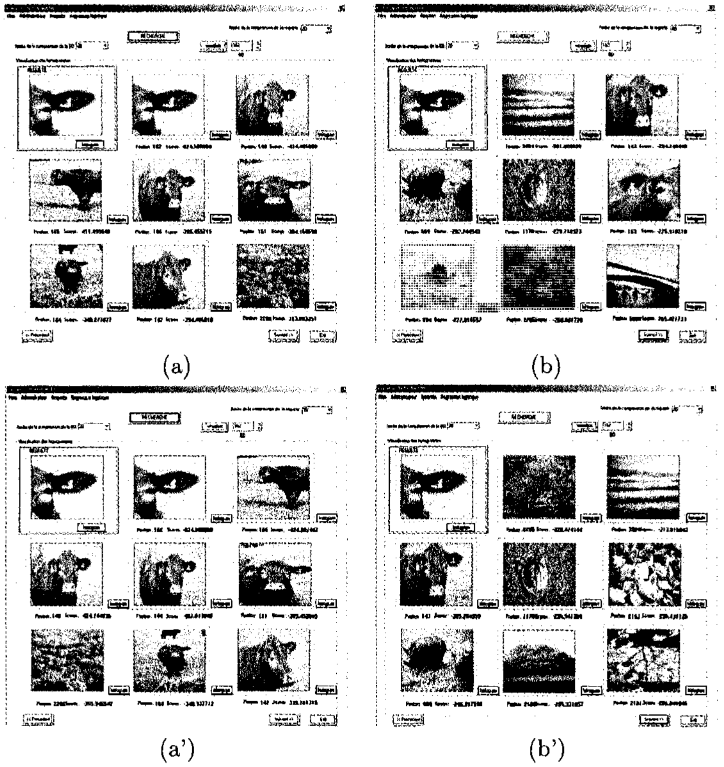
Thanks to the above precision-scope curves we can notice the degradation of the querying when we neglect the scaling factors. In fact, because of the quantization if we keep the metric terms $\tilde{w}_0|\tilde{Q}_4[0] - \tilde{T}_4[0]|$ and $\tilde{w}_0|\tilde{Q}_5[0] - \tilde{T}_5[0]|$ equal to zero by affecting a zero to $\tilde{w}_0^4$ and $\tilde{w}_0^5$, the metric will not distinguish two color component multispectral gradient module mean histograms Q_4 and T_4 or Q_5 and T_5 having the same curvature variations at the same X-coordinates, but these curvatures have different magnitudes. Consequently, the metric can not distinguish two LAB color images having almost similar colors and luminances, but different object shapes. For this reason, it's very important to consider the scaling factor of each decomposed multispectral gradient module mean histogram of a LAB color image component in the querying. In fact, this scaling factor represents the average of the overall gradient module values of each LAB color image component, and changes from image to other having different edge shapes. The following three Figures give three examples of the degradation of our querying results when we don't consider the scaling factors of the Daubechies-8 decomposed versions of the histograms $\bar{h}_a^e$ and $\bar{h}_b^e$. We choose the same query for the three examples. For each example the query is located at the top-left of the dialog box.



(a) (b)

(a') (b')

Figure 16: Comparison ($m = 30$): a) first 7 color images retrieved after considering the scaling factors, b) second 7 color images retrieved after considering the scaling factors, a') first 7 color images retrieved after neglecting the scaling factors and b') second 7 color images retrieved after neglecting the scaling factors.



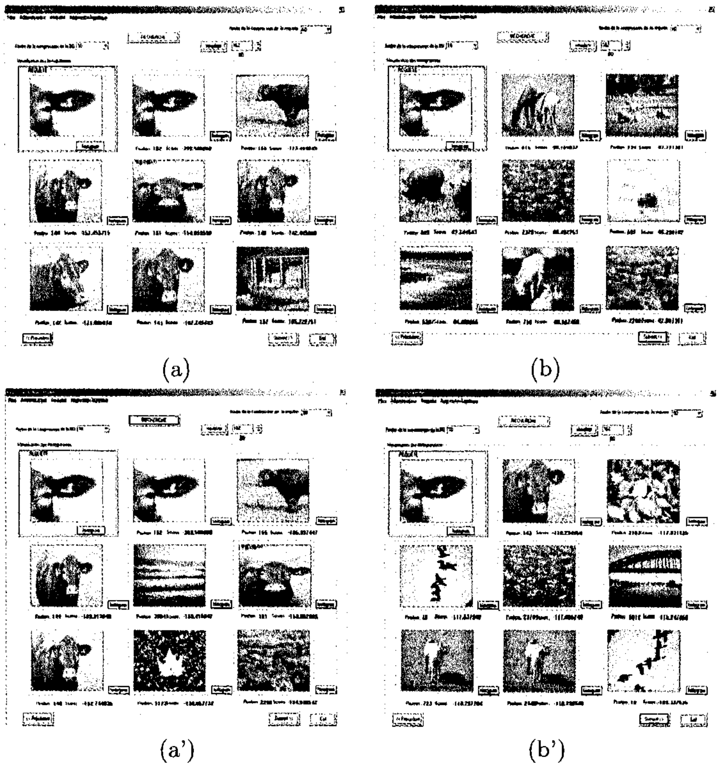
(a)

(b)

(a')

(b')

Figure 17: Comparison ($m = 20$): a) first 7 color images retrieved after considering the scaling factors, b) second 7 color images retrieved after considering the scaling factors, a') first 7 color images retrieved after neglecting the scaling factors and b') second 7 color images retrieved after neglecting the scaling factors.


(a) (b)

(a') (b')

Figure 18: Comparison ($m = 10$): a) first 7 color images retrieved after considering the scaling factors, b) second 7 color images retrieved after considering the scaling factors, a') first 7 color images retrieved after neglecting the scaling factors and b') second 7 color images retrieved after neglecting the scaling factors.

Fifth experiment

The metric is very sensitive to the translation. Therefore, it can not distinguish two color images having similar objects, but different background and object colors, which is a big constraint for the querying. In fact, we assume that these two color image multispectral gradient module mean histograms have similar mode shapes and magnitudes, but these modes are translated to each other because of the color difference. In order to solve this problem, we compute the mean of each database color image multispectral gradient module mean histograms, and then we shift these latters until that their mean pixels match to their center pixels which correspond to the X-coordinates 128 in our case. However, because of this shift, the multispectral gradient module mean histograms can exceed their supports, which affects their scaling factor values after their Daubechies-8 wavelet decompositions, and then affects the querying results. Consequently, in order to preserve the same scaling factor values, we double the resolution of each histogram to 512 pixels, we perform the shift, and then we reconstruct periodically each histogram for the 256 remaining pixels. The mean of a LAB color image multispectral gradient module mean histogram $\bar{h}_k^e$ having 2^J pixels is given by

$$m_{\bar{h}_k^e} = \frac{\sum_{i=0}^{2^J-1} i\bar{h}_k^e(i)}{\sum_{i=0}^{2^J-1} \bar{h}_k^e(i)}, \tag{46}$$

for each LAB color component $k = a, b$. We denote the transformed versions of the multispectral gradient module mean histograms $\bar{h}_a^e$ and $\bar{h}_b^e$ are denoted by $\bar{h}_{a_T}^e$ and $\bar{h}_{b_T}^e$, respectively. This last experiment is carried out to show how the $\bar{h}_a^e$ and $\bar{h}_b^e$ transformation used to make the querying invariant to the color intensities of the query color image, improve the querying results. Each human subject is asked to formulate a query from the database and to execute a querying, using $N = 5$ feature histograms which are h_L, h_a^h, h_b^h, $\bar{h}_{a_T}^e$ and $\bar{h}_{b_T}^e$, to represent the query color images, and to give a goodness score to each retrieved image, then to reformulate a query from the database and to execute a querying, using $N = 5$ feature histograms which are h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$, to represent the query color image, and to give a goodness score to each retrieved image. Each querying is repeated twenty times by choosing a new query each time. We repeat this experiment for different orders of compression $m \in \{30, 20, 10\}$, we compute the metric weights by the logistic regression and we keep the weightfactors $\{\gamma_l\}_{l=1}^3$ equal to $\frac{1}{2}$ and γ_4 and γ_5 equal to 1 to give more importance to the edge region features. The resulted precision-scope curves for each compression order are

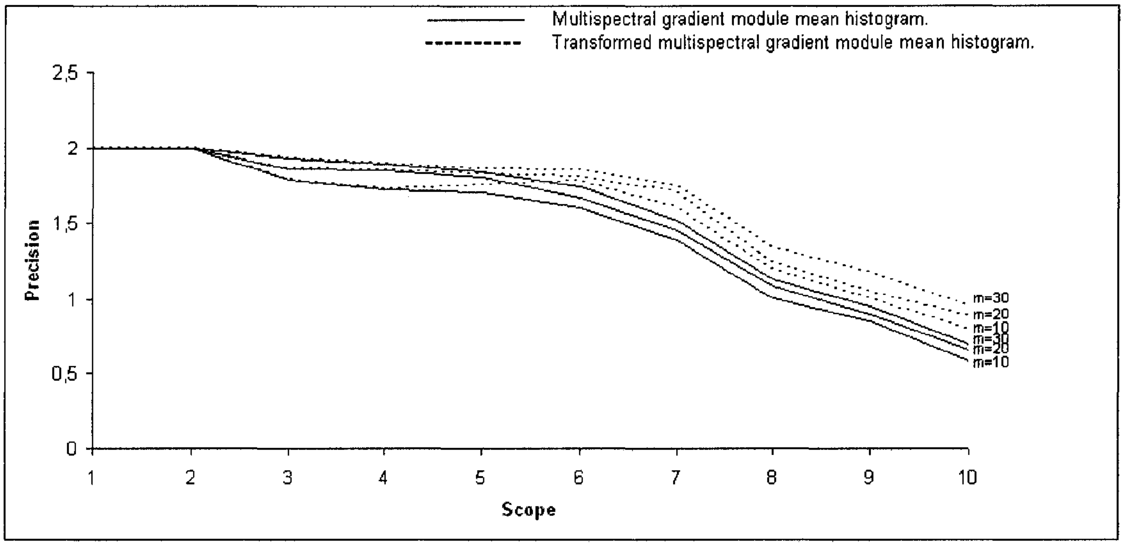
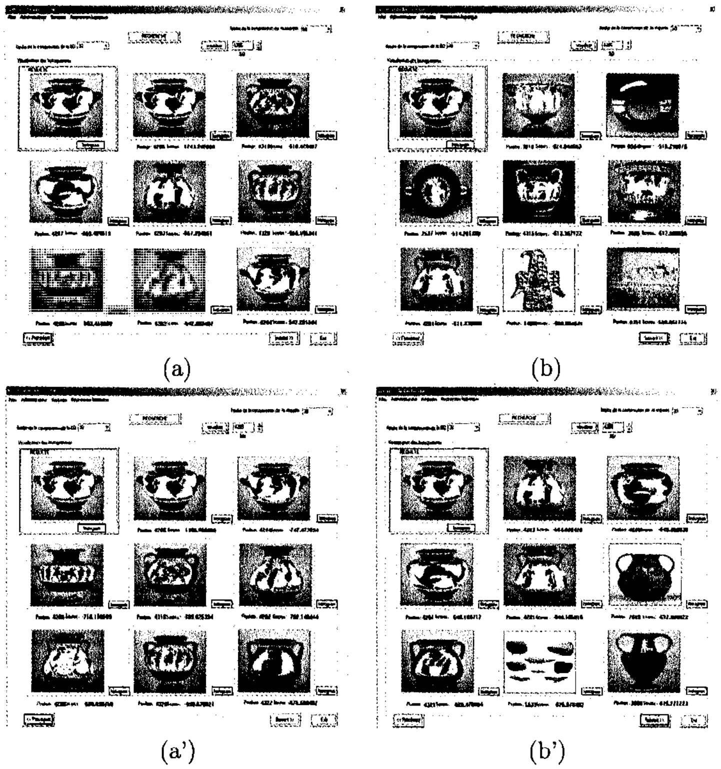


Figure 19: Evaluation: precision-scope curves for retrieval in database of LAB color images represented by h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, and for retrieval in the same database of LAB color images represented by h_L, h_a^h, h_b^h, $\bar{h}_{aT}^e$ and $\bar{h}_{bT}^e$.

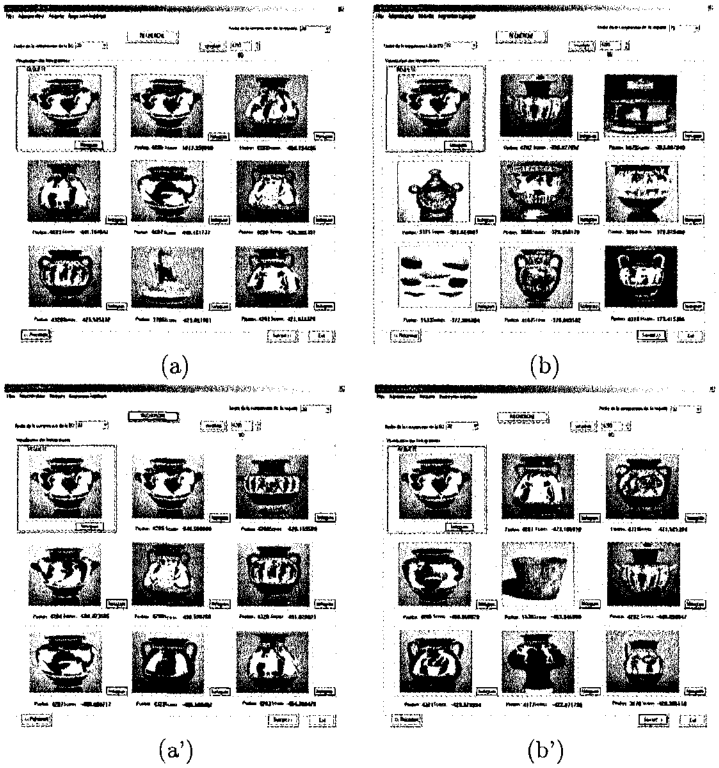
The following three Figures give three examples of the improvement of our querying results when we represent each database LAB color image by h_L, h_a^h, h_b^h, $\bar{h}_{aT}^e$ and $\bar{h}_{bT}^e$. We choose the same query for the three examples. For each example the query is located at the top-left of the dialog box



(a) (b)

(a') (b')

Figure 20: Comparison ($m = 30$): a) first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, b) second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, a') first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_{a_T}^e$ and $\bar{h}_{b_T}^e$ each and b') second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_{a_T}^e$ and $\bar{h}_{b_T}^e$ each.



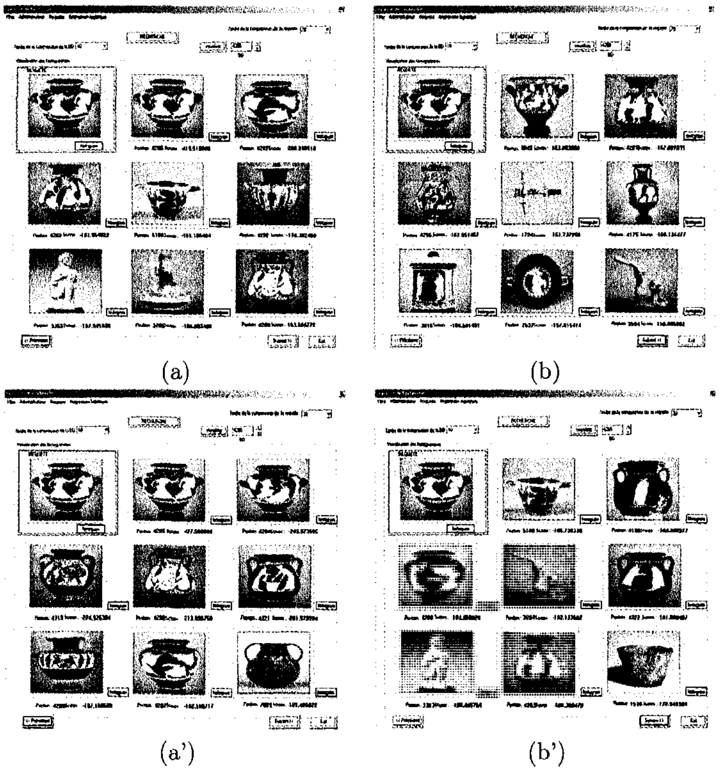
(a)

(b)

(a')

(b')

Figure 21: Comparison ($m = 20$): a) first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, b) second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, a') first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_{a\tau}^e$ and $\bar{h}_{b\tau}^e$ each and b') second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_{a\tau}^e$ and $\bar{h}_{b\tau}^e$ each.


(a)

(b)

(a')

(b')

Figure 22: Comparison ($m = 10$): a) first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, b) second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_a^e$ and $\bar{h}_b^e$ each, a') first 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_{a\tau}^e$ and $\bar{h}_{b\tau}^e$ each and b') second 7 color images retrieved after being represented by their h_L, h_a^h, h_b^h, $\bar{h}_{a\tau}^e$ and $\bar{h}_{b\tau}^e$ each.

7 Conclusion

We presented a simple, fast and effective querying method. In this method we improved the spatial information combination with the colors given by the laplacian weighted histogram by introducing a multispectral gradient module mean histogram. Also, we showed that through the use of the one-dimensional metric proposed by [5] and through a one-dimensional Daubechies-8 decomposition and compression of color image feature vectors, we could make a good compromise between the querying computational complexity and effectiveness. In order to represent our color images we used the LAB color space, because it's perceptually uniform and it allows a good separation between the colors and the luminance. Thanks to this separation the color is represented by just two components and then each color image feature is represented by two histograms instead of three, which is the case for other color spaces, like the RGB color space. The standard logistic regression is a good tool to compute the metric weights and to improve the querying results. However, it can provide inaccurate weights when the number of observations is very large. For this reason, in the future work we will try an other statistical method in order to improve the discriminatory capacity of the metric when we have larger color image databases.

References

[1] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L Yeo. Wavelet transforms that map integers to integers. *Appl. and Comput. Harmonic Analysis*, 5(3):332–369, 1998.

[2] A. Smeulder, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE PAMI*, 22:1349–1380, 2000.

[3] R. S. Berns. *Billmeyer and Saltzman's Principles of Color Technology.* Wiley-Interscience, New York, 2000.

[4] C. Drewniok. Multi-Spectral Edge Detection. Some Experiments on Data from Landsat-TM. *International Journal of Remote Sensing*, 15(18):3743–3765, 1994.

[5] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *in Proc. SIGGRAPH Computer Graphics*, pages 278–280, Los Angeles, CA, 1995.

[6] C. Vertan and N. Boujemaa. Upgrading Color Distributions for Image Retrieval: can we do better?. In *International Conference on Visual Information Systems*, Lyon, France, 2000.

[7] D. Malcara. *Color Vision and Colorimetry, Theory and Applications.* SPIE Press, Bellingham, Washington, USA, 2002.

[8] G. Caenen and E. J. Pauwels. Logistic regression models for relevance feedback in content-based Image Retrieval. *Proceedings of SPIE, Storage and Retrieval for Media Databases 2002*, 4676:49–58, 2002.

[9] G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. *In IEEE Workshop on Applications of Computer Vision*, pages 96–102, 1996.

[10] G. Pass and R. Zabih. Comparing images using joint histograms. *In Journal of Multimedia Systems*, 7(3):234–240, 1999.

[11] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae.* Wiley-Interscience, New York, 1982.

[12] H. Chidiac and D. Ziou. Classification of Image Edges. In *Vision Interface*, Trois-Rivières, Canada, 1999.

[13] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu and R. Zabih. Spatial Color Indexing and Applications. *International Journal of Computer Vision*, 35(3):245–268, 1999.

[14] J. Z. Wang, G. Wiederhold, O. Firschein, and X. W. Sha. Content-Based Image Indexing and Searching Using Daubechies' Wavelets. *International Journal on Digital Libraries*, 1 (4):311–328, 1998.

[15] K.-C. Liang and C.-C. J. Kuo. WaveGuide: a joint wavelet-based image representation and description system . *IEEE Transactions on Image Processing*, 8 (11):1619–1629, 1998.

[16] M. J. Swain and D. H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[17] N. Khelil and A. Benazza-Benyahia. Fast Scalable Wavelet-Based Retrieval of Multispectral Images. In *International Symposium on Image/Video Communications over fixed and mobile networks*, Brest,France, 2004.

[18] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical Evaluation of Dissimilarity Measures for Color and Texture. *Computer Vision and Image Understanding Journal*, 84 (1):25–43, 2001.

[19] Y. Rui, T. S. Huang, and S.-F. Chang. Image Retrieval: Current Techniques, Promising Directions and Open Issues . *Journal of Visual Communication and Image Representattion*, 10:39–62, 1999.

CHAPITRE 2

Pseudo-métrique pondérée pour une méthode rapide de recherche d'images par le contenu

Dans ce chapitre, nous proposons un nouveau modèle bayésien de régression logistique basé sur une méthode variationnelle. Une comparaison de ce nouveau modèle avec le modèle classique de régression logistique est effectuée dans le cadre de la recherche d'images. Nous avons illustré que le modèle bayésien permet une meilleure amélioration de la capacité à discriminer de la pseudo-métrique et de la précision de recherche que le modèle classique. Une évaluation comparative a été effectuée sur les bases de données d'images couleurs connues WANG et ZuBud.

Nous présentons dans les pages qui suivent, un article intitulé **Weighted Pseudo-Metric for a Fast CBIR Method** qui est accepté pour publication dans le journal **Machine Graphics and Vision (MGV)**. Une version préliminaire de l'article a été présentée à l'**International Conference on Computer Vision and Graphics (ICCVG'06)**, Varsovie, Pologne, 2006.

Weighted Pseudo-Metric for a Fast CBIR Method

R. Ksantini[1], D. Ziou[1], B. Colin[2], and F. Dubeau[2]

(1) Département d'informatique, Faculté des sciences
Université de Sherbrooke
Sherbrooke, Qc, Canada J1K 2R1.
Email: riadh.ksantini@usherbrooke.ca
djemel.ziou@usherbrooke.ca
(2) Département de mathématiques, Faculté des sciences
Université de Sherbrooke
Sherbrooke, Qc, Canada J1K 2R1.
Email: bernard.colin@usherbrooke.ca
francois.dubeau@usherbrooke.ca

Keywords: Color Image Retrieval, Weighted Pseudo-Metric, Logistic Regression Models.

Abstract

In this paper, a simple and fast querying method for content-based image retrieval is presented. In order to measure the similarity degree between two color images both quickly and effectively, we use a weighted pseudo-metric which makes use of the one-dimensional Daubechies decomposition and compression of the extracted feature vectors. In order to improve the discriminatory capacity of the pseudo-metric, we compute its weights using a classical logistic regression model and a Bayesian logistic regression model, separately. The Bayesian logistic regression model was shown to be a significantly better tool than the classical logistic regression model to improve the retrieval performance. Experimental results are reported on the WANG and ZuBuD color image databases proposed by [11].

1 Introduction

The rapid expansion of the Internet and the wide use of digital data in many real world applications in the field of medecine, security, communications, commerce and academia, increased the need for both efficient image database creation and retrieval procedures. For this reason, content-based image retrieval (CBIR) approach was proposed [6]. In this approach, each image from the database is associated with a feature vector capturing certain visual features of the image such as color, texture and shape. Then, a similarity measure is used to compare these feature vectors and to find similarities between images with the assumption that images that are close to each other in the feature space are also visually similar. Distance measures like the Euclidean distance have been the most widely used to measure similarities between feature vectors in the content-based image retrieval (CBIR) systems. However, similarity measures make no assumption about the probability distributions and the local relevances of the feature vectors, thereby irrelevant features might hurt retrieval performance. Probabilistic approaches are a promising solution to this CBIR problem [13], that when compared to the standard CBIR methods based on the distance measures, can lead to a significant gain in retrieval accuracy. In fact, these approaches are capable of generating probabilistic similarity measures and highly customized metrics for computing image similarity. As to previous works based on these probabilistic approaches, J. Peng *et al.* [4] used a binary classification to classify the database color image feature vectors as relevant or irrelevant, G. Caenen and E. J. Pauwels [10] used the classical quadratic logistic regression model, in order to classify database image feature vectors as relevant or irrelevant and S. Aksoy and R. M. Haralick [8] measure the similarity degree between a query image and a database image using a likelihood ratio derived from a Bayesian classifier. In this paper, we propose a simple and fast querying method for content-based image retrieval. In order to measure the similarity degree between two color images, we use a weighted pseudo-metric used in [14], which makes use of the compressed and quantized versions of the Daubechies-8 wavelet decomposed histograms. In order to discriminate most effectively, the pseudo-metric weights are adjusted using separately a classical logistic regression model and a Bayesian logistic regression model based on a variational method. The Bayesian logistic regression model was shown to be a significantly better tool than the classical logistic regression model to improve the retrieval performance. Evaluation and comparison of both models were conducted on the WANG and ZuBuD color image databases proposed by [11].

This paper is organized as follows. In the next section, we briefly redefine the pseudo-metric. In section 3, we will present the pseudo-metric weight adjustment using the classical and Bayesian logistic regression models. Then, we will describe the data training performed for both models. The color image retrieval method and the feature vectors that we use to represent the database color images are presented in section 4. Finally, in section 5, we will perform some experiments to validate the Bayesian logistic regression model and we will use the precision and scope [16], in order to show the advantage of the Bayesian logistic regression model over the classical logistic regression one, in terms of querying results.

2 The pseudo-metric

Given a query feature vector Q and a featurebase of $|DB|$ feature vectors T_k ($k = 1, ..., |DB|$) having 2^J components each, our aim is to retrieve in the featurebase the most similar feature vectors to Q. To achieve this, Q and the $|DB|$ feature vectors are Daubechies-8 wavelet decomposed, compressed to m coefficients each and quantized. Then, to measure the similarity degree between Q and a target feature vector T_k of the featurebase, we use the pseudo-metric used in [14] and given by the following expression

$$\| Q, T_k \| = \tilde{w}_0 |\tilde{Q}[0] - \tilde{T}_k[0]| - \sum_{i:\tilde{Q}^c_q[i] \neq 0} w_{bin(i)} (\tilde{Q}^c_q[i] = \tilde{T}^c_{kq}[i]), \tag{1}$$

where

$$\left(\tilde{Q}^c_q[i] = \tilde{T}^c_{kq}[i]\right) = \begin{cases} 1 & \text{if } \tilde{Q}^c_q[i] = \tilde{T}^c_{kq}[i] \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

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

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

Since the pseudo-metric makes use of the one-dimensional Daubechies-8 decomposition and compression of the extracted feature vectors, the retrieval will be done quickly and effectively.

3 Pseudo-metric weight adjustment

In order to improve the discriminatory power of the pseudo-metric, we compute its weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ using a classical logistic regression model and a Bayesian logistic regression model, separately. We define two classes, the relevance class denoted by Ω_0 and the irrelevance class denoted by Ω_1, in order to classify the feature vector pairs as similar or dissimilar. The basic principle of using the Bayesian logistic regression model and the classical logistic regression one is to perform a good linear separation between Ω_0 and Ω_1, and then to compute the weights which represent the local relevances of the pseudo-metric components.

3.1 The classical logistic regression model

In this model, each feature vector pair is represented by an explanatory vector and a binary target variable. Specifically, for the i-th pair of feature vectors which are Daubechies-8 wavelet decomposed, compressed and quantized, we associate an explanatory vector $X_i = (\tilde{X}_{0,i}, X_{0,i}, ..., X_{J-1,i}, 1) \in \mathbb{R}^J \times \{1\}$ and a binary target S_i which is either 0 or 1, depending on whether or not the two feature vectors are intended to be similar. $\tilde{X}_{0,i}$ is the absolute value of the difference between the scaling factors of feature vectors and $\{X_{k,i}\}_{k=0}^{J-1}$ are the numbers of mismatches between their J resolution level coefficients. We suppose that we have n_0 pairs of similar feature vectors and n_1 pairs of dissimilar ones. Thus, the class Ω_0 contains n_0 explanatory vectors and their associated binary target variables $\{X_i^r, S_i^r = 0\}_{i=1}^{n_0}$ to represent the pairs of the similar feature vectors, and the class Ω_1 contains n_1 explanatory vectors and their associated binary target variables $\{X_j^{ir}, S_j^{ir} = 1\}_{j=1}^{n_1}$ to represent the pairs of the dissimilar feature vectors. According to [14], the pseudo-metric weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ and an intercept v are chosen to optimize a conditional log-likelihood. For this reason, standard optimization algorithms such as the Fisher scoring and gradient ascent algorithms [3], can be invoked. However, according to [12] and [5], in several cases, especially because of the exponential in the likelihood function or because of the existence of many zero explanatory vectors, the maximum likelihood can fail and estimates of the parameters of interest (weights and intercept) may not be optimal or may not exist or may be on the boundary of the parameter space. This problem can be solved by smoothing the parameter of interest estimates, assuming a certain prior distribution for the parameters. This motivates the adoption of a Bayesian logistic regression model with gaussian prior over the parameters.

3.2 The Bayesian logistic regression model

In the Bayesian logistic regression framework, there are three main components which are 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 mean components are the parameter of interest estimates. However, when the posterior distribution has no tractable form, its mean computation involves high-dimensional integration which has high computational cost. According to [7], it's possible to use accurate variational transformations in order to approximate the likelihood function with a simpler tractable exponential form. In this case, thanks to the conjugacy, with a gaussian prior distribution over the parameters of interest combined with the likelihood approximation, the posterior distribution is approximated with a closed gaussian form. However, in this model, to each explanatory vector a variational parameter is associated. Therefore, if the number of observations is large, the number of variational parameters updated to optimize the posterior distribution approximation is also large, thereby the computational cost is high. In the model that we propose, we use variational transformations and the Jensen's inequality in order to approximate the likelihood function with tractable exponential form. The explanatory vectors are not observed but instead are distributed according to two specific distributions. The posterior distribution is also approximated with a gaussian which depends only on two variational parameters. The computation of the posterior distribution approximation mean is fast and has low computational complexity. In this model, we 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 $\underline{\mathrm{X}}_0 = (\tilde{\underline{\mathrm{X}}}_{0,0}, \underline{\mathrm{X}}_{0,0}, ..., \underline{\mathrm{X}}_{J-1,0}, 1)$ and $\underline{\mathrm{X}}_1 = (\tilde{\underline{\mathrm{X}}}_{0,1}, \underline{\mathrm{X}}_{0,1}, ..., \underline{\mathrm{X}}_{J-1,1}, 1)$, 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. For $\underline{\mathrm{X}}_0$ 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 for $\underline{\mathrm{X}}_1$ 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. Parameters of interest (weights and intercept) are considered as random variables and are denoted by the random vector $\underline{\mathrm{W}} = (\underline{\tilde{w}}_0, \underline{w}_0, ..., \underline{w}_{J-1}, \underline{v})$. We assume that $\underline{\mathrm{W}} \sim \pi(\underline{\mathrm{W}})$, where π is a gaussian prior with prior mean μ and covariance matrix Σ. Using Bayes' rule, the posterior distribution over $\underline{\mathrm{W}}$ is given by

$$P(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1) = \frac{\left[\sum_{x_0 \in \Omega_0, x_1 \in \Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i|\underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\right]\pi(\underline{\mathrm{W}})}{P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1)}, \tag{4}$$

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 $i \in \{0,1\}$ and $F(x) = \frac{e^x}{1+e^x}$. Using a variational approximation [7] and the Jensen's inequality, the posterior distribution is approximated as follows

$$\begin{aligned} P(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1) &\geq \frac{\underline{\mathrm{P}}\big(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}\big)\pi(\underline{\mathrm{W}})}{P(\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1)}, \qquad (5) \\ &\propto \underline{\mathrm{P}}\big(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}\big)\pi(\underline{\mathrm{W}}), \qquad (6) \end{aligned}$$

where

$$\begin{aligned}
&\underline{\mathrm{P}}(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}) \\
&= \left[\prod_{i=0}^{1} F(\epsilon_i)\right] e^{\left[\sum_{i=0}^{1}\left[\frac{E_{q_i}[H_i]-\epsilon_i}{2}\right]-\sum_{i=0}^{1}\left[\varphi(\epsilon_i)\left(E_{q_i}[H_i^2]-\epsilon_i^2\right)\right]\right]},
\end{aligned}$$

where E_{q_0} and E_{q_1} are the expectations with respect to the distributions q_0 and q_1, respectively, $\varphi(\epsilon_i) = \frac{tanh(\frac{\epsilon_i}{2})}{4\epsilon_i}$ and $\{\epsilon_i\}_{i=0}^{1}$ are the variational parameters. Therefore, the approximation of the posterior distribution is considered as an adjustable lower bound and as a proper Gaussian distribution with a posterior mean μ_{post} and covariance matrix Σ_{post} which are estimated by the following Bayesian update equations

$$\begin{aligned}
(\Sigma_{post})^{-1} &= (\Sigma)^{-1} + 2\sum_{i=0}^{1}\left[\varphi(\epsilon_i)E_{q_i}[x_i(x_i)^t]\right], \\
\mu_{post} &= \Sigma_{post}\left[(\Sigma)^{-1}\mu + \sum_{i=0}^{1}\left[(i-\frac{1}{2})E_{q_i}[x_i]\right]\right].
\end{aligned}$$

The weight and intercept computation algorithm is in two phases. The first phase is the initialization of q_0, q_1 and the gaussian prior $\pi(\underline{\mathrm{W}})$, and the second phase is iterative and allows the computation of Σ_{post} and μ_{post} through the above Bayesian update equations, while using an EM type algorithm [1], [15], in order to find the variational parameters $\{\epsilon_i\}_{i=0}^{1}$ at each iteration to have an optimal approximation to the posterior distribution. In the initialization phase, q_0 and q_1 are chosen to model Ω_0 and Ω_1, respectively, and because of the absence of prior knowledge about the weights and the intercept, $\pi(\underline{\mathrm{W}})$ is chosen univariate with zero mean and large variances [9]. The values of μ_{post} components are the desired estimates of the pseudo-metric weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ and the intercept v.

3.3 Training

Let us consider a color image database which consists of several color image sets such that each set contains color images which are perceptually close to each other in terms of object shapes and colors. In order to compute the pseudo-metric weights and the intercept by the classical logistic regression model, we have to create the relevance class Ω_0 and the irrelevance class Ω_1. To create Ω_0, we draw all possible pairs of feature vectors representing color images belonging to the same set in the database, and for each pair we compute an explanatory vector and associate a binary target variable equal to 0. Similarly, to create Ω_1, we draw all possible pairs of feature vectors representing color images belonging to different sets in the database, and for each pair we compute an explanatory vector and associate a binary target variable equal to 1. For the Bayesian logistic regression model, we create the Ω_0 and Ω_1 with the same way, but instead of associating a binary target variable value for each explanatory vector of Ω_0 and Ω_1, we associate a binary target variable $\underline{\mathrm{S}}_0$ equal to 0 for all Ω_0 explanatory vectors and a binary target variable $\underline{\mathrm{S}}_1$ equal to 1 for all Ω_1 explanatory vectors.

4 Color image retrieval method

4.1 Color image database preprocessing and the querying algorithm

The retrieval method is in two phases. The first phase is a preprocessing phase done once for the entire database containing $|DB|$ color images. The second phase is the querying phase.

Preprocessing:

1. Choose N feature vectors for comparison.
2. Compute the N feature vectors T_{li} $(l \in \{1, ..., N\})$ for each i-th color image of the database, where $i \in \{1, ..., |DB|\}$.
3. The feature vectors representing the database color images are Daubechies-8 wavelets decomposed, compressed to m coefficients each and quantized.
4. Organize the decomposed, compressed and quantized feature vectors into search arrays Θ^l_+ and Θ^l_- $\big(l = 1, ..., N\big)$ which are used to optimize the pseud-metric computation process [14].
5. Adjustment of the metric weights $\tilde{w}^l_0$ and $\{w^l_k\}^{J-1}_{k=0}$ for each featurebase T_{li} $(i = 1, ..., |DB|)$ representing the database color images, where $l \in \{1, ..., N\}$.

Querying algorithm:

1. Given a query color image, we denote the feature vectors representing the query image by Q_l $\big(l = 1, ..., N\big)$.
2. The feature vectors representing the query image are Daubechies-8 wavelets decomposed, compressed to m coefficients each and quantized.
3. The similarity degrees between Q_l $\big(l = 1, ..., N\big)$ and the database color image feature vectors T_{li} $\big(l = 1, ..., N\big)$ $(i = 1, ..., |DB|)$ are represented by the arrays $Score_l$ $\big(l = 1, ..., N\big)$ such that $Score_l[i] =\|$

$Q_l, T_{li} \|$ for each $i \in \{1, ..., |DB|\}$. These arrays are returned by the procedure Retrieval(Q_l, m, Θ^l_+, Θ^l_-) ($l = 1, ..., N$), respectively. The procedure Retrieval is used to optimize the querying process [14].

4. The similarity degrees between the query color image and the database color images are represented by a resulted array $TotalScore$, such as, $TotalScore[i] = \sum_{l=1}^{N} \gamma_l Score_l[i]$ for each $i \in \{1, ..., |DB|\}$, where $\{\gamma_l\}_{l=1}^{N}$ are weightfactors used to fine-tune the influence of each individual feature.

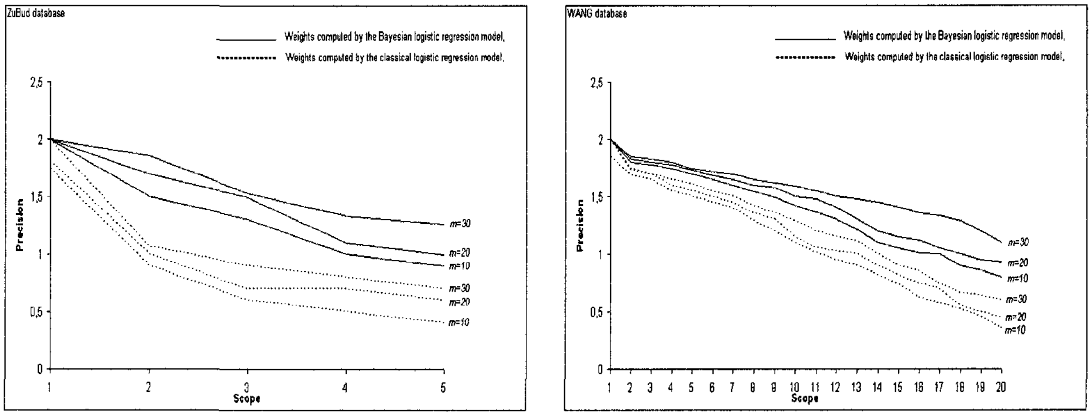
5. Organize the database color images in order of increasing resulted similarity degrees of the array $TotalScore$. The most negative resulted similarity degrees correspond to the closest target images to the query image. Finally, return to the user the closest target color images to the query color image and whose number is denoted by RI and chosen by the user.

4.2 Used feature vectors

Before feature vector extraction, a color image is represented in the perceptually uniform LAB color space. In order to describe the luminance, colors and edges of the color image, we use luminance histogram and weighted histograms proposed by [14]. The weighted histograms are the color histograms constructed after edge region elimination and the multispectral gradient module mean histograms. We denote the luminance histogram by h_L, the multispectral gradient module mean histograms by $\bar{h}^e_a$ and $\bar{h}^e_b$, and the color histograms constructed after edge region elimination by h^h_a and h^h_b. The image texture description is performed by kurtosis and skewness histograms [2]. Kurtosis histograms are denoted by h^κ_L, h^κ_a and h^κ_b, and skewness histograms are denoted by h^s_L, h^s_a and h^s_b. They are obtained by local computations of the kurtosis and skewness values at the luminance and chrominance image pixels. Then, a linear interpolation is used to represent the kurtosis and skewness values between 0 and 255. Since each used feature vector is a histogram having 256 components, we set J equal to 8 in the following section.

5 Experimental results

The choices of the distributions q_0 and q_1 and the querying evaluation will be conducted on the WANG and ZuBuD color image databases described in [11]. Since from each color image of the ZuBuD and WANG databases we extract $N = 11$ histograms which are h_L, h^h_a, h^h_b, $\bar{h}^e_a$, $\bar{h}^e_b$, h^κ_L, h^κ_a, h^κ_b, h^s_L, h^s_a and h^s_b, each database is represented by eleven featurebases. The choices of q_0 and q_1 will be separately performed for each of those. For each featurebase, we assume that $\tilde{\underline{X}}_{0,0}$ and $(\underline{X}_{0,0}, ..., \underline{X}_{J-1,0})$ are independent. We make the same assumption for $\underline{\tilde{X}}_{0,1}$ and $(\underline{X}_{0,1}, ..., \underline{X}_{J-1,1})$. Moreover, we suppose that the random vector $(\underline{X}_{0,0}, ..., \underline{X}_{J-1,0})$ random variables are independent and each one of them follows a poisson distribution. Analogously, we make the same choice for $(\underline{X}_{0,1}, ..., \underline{X}_{J-1,1})$. Also, we assume that the random variable $\underline{\tilde{X}}_{0,0}$ follows a gaussian mixture distribution, which is the same choice for $\underline{\tilde{X}}_{0,1}$. Generally, to carry out an evaluation in the image retrieval field, two principal issues are required: the acquisition of ground truth and the definition of performance criteria. For ground truth, three external persons participate in the evaluation. Concerning performance criteria, we represent the evaluation results by the precision-scope curve $Pr = f(RI)$ [16]. In each querying performed in the evaluation experiment, each human subject is asked to give a goodness score to each retrieved image. The goodness score is 2 if the retrieved image is almost similar to 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 precision is computed as follows: $Pr =$ the sum of goodness scores for retrieved images$/RI$. Therefore, the curve $Pr = f(RI)$ gives the precision for different values of RI which lie between 1 and 20 when we perform the evaluation on the WANG database, and between 1 and 5 when we perform the evaluation on the ZuBuD database. When the human subjects perform different queries in the evaluation experiment, we compute an average precision for each value of RI, and then we construct the precision-scope curve. 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 classical logistic regression model, and assign goodness score to each retrieved image; and then to reformulate a query from the database, execute the querying procedure using weights computed by the Bayesian logistic regression model, and assign goodness score to each retrieved image. Each human subject repeats the querying process twenty times, choosing a new query from the database each time. We repeat this experience for different orders of compression $m \in \{30, 20, 10\}$, keeping the weightfactors $\{\gamma_l\}_{l=1}^{3}$ 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. To evaluate the querying in the ZuBuD database, each human subject is asked to follow the preceding steps, while formulating the queries from the database query part. For the ZuBuD and WANG databases, the resulting precision-scope curves for each compression order $m \in \{30, 20, 10\}$ are given in Figure 1.



(a)

(b)

Figure 1: Evaluation ((a) ZuBud database and (b)WANG database): precision-scope curves for retrieval using weights computed by the classical logistic regression model and weights computed by the Bayesian logistic regression model.

6 Conclusion

We presented a simple, fast and effective color image querying method. In order to measure the similarity degree between two color images both quickly and effectively, we used a weighted pseudo-metric which makes use of the one-dimensional Daubechies decomposition and compression of the extracted feature vectors. A Bayesian logistic regression model and a classical logistic regression one were used to improve the discriminatory capacity of the pseudo-metric. Evaluations of the querying method showed that the Bayesian logistic regression model is a better tool than the classical logistic regression one to compute the pseudo-metric weights and to improve the querying results. Thanks to the effectiveness and flexibility of the Bayesian logistic regression model, the use of the pseudo-metric for comparison and its weight computation, can be customized to other featurebases representing other image databases. Precisely, a user can compute the pseudo-metric weights after choosing q_0 and q_1 according to his data, and then can perform effective and fast querying by using the pseudo-metric for comparison.

References

[1] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the EM Algorithm," *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1-38, 1977.

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

[3] 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. American Statistical Association*, vol. 86, pp. 68-78, 1991.

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

[5] 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.

[6] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, no. 1, pp. 25-37, 2000.

[7] 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.

[8] 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.

[9] P. Congdon, *Bayesian Statistical Modelling*, Chichester, John Wiley, 2001.

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

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

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

[14] 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.

[15] R. Ksantini, D. Ziou, B. Colin, and F. Dubeau, "Weighted Pseudo-Metric Discriminatory Power Improvement Using a Bayesian Logistic Regression Model Based on a Variational Method". University of Sherbrooke, Research Report N^o 16, 2006.

[16] 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.

CHAPITRE 3

Amélioration de la capacité à discriminer d'une pseudo-métrique en utilisant un modèle bayésien de régression logistique fondé sur une méthode variationnelle

Dans ce chapitre, nous détaillons la dérivation du nouveau modèle bayésien de régression logistique basé sur une méthode variationnelle introduite au chapitre 2. Nous effectuons une comparaison exhaustive entre ce modèle et le modèle classique de régression logistique dans le cadre de la recherche d'images et dans un cadre général. Plus spécifiquement, dans ce cadre général, nous comparons le modèle bayésien à d'autres classificateurs linéaires apparaissant dans la littérature. Ensuite, nous comparons notre méthode de recherche utilisant le modèle bayésien de régression logistique à d'autres méthodes de recherches déjà publiées. Les expérimentations et comparaisons ont été effectuées sur les bases de données d'images couleurs connues WANG, ZuBud, UW et CalTech et sur plusieurs ensembles de données réelles et synthétiques.

Nous présentons dans les pages qui suivent, un article intitulé **Weighted Pseudo-Metric Discriminatory Power Improvement Using a Bayesian Logistic Regression Model Based on a Variational Method** qui est accepté pour publication dans l'**IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)**.

Weighted Pseudo-Metric Discriminatory Power Improvement Using a Bayesian Logistic Regression Model Based on a Variational Method

R. Ksantini[1], D. Ziou[1], B. Colin[2] and F. Dubeau[2]

(1) Département d'informatique, Faculté des sciences
Université de Sherbrooke
Sherbrooke, QC, Canada J1K 2R1.
Email: riadh.ksantini@usherbrooke.ca
djemel.ziou@usherbrooke.ca
(2) Département de mathématiques, Faculté des sciences
Université de Sherbrooke
Sherbrooke, QC, Canada J1K 2R1.
Email: bernard.colin@usherbrooke.ca
francois.dubeau@usherbrooke.ca

Keywords: Image Retrieval, Logistic Regression, Variational Method, Weighted Pseudo-Metric.

Abstract

In this paper, we investigate the effectiveness of a Bayesian logistic regression model to compute the weights of a pseudo-metric, 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 pseudo-metric makes use of the compressed and quantized versions of wavelet decomposed feature vectors, and in our previous work, the weights were adjusted by 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 pseudo-metric weights and improve retrieval and classification performance. Finally, we perform a comparison with results obtained by other retrieval methods.

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, 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 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: G. Caenen and E. J. 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. S. Aksoy and R. M. 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 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. N. 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. T. Westerveld and A. P. 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. V. 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 modelled, respectively, by multinomial distribution and Gaussian kernels, images are ranked according to the conditional probability. S. 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 which can be used for CBIR purposes. We will now review some relevant examples of this work. J. 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 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. S. 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. T. Hastie and R. 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. C. Domeniconi *et al.* [30] pursue the idea presented in [29], but use 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. S. 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 which uses this Mahalanobis distance idea is Fisher's Linear Discriminant Analysis (LDA) (see e.g. [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 E. P. 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. J. Goldberger *et al.* [27] propose neighborhood component analysis (NCA), a novel algorithm for learning a Mahalanobis distance, designed to improve the KNN classification algorithm. The algorithm maximizes a non-convex 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 K. Q. Weinberger [28] and A. Globerson and S. Roweis [31], who pursue essentially the same goals as NCA, but differ in their construction of convex objective functions.

Our retrieval approach consists of learning a weighted pseudo-metric using a Bayesian logistic regression model based on a variational method, and it has several advantages. First, the pseudo-metric 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 pseudo-metric 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 which allows the training to have low computational complexity, while preserving a good classification performance. In our previous work [8], the pseudo-metric 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 pseudo-metric 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 pseudo-metric and explain the fast feature vector querying algorithm. In Section 3, we explain the data training process and describe the adjustment of the pseudo-metric 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 pseudo-metric 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 models is performed for CBIR, and a comparison with results for different retrieval methods is provided.

2 The pseudo-metric and the fast feature vector querying algorithm

2.1 The pseudo-metric

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 pseudo-metric is given by the following expression:

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

where

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

$\tilde{Q}[0]$ and $\tilde{T}[0]$ are the scaling function coefficients; $\tilde{Q}_q^c[i]$ and $\tilde{T}_q^c[i]$ represent the i-th 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

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

To compute the pseudo-metric 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

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

where

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

Since the term $\sum_{i:\tilde{Q}_q^c[i]\neq 0} w_{bin(i)}$ is independent of the vectors $\tilde{T}_q^c$ and $\tilde{Q}_q^c$, we can discard it. Therefore, our pseudo-metric 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 which 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 i-th 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 i-th 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 $(k = 1, ..., |DB|)$, 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 $k \in \{1, ..., |DB|\}$
    For each $k \in \{1, ..., |DB|\}$ 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($\tilde{Q}$,$m$)
    $\tilde{Q}_q^c \leftarrow$ Quantify($\tilde{Q}^c$)
    For each $\tilde{Q}_q^c[i] \neq 0$ do

        If $\tilde{Q}_q^c[i] > 0$ 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 $k \in \{1, ..., |DB|\}$. The elements of $Score$, which are the degrees of similarity between the query Q and the feature vectors T_k ($k \in \{1, ..., |DB|\}$), can be negative or positive. The most negative similarity degree corresponds to the closest target to the query Q.

3 Pseudo-metric weight adjustment

The weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ are adjusted in such a way that the pseudo-metric 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 $\{X_i^r\}_{i=1}^{n_0}$ to represent the pairs of similar feature vectors, and Ω_1 contains n_1 explanatory vectors $\{X_j^{ir}\}_{j=1}^{n_1}$ to represent the pairs of dissimilar feature vectors. Given an explanatory vector $X_i^r = (\tilde{X}_{0,i}^r, X_{0,i}^r, ..., X_{J-1,i}^r, 1) \in \Omega_0$ representing a pair of similar feature vectors which are wavelet decomposed, compressed and quantized, $\tilde{X}_{0,i}^r$ is the absolute value of the difference between their scaling factors and $\{X_{k,i}^r\}_{k=0}^{J-1}$ are the numbers of mismatches between their J resolution level coefficients. The components of an explanatory vector $X_j^{ir} = (\tilde{X}_{0,j}^{ir}, X_{0,j}^{ir}, ..., X_{J-1,j}^{ir}, 1) \in \Omega_1$ are computed for a pair of dissimilar feature vectors in the same way. The basic aim of using the Bayesian and classical logistic regression models is to allow a good separation between Ω_0 and Ω_1 by hyperplane, and to compute the weights which represent the local relevances of the pseudo-metric 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 which are perceptually close to each other in terms of visual features such as color, texture and shape. The purpose of weighting the pseudo-metric is to make it efficient enough to match images which belong to the same cluster and discriminate between images which 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_i^r of Ω_0 is associated with a binary target variable $S_i^r = 0$ for similarity, and each explanatory vector X_j^{ir} of Ω_1 is associated with a binary target variable $S_j^{ir} = 1$ for dissimilarity. Given two explanatory vectors X_i^r and X_j^{ir}, we model their associated binary target variables S_i^r and S_j^{ir}, respectively, by a relevance probability p_i^r and an irrelevance probability p_j^{ir}, defined as follows:

$$p_j^{ir} = P\Big(S_j^{ir} = 1|X_j^{ir}\Big) = F(\tilde{w}_0\tilde{X}_{0,j}^{ir} + \sum_{k=0}^{J-1} w_k X_{k,j}^{ir} + v) \tag{7}$$

$$p_i^r = P\Big(S_i^r = 0|X_i^r\Big) = F(-\tilde{w}_0\tilde{X}_{0,i}^{r} - \sum_{k=0}^{J-1} w_k X_{k,i}^{r} - v), \tag{8}$$

where $F(x) = \frac{e^x}{1+e^x}$ is the logistic function and v is an unknown intercept which will be computed with the pseudo-metric weights, but will not be considered when using the pseudo-metric 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; i.e., 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 equations (7) and (8) to compute the probabilities p_j^{ir} and p_i^r, then the weights and the intercept are chosen to maximize the following conditional log-likelihood:

$$log\big(L(W = (\tilde{w}_0, w_0, ..., w_{J-1}, v))\big) = \sum_{i=1}^{n_0} log(p_i^r) + \sum_{j=1}^{n_1} log(p_j^{ir}). \tag{9}$$

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 $w_j \geq 0 \; \forall j \in \{0, ..., J-1\}$ [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 R. 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 which 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 which 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 which 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 modelling [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 MAP approach and high-dimensional integration algorithms [12], which have 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 which depends only on two variational parameters. 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 $\underline{X}_0 = (\tilde{\underline{X}}_{0,0}, \underline{X}_{0,0}, ..., \underline{X}_{J-1,0}, 1)$ and $\underline{X}_1 = (\tilde{\underline{X}}_{0,1}, \underline{X}_{0,1}, ..., \underline{X}_{J-1,1}, 1)$, respectively. We suppose that $\underline{X}_0 \sim q_0(\underline{X}_0)$ and $\underline{X}_1 \sim q_1(\underline{X}_1)$, where q_0 and q_1 are two chosen distributions. With $\underline{X}_0$ we associate a binary random variable $\underline{S}_0$ whose realizations are the target variables $\{S_i^r = 0\}_{i=1}^{n_0}$, and with $\underline{X}_1$ we associate a binary random variable $\underline{S}_1$ whose realizations are the target variables $\{S_j^{ir} = 1\}_{j=1}^{n_1}$. We set $\underline{S}_0$ equal to 0 for similarity and we set $\underline{S}_1$ equal to 1 for dissimilarity. The parameters of interest (weights and intercept) are considered as random variables and are denoted by the random vector $\underline{W} = (\underline{\tilde{w}_0}, \underline{w_0}, ..., \underline{w_{J-1}}, \underline{v})$. We assume that $\underline{W} \sim \pi(\underline{W})$, where π is a Gaussian prior with prior mean μ and prior covariance matrix Σ. Using Bayes' rule, the posterior distribution over $\underline{W}$ is given by

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

where

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

Since in the Bayesian approach we generally suppose that the space of unknown parameters is independent from the space of observations, we assume that $\underline{W}$ and $\underline{X}_i$ 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 $i \in \{0,1\}$. So we obtain

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

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

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

where

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

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 is 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 the equation (11), in order to bound it by an exponential form which 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

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

where $\epsilon_i > 0$ is the variational parameter, $\varphi(\epsilon_i) = \frac{tanh(\frac{\epsilon_i}{2})}{4\epsilon_i}$, and $tanh(\frac{\epsilon_i}{2}) = \frac{e^{\frac{\epsilon_i}{2}} - e^{-\frac{\epsilon_i}{2}}}{e^{\frac{\epsilon_i}{2}} + e^{-\frac{\epsilon_i}{2}}}$. So the posterior distribution nominator in the formula (11) can be approximated as follows:

$$\sum_{x_0 \in \Omega_0, x_1 \in \Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i | \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)\ \pi(\underline{\mathrm{W}}) \qquad (14)$$

$$\geq \sum_{x_0 \in \Omega_0, x_1 \in \Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i | \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}, \epsilon_i) q_i(\underline{\mathrm{X}}_i = x_i)\ \pi(\underline{\mathrm{W}}).$$

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

$$\sum_{x_0 \in \Omega_0, x_1 \in \Omega_1} [\prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i | \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}, \epsilon_i) q_i(\underline{\mathrm{X}}_i = x_i)]$$

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

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 the formula (11) can be approximated as follows:

$$[\sum_{x_0 \in \Omega_0, x_1 \in \Omega_1} \prod_{i=0}^{1} P(\underline{\mathrm{S}}_i = i | \underline{\mathrm{X}}_i = x_i, \underline{\mathrm{W}}) q_i(\underline{\mathrm{X}}_i = x_i)] \pi(\underline{\mathrm{W}})$$

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

Thus, the variational posterior approximation is given by

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

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

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

Finally, the posterior distribution is approximated as follows:

$$\begin{aligned} P(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1) &\geq P(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1), && (15) \\ &\propto \underline{P}(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1)\pi(\underline{W}). && (16) \end{aligned}$$

Since $\pi(\underline{W})$ is a Gaussian which is a conjugate of $\underline{P}(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1)$, 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 $P(\underline{W}|\underline{S}_0 = 0, \underline{S}_1 = 1, \{\epsilon_i\}_{i=0}^1, \{q_i\}_{i=0}^1)$ by their Gaussian forms in equation (16), we obtain

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

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

$$\begin{aligned} (\Sigma_{post})^{-1} &= (\Sigma)^{-1} + 2\sum_{i=0}^{1}\left[\varphi(\epsilon_i)E_{q_i}[x_i(x_i)^t]\right], && (17) \\ \mu_{post} &= \Sigma_{post}\left[(\Sigma)^{-1}\mu + \sum_{i=0}^{1}\left[(i-\frac{1}{2})E_{q_i}[x_i]\right]\right]. && (18) \end{aligned}$$

According to equation (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 equation (15), and then an optimal approximation to the posterior distribution. This can be done by an EM algorithm which is derived in Appendix A. The variational parameters are given by

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

where $P(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, (\{\epsilon_i\}_{i=0}^{1})^{old}, \{q_i\}_{i=0}^{1})$ 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 equation (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 $\mu_{post,i} \geq 0 \ \forall \ i \in \{0, ..., J\}$. The values of the μ_{post} components are the desired estimates of the pseudo-metric 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 which model the relevance class Ω_0 and irrelevance class Ω_1 explanatory vectors, respectively.

2. Initialize the covariance matrix Σ^{old} to the identity matrix and the mean μ^{old} to a vector with components equal to 1.

3. Initialize the variational parameters as follows

   For each $i \in \{0, 1\}$ **do**

$$(\epsilon_i^{old})^2 \leftarrow E_{q_i}[(x_i)^t \Sigma^{old} x_i] + (\mu^{old})^t \left[E_{q_i}[x_i(x_i)^t] \right] \mu^{old}$$

   End for

Computation of Σ_{post} and μ_{post}:

1. **Do**

$$\begin{aligned} (\Sigma_{post}^{try})^{-1} &\leftarrow (\Sigma^{old})^{-1} + 2 \sum_{i=0}^{1} \left[\varphi(\epsilon_i^{old}) E_{q_i}[x_i(x_i)^t] \right] \\ \mu_{post}^{try} &\leftarrow \Sigma_{post}^{try} \left[(\Sigma^{old})^{-1} \mu^{old} + \sum_{i=0}^{1} \left[(i - \frac{1}{2}) E_{q_i}[x_i] \right] \right] \end{aligned}$$

   For each $i \in \{0, 1\}$ **do**

$$(\epsilon_i^{try})^2 \leftarrow E_{q_i}[(x_i)^t \Sigma_{post}^{try} x_i] + (\mu_{post}^{try})^t \left[E_{q_i}[x_i(x_i)^t] \right] \mu_{post}^{try}$$

```
        End for

        $\theta \leftarrow \min_{j\in\{0,\dots,J\}} \{\frac{-\mu^{old}_{post,j}}{(\mu^{try}_{post,j} - \mu^{old}_{post,j})}, 1/(\mu^{try}_{post,j} - \mu^{old}_{post,j}) < 0\}$

        $\begin{pmatrix} \mu^{new}_{post} \\ \Sigma^{new}_{post} \\ \epsilon^{new}_{0} \\ \epsilon^{new}_{1} \end{pmatrix} \leftarrow \begin{pmatrix} \mu^{old}_{post} \\ \Sigma^{old}_{post} \\ \epsilon^{old}_{0} \\ \epsilon^{old}_{1} \end{pmatrix} + \theta \left[ \begin{pmatrix} \mu^{try}_{post} \\ \Sigma^{try}_{post} \\ \epsilon^{try}_{0} \\ \epsilon^{try}_{1} \end{pmatrix} - \begin{pmatrix} \mu^{old}_{post} \\ \Sigma^{old}_{post} \\ \epsilon^{old}_{0} \\ \epsilon^{old}_{1} \end{pmatrix} \right]$

    While($|\Sigma^{old}_{post} - \Sigma^{new}_{post}|$ > threshold or $|\mu^{old}_{post} - \mu^{new}_{post}|$ > threshold)
    Return $\mu^{new}_{post}$
```

2. Apply an active set algorithm to correct the false zero solutions of $\mu_{post,i}$ $(i \in \{0, \dots, J\})$ [41], when exiting the iterative phase with $\theta = 0$.

3. Assign the μ_{post} component values to the pseudo-metric 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 $\underline{\mathrm{W}}$. 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.

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-th color image of the database, where $i \in \{1, \dots, |DB|\}$.

3. The feature vectors representing the database color images are wavelet decomposed, compressed to m coefficients each and quantized.

4. Organize the decomposed, compressed and quantized feature vectors into search arrays Θ_+^l and Θ_-^l $\left(l = 1, ..., N\right)$.

5. Adjust the metric weights $\tilde{w}_0^l$ and $\{w_k^l\}_{k=0}^{J-1}$ for each featurebase T_{li} $(i = 1, ..., |DB|)$ representing the database color images, where $l \in \{1, ..., N\}$.

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 $\left(l = 1, ..., N\right)$.

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 $\left(l = 1, ..., N\right)$ and the database color image feature vectors T_{li} $\left(l = 1, ..., N\right)$ $(i = 1, ..., |DB|)$ by the arrays $Score_l$ $\left(l = 1, ..., N\right)$, such that $Score_l[i] =\parallel Q_l, T_{li} \parallel$ for each $i \in \{1, ..., |DB|\}$. These arrays are returned by the procedures Retrieval(Q_l, m, Θ_+^l, Θ_-^l) $\left(l = 1, ..., N\right)$, respectively.

4. Represent the degrees of similarity between the query color image and the database color images by a resultant array $TotalScore$, such that $TotalScore[i] = \sum_{l=1}^{N} \gamma_l Score_l[i]$ for each $i \in \{1, ..., |DB|\}$, where $\{\gamma_l\}_{l=1}^{N}$ are weightfactors 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 logistic regression model (BLRM) to the classical logistic regression model (CLRM) and some relevant state-of-the-art linear classification algorithms which 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 subsection, 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 running time. 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 number of features. We implemented our own C++ code for the BLRM, CLRM, RVM (based on the block-wise algorithm of [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 ten-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 towards 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 Figure 1. Table 1 describes the eight real data sets chosen.

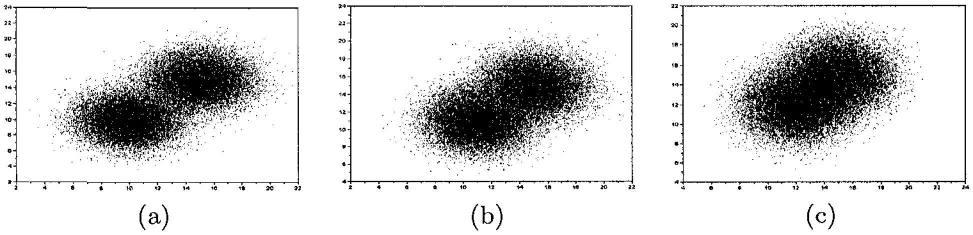


Figure 1: Synthetic data : (a) slightly overlapped clusters (OLR = 0.1), (b) overlapped clusters (OLR = 0.3) and (c) highly overlapped clusters (OLR = 0.5).

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 1: Description of the real data sets.

Image, *Waveform*, *German* and *Breast Cancer* were extracted from the famous UCI collection. More details concerning the original source of these data sets are available in a highly comprehensive online repository [49]. A total of 100 training/test splits are provided by those authors: our results show averages over the first 10 of those. 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 source of these data sets are available in [50]. For each synthetic and real data set, we did 10 independent SVM runs with regularization parameters $C \in \{1, ..., 10\}$, 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 ten-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 running times 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 running times for the various classifiers on the real data sets.

	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

Table 2: Evaluation and comparison on the synthetic data. The four entries (left to right) are training bias (%), variance (%), classifier error (%) and training running time (seconds). Note that classifier error = bias + variance + Bayes error, where the Bayes error is the misclassification rate [46].

	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

Table 3: Evaluation and comparison on the eight real data sets. The three entries (left to right) are test classification accuracy (%), test B index measure ($\in [0, 1]$) and training running time (seconds).

In terms of classification performance and training running time, we can see that the BLRM outperforms the other four classifiers on the synthetic data sets and on all eight real data sets. From tables 2 and 3, it can be seen that the BLRM slightly outperforms the SVM, while achieving significantly lower training running time. 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 running time 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 running time, 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), while 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 which 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 running times 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 subsection, 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, ZuBuD, UW and CalTech color image databases proposed by [17]. The WANG database contains $|DB| = 1000$ color images which were selected manually to form 10 sets (e.g. Africa, beach, ruins, food) of 100 images each. The Zurich Building Image Database (ZuBuD) contains $|DB| = 1005$ color images of buildings selected to form 201 image classes, where each class contains 5 color images of the same building taken from different positions. The UW database contains $|DB| = 1109$ 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 (e.g. barcelona, springflowers, swissmountains): i.e., two images belong to the same class iff their annotations contain identical words. The CalTech database contains $|DB| = 2000$ color images that we selected from the CalTech collection categories (e.g., motorbikes, airplanes, 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 by [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

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

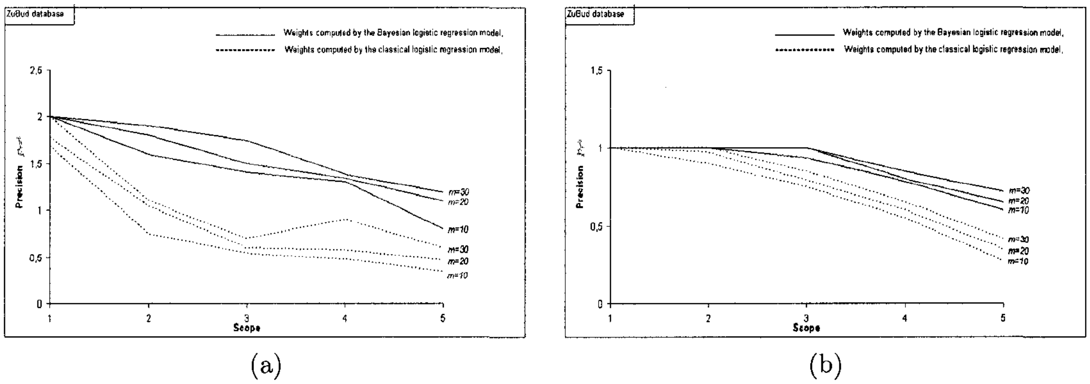
respectively, for each $c \in \{0, ..., 255\}$ and $k = L, a, b$, where I_L^κ, I_a^κ and I_b^κ are the kurtosis images of the luminance L and the chrominances a and b, respectively, and I_L^s, I_a^s and I_b^s are their skewness images. They are obtained by local computation of the kurtosis and skewness 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 1-D 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 subsections.

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 $\tilde{\underline{X}}_{0,0}$ and $(\underline{X}_{0,0}, ..., \underline{X}_{J-1,0})$ are independent. Analogously, for the same reason, we made the same assumption for $\tilde{\underline{X}}_{0,1}$ and $(\underline{X}_{0,1}, ..., \underline{X}_{J-1,1})$. For each histogram featurebase, we suppose that the random vector $(\underline{X}_{0,0}, ..., \underline{X}_{J-1,0})$ 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 $(\underline{X}_{0,1}, ..., \underline{X}_{J-1,1})$. 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 $\tilde{\underline{X}}_{0,0}$ are positive real numbers. We modelled 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 validity function (MML), as it has been shown to give good results in [16]. Similarly, the realizations $\tilde{\underline{X}}_{0,1}$ were modelled 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, while, 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.

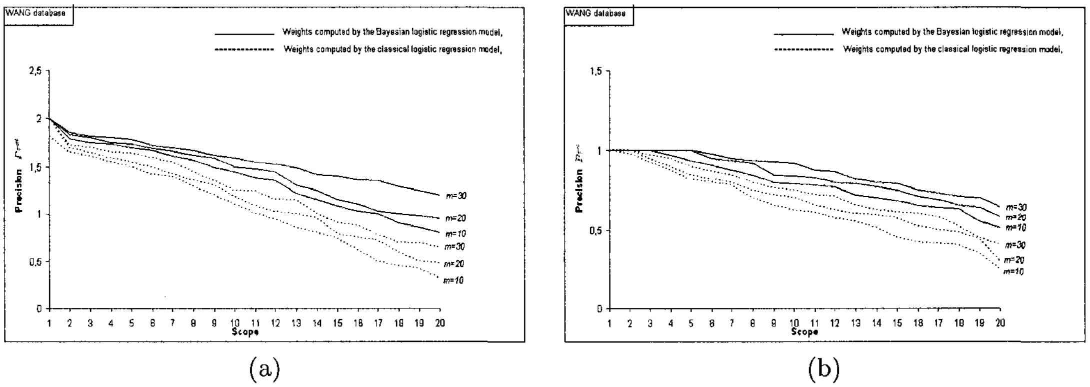
5.2.3 Comparative evaluation of the querying procedure

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 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 labelling of images into classes. In each query performed in the evaluation experiment, each human subject is asked to assign a goodness score and a labelling 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 labelling 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} = the sum of goodness scores for retrieved images/RI and Pr^{c} = the sum of labelling 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 labelling 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 labelling scores to each retrieved image. Each human subject repeats the querying process fifty 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 weightfactors $\{\gamma_l\}_{l=1}^{3}$ 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 the figures below for the ZuBud, WANG, UW and CalTech databases.



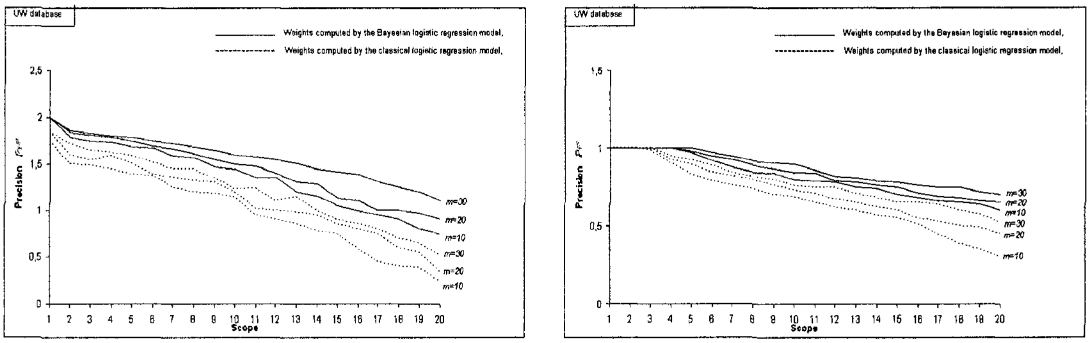
(a) (b)

Figure 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 $m \in \{30, 20, 10\}$.



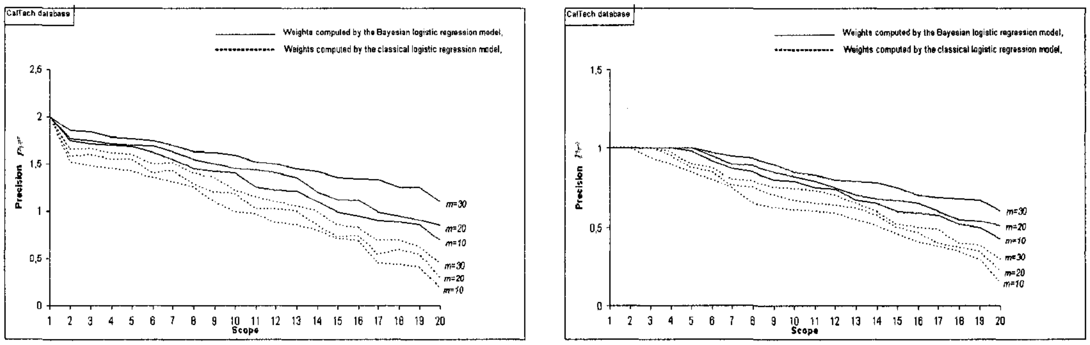
(a) (b)

Figure 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 $m \in \{30, 20, 10\}$.



(a) (b)

Figure 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)

Figure 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 $m \in \{30, 20, 10\}$.

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 the subsection (3.2). In order to compare our image retrieval method, when using the BLRM, to others proposed by [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 $1 - Pr^c(1)$, 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, while in [53] and [52] CalTech and ZuBud were used, respectively. To enable comparison with the results obtained in these works, we set the weightfactors $\{\gamma_l\}_{l=1}^{11}$ equal to 1 to give all features 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 leaving-one-out manner. All images of WANG and UW were selected as queries, while 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 of [17], [53] and [52].

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 %

Table 4: Comparison: ER [%] averages for our retrieval method, when using the BLRM, and other retrieval methods.

6 Conclusion

We have proposed an effective Bayesian logistic regression model 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, computation of the parameters of interest is straightforward and fast. Incorporation of the prior knowledge of the explanatory vectors in the model also optimizes 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 classical logistic regression model. We performed a decontextualized comparison of the Bayesian logistic regression model to the classical logistic regression model and to some relevant state-of-the-art linear classification algorithms. Experiments showed that the Bayesian logistic regression model outperforms these algorithms and the classical logistic regression model in terms of classification performance and training running time. Also, we performed an evaluation and comparison of the Bayesian and classical logistic regression models in the image retrieval context. Experiments showed that the Bayesian logistic regression model 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 which use 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 Communication 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] Nuno 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 on Content-Based Access of Image and Video Libraries*, pp. 14-18, Santa Barbara, California, USA, 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," *Storage and Retrieval for Media Databases, Proc. SPIE*, vol. 4676, pp. 49-58, San Jose, CA, USA, 2002.

[7] S. Aksoy, R. M. Haralick, F. A. Cheikh, and M. Gabbouj, "A Weighted Distance Approach to Relevance Feedback," *15th Int'l Conf. Pattern Recognition (ICPR'00)*, vol. 4, pp. 812-815, Barcelona, Spain, 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. American Statistical Association*, 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*, Chichester, John Wiley, 2001.

[14] G. Koop and D. Poirier, "An Empirical Investigation of Wagner's Hypothesis by Using a Model Occurrence Framework," *J. Royal Statistical Society, 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, 1998.

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

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

[19] H. A. Murthy and S. Haykin, "Bayesian Classification of Surface-Based Ice-Radar Images," *IEEE J. Oceanic Engineering*, 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 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," *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, New York, USA, 2001.

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

[28] K. Weinberger, J. Blitzer, and L. Saul, "Neighbourhood Components Analysis," *Advances in Neural Information Processing Systems 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 18*, pp. 451-458, 2006.

[32] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 539-546, San Diego, CA, 2005.

[33] J. Scott Long, *Regression Models for Categorical and Limited Dependent Variables*, A volume in the Sage Series for Advanced Quantitative Techniques, Thousand Oaks, CA, USA, 1997.

[34] D. Hosmer and S. Lemeshow, *Applied Logistic Regression*, A Wiley-Interscience Publication, John Wiley and Sons Inc, New York, USA, 1989.

[35] J. S. Cramer, *Econometric applications of maximum likelihood methods*, Cambridge University Press, Cambridge, UK, 1986.

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

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

[38] A. Albert, 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*, Wiley, Chichester, UK, 1979.

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

[42] M. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 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 15*, pp. 609-616, 2003.

[45] www.svmlight.joachims.org.

[46] L. Breiman, "Bias, variance and arcing classifiers," *University of California, Dept. of Statistics*, no. 460, Technical Report, 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/.

[50] A. Klautau, "Discriminative Gaussian mixture models: A comparison with kernel classifiers," *Proc. Twentieth Int'l Conf. Machine Learning*, pp. 353-360, Washington, DC, 2003.

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

[52] H. Shao, T. Svoboda, T. Tuytelaars, and L. V. Gool, "HPAT Indexing for Fast Object/Scene Recognition based on Local Appearance," *Proc. CIVR, LNCS 2728*, pp. 71-80, Urbana-Champaign, IL, 2003.

[53] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-invariant Learning," *Proc. CVPR*, pp. 264-271, Blacksburg, VG, 2003.

CHAPITRE 4

Modèles de régression logistique pour une méthode rapide de recherche d'images par le contenu fondée sur la sélection des caractéristiques

Dans ce chapitre, nous introduisons la sélection des caractéristiques pour améliorer la méthode de recherche présentée dans les chapitres précédents. La sélection des caractéristiques est effectuée en utilisant séparément les modèles Bayésien et classique de régression logistique. Elle permet de donner automatiquement plus d'importance aux caractéristiques qui discriminent le plus et moins d'importance aux caractéristiques qui discriminent le moins. Une comparaison des deux modèles est effectuée dans le cadre de la recherche d'images basée sur la sélection des caractéristiques. Les expérimentations ont été effectuées sur les bases de données d'images couleurs connues WANG et ZuBud.

Nous présentons dans les pages qui suivent, un article intitulé **Logistic Regression**

Models for a Fast CBIR Method Based on Feature Selection qui a été publié dans les actes du **Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)** qui a eu lieu à Hyderabad (Inde) en 2007.

Logistic Regression Models for a Fast CBIR Method Based on Feature Selection

R. Ksantini[1], D. Ziou[1], B. Colin[2] and F. Dubeau[2]

(1) Département d'informatique, Faculté des sciences
Université de Sherbrooke
Sherbrooke, QC, Canada J1K 2R1.
Email: riadh.ksantini@usherbrooke.ca
djemel.ziou@usherbrooke.ca
(2) Département de mathématiques, Faculté des sciences
Université de Sherbrooke
Sherbrooke, QC, Canada J1K 2R1.
Email: bernard.colin@usherbrooke.ca
francois.dubeau@usherbrooke.ca

Keywords: Image Retrieval, Logistic Regression models, Variational Method, Weighted Pseudo-Metric, Feature Selection.

Abstract

Distance measures like the Euclidean distance have been the most widely used to measure similarities between feature vectors in the content-based image retrieval (CBIR) systems. However, in these similarity measures no assumption is made about the probability distributions and the local relevances of the feature vectors. Therefore, irrelevant features might hurt retrieval performance. Probabilistic approaches have proven to be an effective solution to this CBIR problem. In this paper, we use a Bayesian logistic regression model, in order to compute the weights of a pseudo-metric to improve its discriminatory capacity and then to increase image retrieval accuracy. The pseudo-metric weights were adjusted by the classical logistic regression model in [Ksantini *et al.*, 2006]. The Bayesian logistic regression model was shown to be a significantly better tool than the classical logistic regression one to improve the retrieval performance. The retrieval method is fast and is based on feature selection. Experimental results are reported on the Zubud and WANG color image databases proposed by [Deselaers *et al.*, 2004].

1 Introduction

The rapid expansion of the Internet and the wide use of digital data in many real world applications in the field of medecine, security, communications, commerce and academia, increased the need for both efficient image database creation and retrieval procedures. For this reason, content-based image retrieval (CBIR) approach was proposed. In this approach, each image from the database is associated with a feature vector capturing certain visual features of the image such as color, texture and shape. Then, a similarity measure is used to compare these feature vectors and to find similarities between images with the assumption that images that are close to each other in the feature space are also visually similar. Distance measures like the Euclidean distance have been the most widely used for feature vector comparison in the CBIR systems. However, these similarity measures are only based on the distances between feature vectors in the feature space. Therefore, because of the lack of information about the relative relevances of the featurebase feature vectors and because of the noise in these vectors, distance measures can fail and irrelevant features might hurt retrieval performance. Probabilistic approaches are a promising solution to this CBIR problem, that when compared to the standard CBIR methods based on the distance measures, can lead to a significant gain in retrieval accuracy. In fact, these approaches are capable of generating probabilistic similarity measures and highly customized metrics for computing image similarity based on the consideration and distinction of the relative feature vector relevances. As to previous works based on these probabilistic approaches, [Peng *et al.*, 2004] used a binary classification to classify the database color image feature vectors as relevant or irrelevant, [Caenen and Pauwels, 2002] used the classical quadratic logistic regression model, in order to classify database image feature vectors as relevant or irrelevant, [Aksoy *et al.*, 2000] used weighted L_1 and L_2 distances, in order to measure the similarity degree between two images and [Aksoy and Haralick, 2001] measure the similarity degree between a query image and a database image using a likelihood ratio derived from a Bayesian classifier.

In this paper, we investigate the effectiveness of a Bayesian logistic regression model based on a variational method, in order to adjust the weights of a pseudo-metric used in [Ksantini *et al.*, 2006], and then to improve its discriminatory capacity and to increase image retrieval accuracy. This pseudo-metric makes use of the compressed and quantized versions of the Daubechies-8 wavelet decomposed feature vectors, and its weights were adjusted by the classical logistic regression. We will show that thanks to the variational method, the used Bayesian logistic regression model is a significantly better tool than the classical logistic regression model to compute the pseudo-metric weights and to improve the querying results. The retrieval method is fast, efficient and based on feature selection. The evaluation of the retrieval method using both models, separately, is performed using precision and scope curves as defined in [Kherfi and Ziou, 2006].

In the next section, we briefly define the pseudo-metric. In section 3, we briefly describe the pseudo-metric weight computation using the classical logistic regression model, while showing the limitations of this latter and that the Bayesian logistic regression model is more appropriate for the pseudo-metric weight computation. Then, we detail the Bayesian logistic regression model. Moreover, we will describe the data training performed for both models. The feature selection based image retrieval method and the feature vectors used to represent the database images are presented in section 4. Finally, in section 5, we will perform some experiments to validate the Bayesian logistic regression model and we will use the precision and scope, in order to show the advantage of the Bayesian logistic regression model over the classical logistic regression one, in terms of querying results.

2 The pseudo-metric

Given a query feature vector Q and a featurebase of $|DB|$ feature vectors T_k $(k = 1, ..., |DB|)$ having 2^J components each, our aim is to retrieve in the featurebase the most similar feature vectors to Q. To achieve this, Q and the $|DB|$ feature vectors are Daubechies-8 wavelets decomposed, compressed to m coefficients each and quantized. Then, to measure the similarity degree between Q and a target feature vector T_k of the featurebase, we use the one-dimensional version of the pseudo-metric used in [Ksantini *et al.*, 2006] and given by the following expression

$$\| Q, T_k \| = \tilde{w}_0 |\tilde{Q}[0] - \tilde{T}_k[0]| - \sum_{i:\tilde{Q}_q^c[i] \neq 0} w_{bin(i)} (\tilde{Q}_q^c[i] = \tilde{T}_{kq}^c[i]), \tag{1}$$

where

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

$\tilde{Q}[0]$ and $\tilde{T}_k[0]$ are the scaling factors of Q and T_k, $\tilde{Q}_q^c[i]$ and $\tilde{T}_{kq}^c[i]$ represent the i-th coefficients of their Daubechies-8 wavelets decomposed, compressed to m coefficients and quantized versions, $\tilde{w}_0$ and the $w_{bin(i)}$'s are the weights to compute, and the bucketing function $bin()$ groups these latters according to the J resolution levels, such as

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

3 The weight computation

In order to improve the discriminatory power of the pseudo-metric, we compute its weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ using a classical logistic regression model and a Bayesian logistic regression model, separately. We define two classes, the relevance class denoted by Ω_0 and the irrelevance class denoted by Ω_1, in order to classify the feature vector pairs as similar or dissimilar. The basic principle of using the Bayesian logistic regression model and the classical logistic regression one is to allow a good linear separation between Ω_0 and Ω_1, and then to compute the weights which represent the local relevances of the pseudo-metric components.

3.1 The classical logistic regression model

In this model, each feature vector pair is represented by an explanatory vector and a binary target variable. Specifically, for the i-th feature vector pair, we associate an explanatory vector $X_i = (\tilde{X}_{0,i}, X_{0,i}, ..., X_{J-1,i}, 1) \in \mathbb{R}^J \times \{1\}$ and a binary target S_i which is either 0 or 1, depending on whether or not the two feature vectors are intended to be similar. $\tilde{X}_{0,i}$ is the absolute value of the difference between the scaling factors of the Daubechies-8 wavelets decomposed, compressed and quantized versions of the two feature vectors and $\{X_{k,i}\}_{k=0}^{J-1}$ are the numbers of mismatches between the J resolution level coefficients of these latter. We suppose that we have n_0 pairs of similar feature vectors and n_1 pairs of dissimilar ones. Thus, the class Ω_0 contains n_0 explanatory vectors and their associated binary target variables $\{X_i^r, S_i^r = 0\}_{i=1}^{n_0}$ to represent the pairs of the similar feature vectors, and the class Ω_1 contains n_1 explanatory vectors and their associated binary target variables $\{X_j^{ir}, S_j^{ir} = 1\}_{j=1}^{n_1}$ to represent the pairs of the dissimilar feature vectors. The pseudo-metric weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ and an intercept v are chosen to optimize the following conditional log-likelihood.

$$L(\tilde{w}_0, w_0, ..., w_{J-1}, v) = \sum_{i=1}^{n_0} log(p_i^r) + \sum_{j=1}^{n_1} log(p_j^{ir}), \tag{4}$$

where p_i^r and p_j^{ir} are the relevance and irrelevance probabilities, respectively, and given by

$$\begin{aligned} p_i^r &= F(-\tilde{w}_0 \tilde{X}_{0,i}^r - \sum_{k=0}^{J-1} w_k X_{k,i}^r - v), \\ p_j^{ir} &= F(\tilde{w}_0 \tilde{X}_{0,j}^{ir} + \sum_{k=0}^{J-1} w_k X_{k,j}^{ir} + v), \end{aligned}$$

where $F(x) = \frac{e^x}{1+e^x}$ is the logistic function. For this reason, standard optimization algorithms such as Fisher scoring and gradient ascent algorithms [Clogg *et al.*, 1991], can be invoked. However, in several cases, especially because of the exponential in the likelihood function or because of the existence of many zero explanatory vectors, the maximum likelihood can fail and estimates of the parameters of interest (weights and intercept) may not be optimal or may not exist or may be on the boundary of the parameter space. Also, as there is complete or quasicomplete separation between Ω_0 and Ω_1, the function L is made arbitrarily large and standard optimization algorithms diverge [Krishnapuram *et al.*, 2005]. Moreover, as Ω_0 and Ω_1 are large and high-dimensional, these standard optimization algorithms have high computational complexity and take long time to converge. The first two problems can be solved by smoothing the parameter of interest estimates, assuming a certain prior distribution for the parameters, thereby reducing the parameter space, and the third problem can be solved by using variational transformations which simplify the computation of the parameter of interest estimates [Jaakkola and Jordan, 2000]. This motivates the adoption of a Bayesian logistic regression model based on variational methods.

3.2 The Bayesian logistic regression model

In the Bayesian logistic regression framework, there are three main components which are 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. Among many priors having different distributional forms, gaussian prior has the advantage of having low computational intensity and of smoothing the parameter estimates toward a fixed mean and away from unreasonable extremes. However, when the likelihood function is not conjugate of the gaussian prior, the posterior distribution has no tractable form and its mean computation involves high-dimensional integration which has high computational cost. According to [Jaakkola and Jordan, 2000], it's possible to use accurate variational transformations in order to approximate the likelihood function with a simpler tractable exponential form. In this case, thanks to the conjugacy, with a gaussian prior distribution over the parameters of interest combined 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 updated to optimize the posterior distribution approximation is also large, thereby the computational cost is high. In the Bayesian logistic regression model that we propose, we use variational transformations and the Jensen's inequality in order to approximate the likelihood function with tractable exponential form. The explanatory vectors are not observed but instead are distributed according to two specific distributions. The posterior distribution is also approximated with a gaussian which depends only on two variational parameters. The computation of the posterior distribution approximation mean is fast and has low computational complexity. In this model, we 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 $\underline{\mathrm{X}}_0 = (\tilde{\underline{\mathrm{X}}}_{0,0}, \underline{\mathrm{X}}_{0,0}, ..., \underline{\mathrm{X}}_{J-1,0}, 1)$ and $\underline{\mathrm{X}}_1 = (\tilde{\underline{\mathrm{X}}}_{0,1}, \underline{\mathrm{X}}_{0,1}, ..., \underline{\mathrm{X}}_{J-1,1}, 1)$, 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. For $\underline{\mathrm{X}}_0$ 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 for $\underline{\mathrm{X}}_1$ 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. Parameters of interest (weights and intercept) are considered as random variables and are denoted by the random vector $\underline{\mathrm{W}} = (\underline{\tilde{w}}_0, \underline{w}_0, ..., \underline{w}_{J-1}, \underline{v})$. We assume that $\underline{\mathrm{W}} \sim \pi(\underline{\mathrm{W}})$, where π is a gaussian prior with prior mean μ and covariance matrix Σ. Using Bayes' rule, the posterior distribution over $\underline{\mathrm{W}}$ is given by

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

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 $i \in \{0,1\}$. Using a variational approximation [Jaakkola and Jordan, 2000] and the Jensen's inequality, the posterior distribution is approximated as follows

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

where

$$\underline{\mathrm{P}}(\underline{\mathrm{W}}|\underline{\mathrm{S}}_0 = 0, \underline{\mathrm{S}}_1 = 1, \{\epsilon_i\}_{i=0}^{1}, \{q_i\}_{i=0}^{1}) = \left[\prod_{i=0}^{1} F(\epsilon_i)\right] e^{\left[\sum_{i=0}^{1} \left[\frac{E_{q_i}[H_i] - \epsilon_i}{2}\right] - \sum_{i=0}^{1} \left[\varphi(\epsilon_i)\left(E_{q_i}[H_i^2] - \epsilon_i^2\right)\right]\right]},$$

where E_{q_0} and E_{q_1} are the expectations with respect to the distributions q_0 and q_1, respectively, $\varphi(\epsilon_i) = \frac{tanh(\frac{\epsilon_i}{2})}{4\epsilon_i}$ and $\{\epsilon_i\}_{i=0}^{1}$ are the variational parameters. Therefore, the approximation of the posterior distribution is considered as an adjustable lower bound and as a proper Gaussian distribution with a posterior mean μ_{post} and covariance matrix Σ_{post} which are estimated by the following Bayesian update equations

$$\begin{aligned}(\Sigma_{post})^{-1} &= (\Sigma)^{-1} + 2\sum_{i=0}^{1}\left[\varphi(\epsilon_i)E_{q_i}[x_i(x_i)^t]\right], && (5)\\ \mu_{post} &= \Sigma_{post}\left[(\Sigma)^{-1}\mu + \sum_{i=0}^{1}\left[(i-\frac{1}{2})E_{q_i}[x_i]\right]\right]. && (6)\end{aligned}$$

The weight and intercept computation algorithm is in two phases. The first phase is the initialization of q_0, q_1 and the gaussian prior $\pi(\underline{W})$, and the second phase is iterative and allows the computation of Σ_{post} and μ_{post} through the Bayesian update equations (5) and (6), respectively, while using an EM type algorithm [Jaakkola and Jordan, 2000], in order to find the variational parameters $\{\epsilon_i\}_{i=0}^{1}$ at each iteration to have an optimal approximation to the posterior distribution. In the initialization phase, q_0 and q_1 are chosen to model Ω_0 and Ω_1, respectively, and because of the absence of prior knowledge about the weights and the intercept, $\pi(\underline{W})$ is chosen univariate with zero mean and large variances [Congdon, 2001]. The values of μ_{post} components are the desired estimates of the pseudo-metric weights $\tilde{w}_0$ and $\{w_k\}_{k=0}^{J-1}$ and the intercept v. Once the parameters of the posterior distribution approximation are computed, its magnitude is given by the term $\prod_{i=0}^{1} F(\epsilon_i)$. This latter becomes very close to 1 as Ω_0 and Ω_1 are linearly separated or quasi separated and tends towards 0 as Ω_0 and Ω_1 become more and more overlapped. Analogically, in the classical logistic regression model, the term e^{2L} has almost the same characteristics as $\prod_{i=0}^{1} F(\epsilon_i)$ [Caenen and Pauwels, 2002]. These two terms will be used to perform feature selection in the retrieval method.

3.3 Training

Let us consider a color image database which consists of several color image sets such that each set contains color images which are perceptually close to each other in terms of object shapes and colors. In order to compute the pseudo-metric weights and the intercept by the classical logistic regression model, we have to construct the relevance class Ω_0 and the irrelevance class Ω_1. To construct Ω_0, we draw all possible pairs of feature vectors representing color images belonging to the same database color image sets, and for each pair we compute an explanatory vector and we associate to this latter a binary target variable equal to 0. Similarly, to construct Ω_1, we draw all possible pairs of feature vectors representing color images belonging to different database color image sets, and for each pair we compute an explanatory vector and we associate to this latter a binary target variable equal to 1. For the Bayesian logistic regression model, we construct the Ω_0 and Ω_1 with the same way, but instead of associating a binary target variable value to each explanatory vector of Ω_0 and Ω_1, we associate a binary target variable $\underline{\mathrm{S}}_0$ equal to 0 to all Ω_0 explanatory vectors and we associate a binary target variable $\underline{\mathrm{S}}_1$ *equal to 1 to all* Ω_1 *explanatory vectors.*

4 Color image retrieval method

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

4.1 Color image database preprocessing

We detail the preprocessing phase done once for all the database color images before the querying in a general case by the following steps.

1. Choose N feature vectors for comparison.

2. Compute the N feature vectors T_{li} $(l \in \{1, ..., N\})$ for each i-th color image of the database, where $i \in \{1, ..., |DB|\}$.

3. The feature vectors representing the database color images are Daubechies-8 wavelets decomposed, compressed to m coefficients each and quantized.

4. Organize the decomposed, compressed and quantized feature vectors into search arrays Θ_+^l and Θ_-^l $\left(l = 1, ..., N\right)$ which are used to optimize the pseud-metric computation process [Ksantini *et al.*, 2006].

5. Adjustment of the metric weights $\tilde{w}_0^l$ and $\{w_k^l\}_{k=0}^{J-1}$ for each featurebase T_{li} $(i = 1, ..., |DB|)$ representing the database color images, where $l \in \{1, ..., N\}$.

4.2 The querying algorithm

We detail the querying algorithm in a general case by the following steps.

1. Given a query color image, we denote the feature vectors representing the query image by Q_l $\left(l = 1, ..., N\right)$.

2. The feature vectors representing the query image are Daubechies-8 wavelets decomposed, compressed to m coefficients each and quantized.

3. The similarity degrees between Q_l $(l = 1, ..., N)$ and the database color image feature vectors T_{li} $(l = 1, ..., N)$ $(i = 1, ..., |DB|)$ are represented by the arrays $Score_l$ $(l = 1, ..., N)$ such that $Score_l[i] =\| Q_l, T_{li} \|$ for each $i \in \{1, ..., |DB|\}$. These arrays are returned by the procedure Retrieval(Q_l, m, Θ_+^l, Θ_-^l) $(l = 1, ..., N)$, respectively. The procedure Retrieval is used to optimize the querying process [Ksantini *et al.*, 2006].

4. The similarity degrees between the query color image and the database color images are represented by a resulted array $TotalScore$, such as, $TotalScore[i] = \sum_{l=1}^{N} \gamma_l Score_l[i]$ for each $i \in \{1, ..., |DB|\}$, where $\{\gamma_l\}_{l=1}^{N}$ are weightfactors used to down-weight the feature which has low discriminatory power. $\gamma_l = e^{2L_l}$ when the weights are computed by the classical logistic regression model, and $\gamma_l = \prod_{i=0}^{1} F(\epsilon_i^l)$ when the weights are computed by the Bayesian logistic regression model.

5. Organize the database color images in order of increasing resulted similarity degrees of the array $TotalScore$. The most negative resulted similarity degrees correspond to the closest target images to the query image. Finally, return to the user the closest target color images to the query color image and whose number is denoted by RI and chosen by the user.

4.3 Used feature vectors

In order to describe the luminance, colors and the edges of a color image, we use luminance histogram and weighted histograms. The image texture description is performed by kurtosis and skewness histograms. Given an $M \times N$ pixel LAB color image, its luminance histogram h_L contains the number of pixels of the luminance L, and can be written as follows

$$h_L(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_L(i,j) - c), \tag{7}$$

for each $c \in \{0, ..., 255\}$, where I_L is the luminance image and δ is the Kronecker symbol at 0. The weighted histograms are the color histogram constructed after edge region elimination and the multispectral gradient module mean histogram. The former is given by

$$h_k^h(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k(i,j) - c)\chi_{[0,\eta]}\Big(\lambda_{max}(i,j)\Big), \tag{8}$$

and the latter is given by

$$\bar{h}_k^e(c) = \frac{h_k^e(c)}{N_{p,k}(c)}, \tag{9}$$

where $N_{p,k}(c)$ is the number of the edge region pixels and is defined as

$$N_{p,k}(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k(i,j) - c) \chi_{]\eta,+\infty[}\Big(\lambda_{max}(i,j)\Big), \tag{10}$$

and

$$h_k^e(c) =$$

$$\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k(i,j) - c) \lambda_{max}(i,j)\ \chi_{]\eta,+\infty[}\Big(\lambda_{max}(i,j)\Big), \tag{11}$$

for each $c \in \{0, ..., 255\}$ and $k = a, b$, where λ_{max} represents the multispectral gradient module [Ksantini *et al.*, 2006], η is a threshold defined by the mean of the multispectral gradient modules computed over all image pixels, I_a and I_b are the images of the chrominances a red/green and b yellow/blue, respectively, and χ is the characteristic function. The multispectral gradient module mean histogram provides information about the overall contrast in the chrominance and the edge region elimination allows the avoidance of overlappings or noises between the color histogram populations caused by the edge pixels. The LAB color image kurtosis and skewness histograms are given by

$$h_k^{\kappa}(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k^{\kappa}(i,j) - c), \tag{12}$$

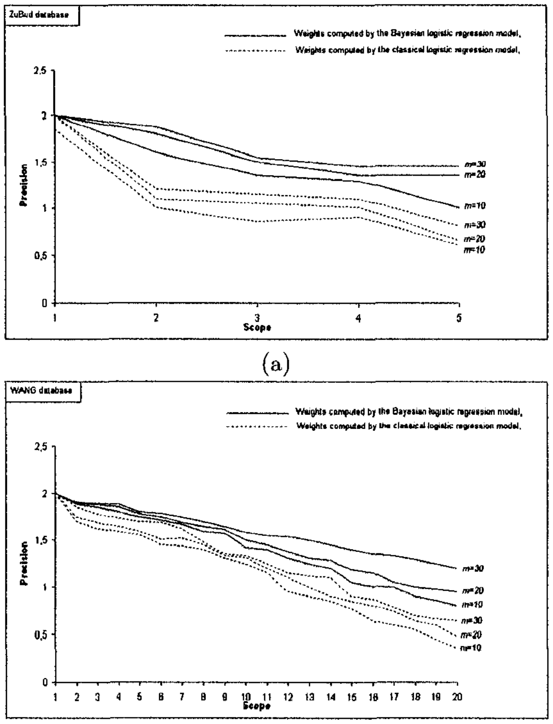
and

$$h_k^{s}(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I_k^{s}(i,j) - c), \tag{13}$$

respectively, for each $c \in \{0, ..., 255\}$ 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 the skewness images of these latter. They are obtained by local computations of the kurtosis and skewness values at the luminance and chrominance image pixels. Then, a linear interpolation is used to represent the kurtosis and skewness values between 0 and 255. Since each used feature vector is a histogram having 256 components, we set J equal to 8 in the following section.

5 Experimental results

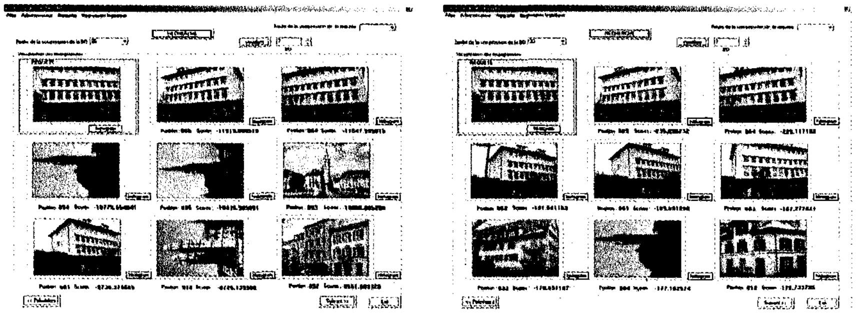
In this section, we will discuss the choices of the distributions q_0 and q_1, in order to validate the Bayesian logistic regression model in the image retrieval context. Finally, we will use the precision and scope as defined in [Kherfi and Ziou, 2006], to evaluate the querying method using both models separately. The choices of the distributions q_0 and q_1 and the querying evaluation will be conducted on the WANG and Zubud color image databases proposed by [Deselaers *et al.*, 2004]. The WANG database contains $|DB| = 1000$ color images which were selected manually to form 10 sets (e.g. Africa, beach, ruins, food) of 100 images each. The Zurich Building Image Database (ZuBuD) contains a training part of $|DB| = 1005$ color images and query part of 115 color images. The training part consists of 201 building image sets, where each set contains 5 color images of the same building taken from different positions. Before the feature vector extractions, we represent the WANG and Zubud database color images in the perceptually uniform LAB color space. Since from each color image of the Zubud and WANG databases we extract $N = 11$ histograms which are given by (7), (8), (9), (12) and (13) respectively, each database is represented by eleven featurebases. The choices of q_0 and q_1 will be separately performed for each featurebase. For each featurebase, we assume that $\tilde{\underline{\mathrm{X}}}_{0,0}$ and $(\underline{\mathrm{X}}_{0,0}, ..., \underline{\mathrm{X}}_{J-1,0})$ are independent. We make the same assumption for $\tilde{\underline{\mathrm{X}}}_{0,1}$ and $(\underline{\mathrm{X}}_{0,1}, ..., \underline{\mathrm{X}}_{J-1,1})$. Moreover, we suppose that the random vector $(\underline{\mathrm{X}}_{0,0}, ..., \underline{\mathrm{X}}_{J-1,0})$ random variables whose realizations are positive integers, are independent and each one of them follows a truncated poisson distribution at its greatest realization, to have a best fit. Analogically, we make the same choice for $(\underline{\mathrm{X}}_{0,1}, ..., \underline{\mathrm{X}}_{J-1,1})$. Also, we assume that the random variable $\tilde{\underline{\mathrm{X}}}_{0,0}$ whose realizations are positive reals, follows a gaussian mixture distribution, which is the same choice for $\tilde{\underline{\mathrm{X}}}_{0,1}$. Generally, to carry out an evaluation in the image retrieval field, two principal issues are required: the acquisition of ground truth and the definition of performance criteria. For ground truth, we use human observations. In fact, three external persons participate in the below evaluation. Concerning performance criteria, we represent the evaluation results by the precision-scope curve $Pr = f(RI)$, where the scope RI is the number of images returned to the user. In each querying performed in the evaluation experiment, each human subject is asked to give a goodness score to each retrieved image. The goodness score is 2 if the retrieved image is almost similar to 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 precision is computed as follows: $Pr =$ the sum of goodness scores for retrieved images$/RI$. Therefore, the curve $Pr = f(RI)$ gives the precision for different values of RI which lie between 1 and 20 when we perform the querying evaluation on the WANG database, and lie between 1 and 5 when we perform the querying evaluation on the ZuBuD database. When the human subjects perform different queryings in the evaluation experiment, we compute an average precision for each value of RI, and then we construct the precision-scope curve. In our evaluation experiment, each color image of the WANG and Zubud databases is represented by $N = 11$ histograms which are 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. In order to evaluate the querying in the WANG database, each human subject is asked to formulate a query from the database and to execute a querying, using weights computed by the classical logistic regression model, and to give a goodness score to each retrieved image, then to reformulate a query from the database and to execute the querying, using weights computed by the Bayesian logistic regression model, and to give a goodness score to each retrieved image. Each human subject performs the querying fifty times by choosing a new query from the database each time. We repeat this experience for different orders of compression $m \in \{30, 20, 10\}$. To evaluate the querying in the ZuBuD database, each human subject is asked to follow the preceding steps, while formulating the queries from the database query part. For the WANG and Zubud databases, the resulted precision-scope curves are given in Figure 1 for compression orders $m \in \{30, 20, 10\}$. The Figure 2 illustrates two retrieval examples in the Zubud database comparing the performances of the regression models for $m = 30$. In each example the query is located at the top-left of the dialog box.



(a)

(b)

Figure 1: Evaluation ((a) ZuBud database and (b) WANG database): precision-scope curves for retrieval using weights computed by the classical logistic regression model and weights computed by the Bayesian logistic regression model.



(a)

(b)

Figure 2: Comparison (ZuBud database): a) first 8 color images retrieved using weights computed by the classical logistic regression model, b) first 8 color images retrieved using weights computed by the Bayesian logistic regression model.

6 Conclusion

We presented a simple, fast and effective color image querying method based on feature selection. In order to measure the similarity degree between two color images both quickly and effectively, we used a weighted pseudo-metric which makes use of the one-dimensional Daubechies decomposition and compression of the extracted feature vectors. A Bayesian logistic regression model and a classical logistic regression one were used to improve the discriminatory capacity of the pseudo-metric and to allow feature selection. Evaluations of the querying method showed that the Bayesian logistic regression model is a better tool than the classical logistic regression one to compute the pseudo-metric weights and to improve the querying results.

References

[Aksoy and Haralick, 2001] S. Aksoy and R. M. Haralick. Feature Normalization and Likelihood-based Similarity Measures for Image Retrieval. *Pattern Recognition Letters*, 22(5):563-582, 2001.

[Aksoy *et al.*, 2000] S. Aksoy, R. M. Haralick, F. A. Cheikh, and M. Gabbouj. A Weighted Distance Approach to Relevance Feedback. In *15th International Conference on Pattern Recognition*, page 4812, Barcelona, Spain, 2000.

[Caenen and Pauwels, 2002] G. Caenen and E. J. Pauwels. Logistic Regression Models for Relevance Feedback in Content-Based Image Retrieval. In *Storage and Retrieval for Media Databases 2002, Proceedings of SPIE*, pages 49-58, San Jose, California, USA, 2002.

[Clogg *et al.*, 1991] 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. *Journal of the American Statistical Association*, 86:68-78, 1991.

[Congdon, 2001] P. Congdon. *Bayesian Statistical Modelling.* John Wiley, Chichester, UK, 2001.

[Deselaers *et al.*, 2004] T. Deselaers, D. Keysers, and H. Ney. Classification Error Rate for Quantitative Evaluation of Content-based Image Retrieval Systems. In *17th International Conference on Pattern Recognition*, pages 505-508, Cambridge, UK, 2004.

[Jaakkola and Jordan, 2000] T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25-37, 2000.

[Kherfi and Ziou, 2006] M. L. Kherfi and D. Ziou. Relevance feedback for CBIR: a new approach based on probabilistic feature weighting with positive and negative examples. *IEEE Transactions on Image Processing*, 15(4):1017-1030, 2006.

[Krishnapuram *et al.*, 2005] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink. Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):957-968, 2005.

[Ksantini *et al.*, 2006] R. Ksantini, D. Ziou, and F. Dubeau. Image Retrieval Based on Region Separation and Multiresolution Analysis. *International Journal of Wavelets, Multiresolution and Information Processing*, 4(1):147-175, 2006.

[Peng *et al.*, 2004] J. Peng, B. Bhanu, and S. Qing. Learning Feature Relevance and Similarity Metrics in Image Databases. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 14-18, Santa Barbara, California, USA, 2004.

CHAPITRE 5

Modèle bayésien d'analyse discriminante logistique fondé sur les noyaux : une amélioration à l'analyse discriminante de Fisher fondée sur les noyaux

L'analyse discriminante de Fisher basée sur les noyaux (KFD) est un classificateur nonlinéaire qui s'est avéré puissant et se comparant avantageusement à plusieurs classificateurs existants. Elle est équivalente à l'analyse discriminante linéaire de Fisher appliquée efficacement dans l'espace des noyaux. Cependant, elle suppose que les matrices de covariance des classes transformées dans l'espace des noyaux soient identiques, ce qui n'est pas pas le cas dans de nombreuses applications. Dans ce chapitre, nous proposons un modèle bayésien d'analyse discriminante logistique basé sur les noyaux (BKLD) qui représente chaque classe transformée par sa propre matrice de covariance.

Ceci peut mener à plus de flexibilité et à de meilleures performances de classification que la KFD. Une comparaison extensive du BKLD à la KFD et à d'autres classificateurs nonlinéaire présents dans a littérature est effectuée. De plus, l'analyse de la complexité de l'algorithme et les performances numériques du BKLD sont détaillées. Nous présentons dans les pages qui suivent, un article intitulé **A Bayesian Kernel Logistic Discriminant Model : An Improvement to the Kernel Fisher's Discriminant Model**. Ce travail sera enrichi d'une application de recherche d'images où de détection d'objets (peau, feux, ombre, etc) et sera soumis au journal **IEEE Transactions on Knowledge and Data Engineering (TKDE)**.

A Bayesian Kernel Logistic Discriminant Model: An Improvement to the Kernel Fisher's Discriminant Model

R. Ksantini[1], D. Ziou[1], B. Colin[2], and F. Dubeau[2]

(1) Département d'informatique, Faculté des sciences
Université de Sherbrooke
Sherbrooke, Qc, Canada J1K 2R1.
Email: riadh.ksantini@usherbrooke.ca
djemel.ziou@usherbrooke.ca
(2) Département de mathématiques, Faculté des sciences
Université de Sherbrooke
Sherbrooke, Qc, Canada J1K 2R1.
Email: bernard.colin@usherbrooke.ca
francois.dubeau@usherbrooke.ca

Keywords: Kernel Fisher's Discriminant, Bayesian Kernel Logistic Discriminant Model, Variational Method, Kernel Basis.

Abstract

The Kernel Fisher's Discriminant (KFD) is a non-linear classifier which has proven to be powerful and competitive to several state-of-the-art classifiers. Its main ingredient is the kernel trick which allows the efficient computation of Fisher's Linear Discriminant in feature space. However, it is assuming equal covariance structure for all transformed classes, which is not true in many applications. In this paper, we propose a novel Bayesian Kernel Logistic Discriminant model (BKLD) which goes one step further by representing each transformed class by its own covariance matrix. This can allow more flexibility and better classification performances than the KFD. The posterior distribution of the BKLD model is elegantly approximated by a tractable Gaussian form using variational transformation and Jensen's inequality, which allow a straightforward computation of the weights. An extensive comparison of the BKLD to the KFD and to other state-of-the-art non-linear classifiers is performed. Also, analysis of algorithm complexity and numerical accuracy is provided.

1 Introduction

In supervised learning we are given a training set of input vectors $\{X_i\}_{i=1}^N$, where $X_i \in \mathbb{R}^k (k \geq 1)$ $\forall i \in \{1, 2, ..., N\}$, along with corresponding tags $\{t_i\}_{i=1}^N$, where $t_i \in \mathbb{N}$ $\forall i \in \{1, 2, ..., N\}$, the latter of which might be class labels in classification. From this training set, we wish to learn a model of the dependency of the targets on the inputs with the objective of making accurate predictions of t for unseen values of X. In real-world data, the presence of class overlap in classification implies that the principal modelling challenge is to avoid over-fitting of the training set. Typically, we base our predictions upon some function $y(X)$ defined over the input space (or training space) $\mathcal{X}$, and learning is the process of inferring the parameters or weights of this function. In order to learn non-linear relations with a linear classifier, we need to select a set of non-linear features and to rewrite the data in the new representation. This is equivalent to applying a fixed non-linear mapping of the data to a feature space $\mathcal{F}$, in which the linear classifier can be used. Hence, the set of hypothesis we consider will be functions of the type

$$y(X; \mathbf{w}) = \sum_{i=1}^{l} w_i \phi_i(X) + w_0 = \mathbf{w}^T \Phi(X), \tag{1}$$

where $\Phi(X) = (1, \phi_1(X), \phi_2(X), ..., \phi_l(X)) : \mathcal{X} \to \mathcal{F}$ is a non-linear map from the input space to some feature space [1]. This means that we will build non-linear classifiers in two steps: first a fixed non-linear mapping transforms the data into a feature space $\mathcal{F}$, and then a linear classifier is used to classify them in the feature space. Analysis of functions of the type (1) is facilitated since the adjustable parameters or weights $\mathbf{w} = (w_0, w_1, w_2, ..., w_l)$ appear linearly, and the objective is to estimate good values for those parameters. While the range of functions of the type (1) that we can address is extremely broad, we concentrate here on functions of the type corresponding to those implemented by some relevant state-of-the-art linearly-parameterized models, the Support Vector Machine (SVM) [3] and the Kernel Fisher's Discriminant (KFD) [2], [4]. The SVM and KFD make predictions based on the function

$$y(X; \mathbf{w}) = \sum_{i=1}^{N} w_i \mathcal{K}(X, X_i) + w_0, \tag{2}$$

where $\phi_i(X) = \mathcal{K}(X, X_i)$ is a kernel function, effectively defining one basis function for each observation in the training set. The use of kernel trick impacts linear decision boundary in the feature space, while implicitly yielding a flexible non-linear separation in the input space. The KFD was firstly proposed by Mika *et al.* [2] and its main idea is to perform the traditional Fisher's linear discriminant in the feature space. Moreover, it has proven to be powerful and competitive to several non-linear classifiers [2]. Subsequently, a number of KFD algorithms [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], have been developed. However, these KFD-based algorithms suffer from the small sample size problem since the kernel-induced feature space is typically of very high dimensionality. Furthermore, they are incapable of dealing with heteroscedastic data (classes with different covariance matrices) that are commonly found in real-world applications. Many methods have been proposed to address the small sample size problem. Mika *et al.* [2] proposed adding a small multiple of the identity matrix to make the inner product matrix invertible. Baudat and Anouar [11] and Xiong *et al.* [12] used QR decomposition to avoid the singularity of the inner product matrix. Park *et al.* [13] proposed the KFD/GSVD algorithm by employing generalized singular value decomposition (GSVD). Yang [14] adopted the technique introduced in Fisherfaces [20], i.e., kernel Fisherfaces. Lu *et al.* [15] proposed the kernel direct discriminant analysis (KDDA) algorithm based on generalization of the LDA algorithm in [21]. Recently, Dai and Qian [16], [17] presented a further enhanced method called the kernel generalized nonlinear discriminant analysis (KGNDA) algorithm which is based on the theoretical foundation established in [18]. More specifically, it attempts to exploit the crucial discriminatory information in the null space of the within-class scatter matrix in the feature space $\mathcal{F}$. In order to address the heteroscedasticity problem, Dai *et al.* proposed recently a novel KFD algorithm called heteroscedastic kernel weighted discriminant analysis (HKWDA) which is based on the idea of weighted pairwise Chernoff criterion proposed in [22].

In this paper, we propose a Bayesian Kernel Logistic Discriminant model (BKLD) which is capable of dealing with heteroscedastic data by representing each transformed class by its own covariance matrix. This can allow more flexibility and better classification performances than the KFD. The objective likelihood function of our model has no tractable form. For this reason, we used variational transformation and the Jensen's inequality to approximate it with a tractable exponential form which depends only on two variational parameters. In order to avoid small sample size problem and to speed up the computation of the model parameters (or weights), we introduce a sparsity-promoting Gaussian prior over them governed by a set of prior parameters, one associated with each weight, whose most values are iteratively estimated using an expectation-maximization (EM) type algorithm. Due to the conjugacy, by combining a Gaussian prior with the likelihood approximation, we obtain a closed Gaussian form approximation to the posterior distribution of the model.

In the next section, we detail the derivation of the BKLD, and define the procedure for obtaining variational parameters and parameter values, and from them, the weights. In section 3, a comparative evaluation is performed to compare the BKLD to the KFD as well as other state-of-the-art non-linear classifiers on a collection of benchmark data sets. Furthermore, an analysis of algorithm complexity and numerical accuracy is provided. Finally, we present our conclusions.

2 The Bayesian Kernel Logistic Discriminant Model

Let $\mathcal{X}_1 = \{X_i\}_{i=1}^{N_1}$ and $\mathcal{X}_2 = \{X_i\}_{i=N_1+1}^{N}$ be two different classes constituting an input space of N samples or vectors. Applying the kernel trick, we use a function Φ to map the classes $\mathcal{X}_1$ and $\mathcal{X}_2$ to two feature classes $\mathcal{F}_1 = \{\Phi(X_i)\}_{i=1}^{N_1}$ and $\mathcal{F}_2 = \{\Phi(X_i)\}_{i=N_1+1}^{N}$, respectively, wherein $\Phi(X_i) = (1, \mathcal{K}(X_i, X_1), \mathcal{K}(X_i, X_2), ..., \mathcal{K}(X_i, X_N))$ $\forall\ i \in \{1, 2, ..., N\}$. Let us denote by $\underline{\Phi}_1$ and $\underline{\Phi}_2$ two random vectors whose realizations represent the vectors of $\mathcal{F}_1$ and the vectors of $\mathcal{F}_2$, respectively. We suppose that $\underline{\Phi}_1 \sim g_1(\underline{\Phi}_1)$ and $\underline{\Phi}_2 \sim g_2(\underline{\Phi}_2)$, where g_1 and g_2 are two Gaussian distributions whose means and covariance matrices are empirically computed from $\mathcal{F}_1$ and $\mathcal{F}_2$. With $\underline{\Phi}_1$ we associate a tag $t_1 = 0$, and with $\underline{\Phi}_2$ we associate a tag $t_2 = 1$. The unknown parameters (weights) are considered as random variables and are denoted by the random vector $\mathbf{w} = (w_0, w_1, ..., w_N)$. We define a 'likelihood' function as:

$$P(t_1 = 0, t_2 = 1|\mathbf{w}) = \sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[\prod_{i=1}^{2} P(t_i = i - 1|\underline{\Phi}_i, \mathbf{w}) g_i(\underline{\Phi}_i) \right], \tag{3}$$

where, given $F(x) = \frac{e^x}{1+e^x}$, $P(t_i = i - 1|\underline{\Phi}_i, \mathbf{w}) = F((2i - 3)\mathbf{w}^T \underline{\Phi}_i)$ $\forall\ i \in \{1, 2\}$ represent logistic modelings of t_1 and t_2 given the realizations of $\underline{\Phi}_1$ and $\underline{\Phi}_2$, respectively. The maximization of the likelihood function $P(t_1 = 0, t_2 = 1|\mathbf{w})$ with respect to the weights $\mathbf{w} = (w_0, w_1, ..., w_N)$ makes our model equivalent to the optimal linear Bayes classifier modelling $\mathcal{F}_1$ and $\mathcal{F}_2$ by g_1 and g_2, respectively. However, with as many parameters or weights in the model as training examples, we would expect the maximum-likelihood estimation of $\mathbf{w}$ from (3) to lead to severe over-fitting. To avoid this, a common approach is to impose some additional constraint on the parameters, for example, through the addition of a 'complexity' penalty term to the likelihood or error function. This is implicitly effected by the inclusion of the 'margin' term in the SVM [3] and the regularization matrix in the KFD [2]. Here, though, we adopt a Bayesian perspective, and 'constrain' the parameters by defining an explicit prior probability distribution over them. We encode a preference for smoother (less complex) functions by making the popular choice of a zero-mean Gaussian prior distribution over $\mathbf{w}$:

$$\pi(\mathbf{w}|\beta) = \prod_{i=0}^{N} \mathcal{N}(w_i|0, \beta_i^{-1}), \tag{4}$$

with $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_N)$ a vector of $N+1$ prior parameters. Importantly, there is an individual prior parameter associated independently with every weight, moderating the strength of the prior thereon. This has the advantage of promoting the sparsity of the model and thereby speeding up the computation of the weight estimates. Having defined the prior, Bayesian inference proceeds by computing, from the Bayes' rule, the posterior over the unknown weights:

$$P(\mathbf{w}|t_1 = 0, t_2 = 1) = \frac{\sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[\prod_{i=1}^{2} P(t_i = i-1|\underline{\Phi}_i, \mathbf{w}) g_i(\underline{\Phi}_i)\right] \pi(\mathbf{w}|\boldsymbol{\beta})}{P(t_1 = 0, t_2 = 1)}, \tag{5}$$

where

$$P(t_1 = 0, t_2 = 1) = \int \sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[\prod_{i=1}^{2} P(t_i = i-1|\underline{\Phi}_i, \mathbf{w}) g_i(\underline{\Phi}_i)\right] \pi(\mathbf{w}|\boldsymbol{\beta}) d\mathbf{w}, \tag{6}$$

is the normalizing term. The computation of the posterior distribution is intractable. However, we can approximate it by a variational posterior approximation with a Gaussian form, whose mean and covariance matrix computation is feasible. To obtain this variational posterior approximation, we perform two successive approximations to the likelihood function, in order to bound it by an exponential form which is a conjugate of the Gaussian prior.

First approximation:

This first approximation is based on a variational transformation of the sigmoid function $F(x)$ of the logistic regression. According to [5], the variational approximation of the sigmoid function in $H_i = (2i-3)\mathbf{w}^T \underline{\Phi}_i$ $\forall$ $i \in \{1, 2\}$ is given by

$$\begin{aligned} P(t_i = i-1|\underline{\Phi}_i, \mathbf{w}) &= F(H_i), \\ &\geq F(\epsilon_i) e^{\left[\frac{(H_i - \epsilon_i)}{2} - \varphi(\epsilon_i)\left(H_i^2 - \epsilon_i^2\right)\right]} = P(t_i = i-1|\underline{\Phi}_i, \mathbf{w}, \epsilon_i), \end{aligned} \tag{7}$$

where $\epsilon_i > 0$ is the variational parameter, $\varphi(\epsilon_i) = \frac{tanh(\frac{\epsilon_i}{2})}{4\epsilon_i}$, and $tanh(\frac{\epsilon_i}{2}) = \frac{e^{\frac{\epsilon_i}{2}} - e^{-\frac{\epsilon_i}{2}}}{e^{\frac{\epsilon_i}{2}} + e^{-\frac{\epsilon_i}{2}}}$ $\forall$ $i \in \{1, 2\}$. So the likelihood function can be approximated as follows:

$$\begin{aligned} &\sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[\prod_{i=1}^{2} P(t_i = i-1|\underline{\Phi}_i, \mathbf{w}) g_i(\underline{\Phi}_i)\right] \\ &\qquad \geq \sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[\prod_{i=1}^{2} P(t_i = i-1|\underline{\Phi}_i, \mathbf{w}, \epsilon_i) g_i(\underline{\Phi}_i)\right]. \end{aligned} \tag{8}$$

Second approximation:

The first approximation is insufficient to approximate the likelihood function 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

$$\begin{aligned} &\sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[\prod_{i=1}^{2} P(t_i = i-1 | \underline{\Phi}_i, \mathbf{w}, \epsilon_i) g_i(\underline{\Phi}_i) \right] && (9) \\ &= \left[\prod_{i=1}^{2} F(\epsilon_i) \right] \sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[e^{\left[\sum_{i=1}^{2} \left[\frac{(H_i - \epsilon_i)}{2} - \varphi(\epsilon_i) \left(H_i^2 - \epsilon_i^2 \right) \right] \right]} \prod_{i=1}^{2} g_i(\underline{\Phi}_i) \right], \\ &\geq \left[\prod_{i=1}^{2} F(\epsilon_i) \right] e^{\left[\sum_{i=1}^{2} \left[\frac{E_{g_i}[H_i] - \epsilon_i}{2} \right] - \sum_{i=1}^{2} \left[\varphi(\epsilon_i) \left(E_{g_i}[H_i^2] - \epsilon_i^2 \right) \right] \right]}, \\ &= \underline{\mathrm{P}}\big(t_1 = 0, t_2 = 1 | \mathbf{w}, \{\epsilon_i\}_{i=1}^{2}\big), \end{aligned}$$

where E_{g_1} and E_{g_2} are the expectations with respect to the distributions g_1 and g_2, respectively.

Finally, thanks to the two above approximations, the posterior distribution numerator in the formula (5) can be approximated as follows:

$$\begin{aligned} &\sum_{\underline{\Phi}_1 \in \mathcal{F}_1, \underline{\Phi}_2 \in \mathcal{F}_2} \left[\prod_{i=1}^{2} P(t_i = i-1 | \underline{\Phi}_i, \mathbf{w}) g_i(\underline{\Phi}_i) \right] \pi(\mathbf{w}|\boldsymbol{\beta}) && (10) \\ &\qquad \geq \underline{\mathrm{P}}\big(t_1 = 0, t_2 = 1 | \mathbf{w}, \{\epsilon_i\}_{i=1}^{2}\big) \pi(\mathbf{w}|\boldsymbol{\beta}). \end{aligned}$$

Thus, the variational posterior approximation denoted by $P\big(\mathbf{w}|t_1 = 0, t_2 = 1, \{\epsilon_i\}_{i=1}^{2}, \boldsymbol{\beta}\big)$ is given by normalizing the lower bound of the inequality (10). Note that although $\underline{\mathrm{P}}\big(t_1 = 0, t_2 = 1|\mathbf{w}, \{\epsilon_i\}_{i=1}^{2}\big)$ is a lower bound on the true likelihood function, our variational posterior approximation is a proper density and thus no longer a bound. Given that $\pi(\mathbf{w}|\boldsymbol{\beta})$ is a Gaussian which is a conjugate of the exponential variational form $\underline{\mathrm{P}}\big(t_1 = 0, t_2 = 1|\mathbf{w}, \{\epsilon_i\}_{i=1}^{2}\big)$, the variational posterior approximation is a Gaussian with a posterior mean μ_{post} and a posterior covariance matrix Σ_{post}. Thus, omitting the algebra, Σ_{post} and μ_{post} are given by the following Bayesian update equations:

$$(\Sigma_{post})^{-1} = A^{-1} + 2 \sum_{i=1}^{2} \left[\varphi(\epsilon_i) E_{g_i}[\underline{\Phi}_i \underline{\Phi}_i^T] \right], \quad (11)$$

$$\mu_{post} = \Sigma_{post} \left[\sum_{i=1}^{2} \left[(i - \frac{3}{2}) E_{g_i}[\underline{\Phi}_i] \right] \right], \quad (12)$$

with $A = \text{diag}(\beta_0^{-1}, \beta_1^{-1}, ..., \beta_N^{-1})$. Since for each feature class the dimensionality is greater than the number of vectors, we expect that the term $2\sum_{i=1}^{2}\left[\varphi(\epsilon_i)E_{g_i}[\underline{\Phi}_i\underline{\Phi}_i^T]\right]$ of equation (11) to be singular. However, this singularity is avoided after adding the regularization matrix A^{-1}, thereby making $(\Sigma_{post})^{-1}$ invertible. According to equation (11), Σ_{post} depends on the variational parameters $\{\epsilon_i\}_{i=1}^{2}$ and the prior parameters $\{\beta_i\}_{i=0}^{N}$, so we must specify these. We have to find the values of $\{\epsilon_i\}_{i=1}^{2}$ and $\{\beta_i\}_{i=0}^{N}$ that yield a tight lower bound in equation (10). This can be done by an EM type algorithm. More precisely, we want to find $\{\epsilon_i\}_{i=1}^{2}$ and $\{\beta_i\}_{i=0}^{N}$ that maximize the following lower bound of the log marginal likelihood

$$log\Big(\int \underline{P}(t_1 = 0, t_2 = 1|\mathbf{w}, \{\epsilon_i\}_{i=1}^{2})\pi(\mathbf{w}|\boldsymbol{\beta})d\mathbf{w}\Big). \tag{13}$$

In the EM formalism, this can be achieved by iteratively maximizing the following expectation

$$\begin{aligned}\int log\Big(\underline{P}(t_1 = 0, t_2 = 1|\mathbf{w}, \{\epsilon_i\}_{i=1}^{2})\pi(\mathbf{w}|\boldsymbol{\beta})\Big)P\big(\mathbf{w}|t_1 = 0, t_2 = 1, (\{\epsilon_i\}_{i=1}^{2})^{old}, \boldsymbol{\beta}^{old}\big)d\mathbf{w}\\ = Q\Big(\{\epsilon_i\}_{i=1}^{2}, (\{\epsilon_i\}_{i=1}^{2})^{old}, \boldsymbol{\beta}, \boldsymbol{\beta}^{old}\Big),\end{aligned}$$

with respect to $\{\epsilon_i\}_{i=1}^{2}$ and $\{\beta_i\}_{i=0}^{N}$, where $P\big(\mathbf{w}|t_1 = 0, t_2 = 1, (\{\epsilon_i\}_{i=1}^{2})^{old}, \boldsymbol{\beta}^{old}\big)$ is the variational posterior approximation based on the previous values of $\{\epsilon_i\}_{i=1}^{2}$ and $\{\beta_i\}_{i=0}^{N}$. Taking the partial derivatives of Q with respect to $\{\epsilon_i\}_{i=0}^{1}$ and $\{\beta_i\}_{i=0}^{N}$, then equalizing to zero leads to

$$\begin{aligned}\epsilon_i^2 &= E_{g_i}[\underline{\Phi}_i^T\Sigma_{post}\underline{\Phi}_i] + \mu_{post}^T\Big[E_{g_i}[\underline{\Phi}_i\underline{\Phi}_i^T]\Big]\mu_{post}, \forall i \in \{1, 2\}, &(14)\\ \beta_j &= \frac{1}{\Sigma_{post,jj} + \mu_{post,j}^2}, \forall j \in \{0, 1, ..., N\}. &(15)\end{aligned}$$

Owing to the EM formulation, each update for $\{\epsilon_i\}_{i=1}^{2}$ and $\{\beta_i\}_{i=0}^{N}$ corresponds to a monotone improvement to the variational posterior approximation [6]. The weight 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 (11) and (12), respectively, while using equations (14) and (15) to find the variational parameters and prior parameters at each iteration. The values of the μ_{post} components are the desired estimates of the weights $\{w_i\}_{i=0}^{N}$.

Initialization:

1. Compute the means and the covariance-variance matrices of the Gaussians g_1 and g_2 empirically from $\mathcal{F}_1$

and $\mathcal{F}_2$, respectively.

2. Initialize the variational parameters $\{\epsilon_i\}_{i=1}^{2}$ and the prior parameters $\{\beta_i\}_{i=0}^{N}$.

Computation of Σ_{post} and μ_{post}:

```
1. Do
        $(\Sigma_{post}^{new})^{-1} \leftarrow (A^{-1} + 2\sum_{i=1}^{2} [\varphi(\epsilon_i^{old}) E_{g_i}[\underline{\Phi}_i \underline{\Phi}_i^T]]$
        $\mu_{post}^{new} \leftarrow \Sigma_{post}^{new} \left[ \sum_{i=1}^{2} [(i - \frac{3}{2}) E_{g_i}[\underline{\Phi}_i]] \right]$
    For each $i \in \{1, 2\}$ do
        $(\epsilon_i^{new})^2 \leftarrow E_{g_i}[\underline{\Phi}_i^T \Sigma_{post}^{new} \underline{\Phi}_i] + (\mu_{post}^{new})^T \left[ E_{g_i}[\underline{\Phi}_i \underline{\Phi}_i^T] \right] \mu_{post}^{new}$
    End for
    For each $j \in \{0, 1, ..., N\}$ do
        $\beta_j^{new} \leftarrow \frac{1}{\Sigma_{post,jj}^{new} + (\mu_{post,j}^{new})^2}$
    End for
   While($|\Sigma_{post}^{old} - \Sigma_{post}^{new}| >$ threshold or $|\mu_{post}^{old} - \mu_{post}^{new}| >$ threshold)
   Return $\mu_{post}^{new}$
2. Assign the $\mu_{post}$ component values to the weights $\{w_i\}_{i=0}^{N}$.
```

The iterative phase of the above algorithm scales with the size of the training set. In fact, it is dominated by the inversion of the posterior covariance matrix which requires $\mathcal{O}((N+1)^3)$ operations at each iteration. Since the BLRM is formulated for binary or two-class problems, the "1-versus-all" approach can be used for polychotomous classification, where the number of classes is greater than two. However, in the following we will focus on two-class problems.

3 Experimental Results

3.1 Comparative Evaluation

To evaluate the performance of our new approach, we performed an extensive comparison to other state-of-the-art classifiers. The experimental setup was chosen in analogy to [7] and we compared the BKLD to the KFD, the single RBF classifier [7], the regularized AdaBoost (AB$_R$) and the SVM (with Gaussian RBF kernel $\mathcal{K}(X, X_i) = e^{-||X-X_i||^2/\sigma}$), where σ is the positive 'width' parameter. For the BKLD we used Gaussian RBF too as it has proven flexible and useful in SVM. We used 13 artificial and real word data sets from the UCI, DELVE and STATLOG benchmark repositories (except for Banana).[1] Some of the problems are originally not binary, hence a random partition into two classes was used. Then, 100 partitions into test and training set (about 60%:40%) were generated. On each of these data sets we trained and tested all classifiers (see [7] for details). The results in table 1 show the average test error over these 100 runs. The optimization of the necessary parameters (regularization parameter C for AB_R, (C, σ) for SVM, σ for the BKLD and (σ, T) for the KFD, where T is a threshold computed by the SVM [7]), were performed on the first five realizations of each data set. On each of these realizations, a 5-fold cross validation procedure gives a good model. Finally, the model parameters are computed as the median of the five estimations and used throughout the training on all 100 realizations of that data set. This way of estimating the parameters is computationally highly expensive, but makes the comparison more robust and the results more reliable [7].

[1] The data sets can be obtained via **http://www.first.gmd.de/˜raetsch/**

Table 1: Comparison among the five methods: Single RBF classifier, regularized AdaBoost, Support Vector Machine, Kernel Fisher's Discriminant and the Bayesian Kernel Logistic Discriminant: Estimation of the averages of test classification errors in % on 13 data sets (best method in bold face, second best emphasized) .

	RBF	AB_R	SVM	KFD	BKLD
Banana	*10.8*	10.9	11.5	*10.8*	**10.2**
B. Cancer	27.6	26.5	26.0	*25.8*	**25.1**
Diabetes	24.3	23.8	23.5	*23.2*	**22.7**
German	24.7	24.3	*23.6*	23.7	**23.3**
Heart	17.6	16.5	*16.0*	16.1	**15.7**
Image	3.3	**2.7**	*3.0*	4.8	3.7
Ringnorm	1.7	1.6	1.7	*1.5*	**0.8**
F. Solar	34.4	34.2	*32.4*	33.2	**30.9**
Splice	*10.0*	**9.5**	10.9	10.5	*10.0*
Thyroid	4.5	4.6	4.8	*4.2*	**4.0**
Titanic	23.3	*22.6*	**22.4**	23.2	22.7
Twonorm	2.9	2.7	3.0	*2.6*	**1.5**
Waveform	10.7	*9.8*	9.9	9.9	**8.7**

The experiments show that the BKLD is competitive or even superior to the other classifiers on almost all data sets (an exception being Image). Also, we can notice that the BKLD outperforms the KFD on all data sets as it represents each class in the feature space by its own variance-covariance matrix, which is not the case for the KFD.

3.2 Numerical Accuracy and Algorithmic Complexity

The inverse of posterior covariance matrix Σ_{post}^{-1} which, although positive definite in theory, may become numerically singular in practice (see the update equations (11) and (12)). To solve this problem, we used the Singular Value Decomposition (SVD) [8]. Generally speaking, in the Bayesian treatment, the Gaussian approximation is considered as weakness of the method as the single mode of the Gaussian at the weight estimates can often be unrepresentative of the overall posterior mass, particularly when there are multiple such modes (as is often the case). However, in our method we used variational transformations which have been shown to have much more flexibility than other approximation methods [5]. This flexibility translates into improved accuracy of the approximation. Moreover, our variational posterior distribution $P(\mathrm{w}|t_1 = 0, t_2 = 1, \{\epsilon_i\}_{i=1}^{2}, \beta)$ depends only on two variational parameters. Hence, an optimal solution can be easily obtained through re-starts with random initializations of $\{\epsilon_i\}_{i=1}^{2}$.

As optimization of the prior parameters progresses, the range of β-values typically becomes highly extended as many tend towards very large values. Indeed, many β's typically would tend to infinity if machine precision permitted. In fact, ill-conditioning of the inverse of posterior covariance matrix becomes a problem when, approximately, the ratio of the smallest to largest β-values is in the order of the machine precision. Consider the case of a single $\beta_k \rightarrow \infty$, where for convenience of presentation we choose $k = 1$, the first prior parameter. Using the expression for the inverse of a partitioned matrix, it can be shown that:

$$\Sigma_{post}^{-1} \rightarrow \left(\begin{array}{cc} 0 & 0 \\ 0 & A_{-k}^{-1} + M_{-k} \end{array} \right), \tag{16}$$

where the matrix $M = 2\sum_{i=1}^{2} \left[\varphi(\epsilon_i) E_{g_i}[\underline{\Phi}_i \underline{\Phi}_i^T]\right]$ and the subscript '$-k$' denotes the matrix with the appropriate k-th row and/or column removed. The term $A_{-k}^{-1} + M_{-k}$ is of course the inverse of posterior covariance matrix computed with basis function k pruned. Furthermore, it follows from equations (12) and (16) that $\mu_{post,k} \rightarrow 0$ and as a consequence of $\beta_k \rightarrow \infty$, the model intuitively becomes exactly equivalent to one with basis function $\mathcal{K}(X, X_k)$ excluded. We may thus choose to avoid ill-conditioning by pruning the corresponding basis function from the model at that point. This sparsification of the model during optimization implies that we typically experience a very considerable and advantageous acceleration of the learning algorithm. The disadvantage is that if we believed that the log marginal likelihood might be increased by reintroducing those deleted basis functions (i.e. reducing β_k from ∞) at a later stage, then their permanent removal would be suboptimal. So far, no such case has been found for the used data sets.

Although, typically, the pruning discussed above rapidly reduces N to a manageable size in most problems, the BKLD scales with $\mathcal{O}((N+1)^3)$ at initialization, and N may be very large. This of course leads to extended training times, although the disadvantage of this is significantly offset by the lack of necessity to perform cross-validation over regularization parameters, such as for C in the SVM and the threshold T in the KFD. So, for example, with the exception of the larger data sets (*e.g.* roughly $N > 600$) the benchmark results in table 1 were obtained more quickly for the BKLD than the SVM and KFD (this observation depends on the exact implementations and cross-validation schedules of course). Even so, for large data sets, with computation scaling approximately in $\mathcal{O}((N+1)^3)$, the full BKLD algorithm becomes prohibitively expensive to run. We have therefore developed an alternative algorithm to maximize the marginal likelihood which is constructive. It starts with a single basis function, the bias or intercept w_0, and both adds in further basis functions, or deletes current ones, as appropriate, rather than starting with all possible candidates and pruning. This is a much more efficient approach, as the number of basis functions included at any step in the algorithm tends to remain low. It is, however, a more greedy optimization strategy, although our preliminary results show little, if any, loss of accuracy compared to the standard algorithm. This appears a very promising mechanism for ensuring the BKLD remains practical even for very large basis function sets.

4 Conclusion

We have proposed an effective Bayesian Kernel Logistic Discriminant Model with prior Gaussian over the weights. The model is based on a variational approximation and on the Jensen's inequality. Thanks to these two approximations, computation of the weights has become trivial and straightforward. Our experiments showed that the Bayesian Kernel Logistic Discriminant model is competitive to other state-of-the-art non-linear classifiers, and specifically outperforms the Kernel Fisher's Discriminant on all used data sets. In fact, the advantage of the Bayesian Kernel Logistic Discriminant model over the Kernel Fisher's Discriminant is that it estimates a covariance matrix separately for each transformed class in the feature space, instead of a common matrix for all transformed classes. However, like the Kernel Fisher's Discriminant, the Bayesian Kernel Logistic Discriminant Model has always the drawback of assuming that the transformed classes in the feature space are normally distributed. This can make it ineffective if the transformed class densities are multi-model. According to the likelihood function (see equation (3)), the Bayesian Kernel Logistic Discriminant model is adaptive to any kind of distributions modelling the transformed classes. However, fitting these distributions on these classes can lead to highly biased modelisations, since in the feature space obtained using the kernel trick, the dimension exceeds the number of samples of each class. For this reason, future work will be dedicated to reduce the dimensionality of the feature space using appropriate and effective methods.

References

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press 2000.

[2] S. Mika, G. Ratsch, J. Weston, B. Scholkopf and K. Muller, "Fisher Discriminant Analysis with Kernels," *Neural Networks for Signal Processing Systems*, pp. 41–48, 1999.

[3] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons 1998.

[4] G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using Kernel Approach," *Neural Computation*, vol. 12, pp. 2385–2404, 2000.

[5] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, no. 1, pp. 25–37, 2000.

[6] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the EM Algorithm," *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[7] G. Ratsch, T. Onoda and K. -R. Muller, "Soft Margins for Adaboost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2000.

[8] A. H. Roger and R. J. Charles, *Topics in Matrix Analysis*, Cambridge University Press 1991.

[9] G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons 1992.

[10] J. W. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "Boosting linear discriminant analysis for face recognition," *In Proceedings of the IEEE International Conference on Image Processing*, pp. 657–660, 2003.

[11] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, pp. 2385-2404, 2000.

[12] T. Xiong, J.P. Ye, Q. Li, V. Cherkassky, and R. Janardan, "Efficient kernel discriminant analysis via QR decomposition," *In Advances in Neural Information Processing Systems 17*, 2005.

[13] C.H. Park and H. Park, "Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition," *SIAM Journal on Matrix Analysis and Application*, to appear (http://www-users.cs.umn.edu/ hpark/pub.html).

[14] M.H. Yang, "Kernel eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods," *In Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 215-220, May 2002.

[15] J.W. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Transactions on Neural Networks*, vol. 12, pp. 117-126, 2003.

[16] G. Dai and Y.T. Qian, "Modified kernel-based nonlinear feature extraction," *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 17-21, 2004.

[17] G. Dai and Y.T. Qian, "Kernel generalized nonlinear discriminant analysis algorithm for pattern recognition," *In Proceedings of the IEEE International Conference on Image Processing*, pp. 2697-2700, 2004.

[18] J. Yang, A.F. Frangi, J.Y. Yang, D. Zhang, and Z. Jin, "KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 230-244, 2005.

[19] G. Dai, D.Y. Yeung, H. Chang, "Extending kernel Fisher discriminant analysis with the weighted pairwise Chernoff criterion," *Proceedings of the Ninth European Conference on Computer Vision (ECCV)*, pp. 308–320, Graz, Austria, May 2006.

[20] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711-720, 1997.

[21] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067-2070, 2001.

[22] A.K. Qin, P.N. Suganthan, and M. Loog, "Uncorrelated heteroscedastic LDA based on the weighted pairwise Chernoff criterion," *Pattern Recognition*, vol. 38, no. 4, pp. 613-616, 2005.

CONCLUSION

Dans cette thèse, nous nous sommes intéressés aux problèmes de la recherche d'images par le contenu, l'extraction des caractéristiques, l'analyse multirésolution et la classification linéaire et nonlinéaire. Nos réalisations et contributions peuvent être résumées comme suit.

Dans le premier chapitre, nous avons proposé une méthode simple et rapide de recherche d'images par le contenu. Pour représenter les images couleurs, nous avons introduit de nouveaux descripteurs de caractéristiques qui sont des histogrammes pondérés par le gradient multispectral. Afin de mesurer le degré de similarité entre deux images d'une façon rapide et efficace, nous avons utilisé une pseudo-métrique pondérée qui se sert de la décomposition en ondelettes Daubechies-8 et de la compression des histogrammes extraits des images. Les poids de la pseudo-métrique ont été ajustés par le modèle classique de régression logistique pour améliorer sa capacité de discrimination et la précision de la recherche. Ce travail a été présenté dans la Conférence Internationale en Recherche Opérationnelle (CIRO'05), Marrakech, Maroc, 2005, et a été publié dans le numéro de mars 2006 du journal international **International Journal of Wavelets, Multiresolution and Information Processing (IJWMIP)**.

Dans le deuxième chapitre, nous avons proposé un nouveau modèle bayésien de régression logistique basé sur une méthode variationnelle. Une comparaison de ce nouveau modèle au modèle classique de régression logistique a été effectuée dans le cadre de la recherche d'images. Nous avons illustré que le modèle bayésien permet une meilleure amélioration de la capacité à discriminer de la pseudo-métrique et de la précision de recherche que le modèle classique. Ce travail a été présenté dans la Conférence internationale **International Conference on Computer Vision and Graphics (ICCVG'06)**, Varsovie, Pologne, 2006, et sera publié dans le journal international **Machine Graphics and Vision (MGV)**.

Dans le troisième chapitre, nous avons détaillé la dérivation du nouveau modèle bayésien de régression logistique basé sur une méthode variationnelle introduite au chapitre 2 et nous avons effectué une comparison exhaustive de ce modèle au modèle classique de régression logistique dans le cadre de la recherche d'images et dans un cadre général. Plus spécifiquement, dans ce cadre général, nous avons comparé le modèle Bayésien à d'autres classificateurs linéaires apparaissant dans la littérature. Ensuite, nous avons comparé notre méthode de recherche utilisant le modèle Bayésien de régression logistique à d'autres méthodes de recherches déjà publiées. Les expérimentations et comparaisons ont été effectuées sur les bases de données d'images couleurs connues WANG, ZuBud, UW et CalTech et sur plusieurs ensembles de données réelles et synthétiques. Ce travail sera publié dans le journal international **IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)**.

Dans le quatrième chapitre, nous avons introduit la sélection des caractéristiques pour améliorer la méthode de recherche présentée dans les chapitres précédents. La séléction des caractéristiques a été effectuée en utilisant séparément les modèles Bayésien et classique de régression logistique. Elle permet de donner automatiquement plus d'importance aux caractéristiques qui discriminent le plus et moins d'importance aux caractéristiques qui discriminent le moins. Une comparison des deux modèles a été effectuée dans le cadre de la recherche d'images basée sur la sélection des caractéristiques. Les expérimentations ont été effectuées sur les bases de données d'images couleurs connues WANG et ZuBud. Ce travail a été publié dans les actes de **Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)**, Hyderabad, Inde, 2007.

Dans le cinquième chapitre, nous avons proposé un nouveau modèle Bayésien d'analyse discriminante logistique basé sur l'usage de noyaux, qui permet une classification nonlinéaire flexible. Ce nouveau modèle a été comparé à l'analyse discriminante de Fisher basée sur des noyaux et à d'autres classificateurs nonlinéaires déjà publiés. Nous comptons enrichir ce travail avec une application de recherche d'images où de détection d'objets (peau, feux, ombre, etc) et le soumettre au journal international **IEEE Transactions on Knowledge and Data Engineering (TKDE)**.