JMLR: Workshop and Conference Proceedings 13: 111-125
2nd Asian Conference on Machine Learning (ACML2010), Tokyo, Japan, Nov. 8–10, 2010.

Content-based Image Retrieval with Multinomial Relevance Feedback

Dorota Głowacka D.GLOWACKA@CS.UCL.AC.UK
Department of Computer Science
University College London

John Shawe-Taylor JST@CS.UCL.AC.UK
Department of Computer Science
University College London

Editor: Masashi Sugiyama and Qiang Yang

Abstract

The paper considers an interactive search paradigm in which at each round a user is presented with a set of k images and is required to select one that is closest to her target. Performance is measured by the number of rounds needed to identify a specific target image or to find an image among the t nearest neighbours to the target in the database. Building on earlier work we assume a multinomial user model with the probabilities of response proportional to a function of the distance to the target. The conjugate prior Dirichlet distribution is used to model the problem motivating an algorithm that trades exploration and exploitation in presenting the images in each round. Experimental results verify the fit of the model with the problem as well as show that the new approach compares favourably with previous work.

Keywords: Image retrieval, Dirichlet distribution, Reinforcement learning

1. Introduction

We consider content-based image retrieval in the case when the user is unable to specify the required content through tags or other explicit properties of the images. In this type of scenario the system must extract information from the user through limited feedback. We consider a protocol that operates through a sequence of rounds in each of which a set of k images is displayed and the user must indicate which is closest to their target. Note that we do not always assume that the target is in the database but rather that there is a hypothetical target image in the user's mind and that the user's likelihood of choosing an image is proportional to an exponentially decaying function of the distance between the displayed images and this target. Performance is assessed by the number of rounds needed before the target image is presented to the user or an image is presented that is among the t nearest neighbours of the target in the database, where $t \geq 1$ is a parameter of the problem.

While this problem has been studied before, we present a novel approach that makes use of the Dirichlet distribution as the conjugate prior to the multinomial distribution in order to model the system's knowledge about the expected responses to the images. This enables us to assess the performance of the system along various dimensions. For example the model predicts that the number of rounds required should be a function of the ratio of the number of images in the database and the number t of nearest neighbours that are accepted as proxy's of the target. Our experiments confirm this prediction indicating that the algorithm will perform well independently of the size of the database of images.

Experiments also confirm the predictions of the model and the degree of fit between model and practical protocol. Finally, performance is compared against earlier solutions with favourable results.

The next section will review previous work on this topic with the problem definition and description of our approach given in Section 3. Section 4 describes the aims of our experiments and gives the results obtained.

2. Previous Work

One of the main research problems in content-based image retrieval with relevance feedback is finding a suitable image in as few iterations as possible. In previous research (Zhang and Chen (2002); Tong and Chang (2001); Gosselin et al. (2008); Chang et al. (2005)), active learning was used to select images around the decision boundary for user feedback to speed up the search process. However, the user might find it difficult to label images lying close to the decision boundary, which results in noise being present in the user feedback. This issue is not considered in most previous research. Recent work by Auer and Leung (2009) explicitly models noisy user feedback. In this model, the algorithm selects images for presentation to the user in such way that after obtaining the user feedback, the algorithm can efficiently search for suitable images by eliminating the ones that do not match the user's query.

Traditionally, in content-based image retrieval with user feedback, it is assumed that the images in the dataset are not labelled (Chen et al. (2001); Rui and Huang (2000); Rocchio (1971); Tong and Chang (2001)). Metric functions measuring similarity based on low-level visual features can be obtained by discriminative methods. Long-term learning is used with training datasets from the feedback of different users (He et al. (2002); Fournier and Cord (2002); Tao et al. (2006); Koskela and Laaksonen (2003); Tao et al. (2007); Linenthal and Qi (2008); Wacht et al. (2006); Tao and Tang (2004)). However, because of different perceptions about the same object, different users may give different kinds of feedback for the same query target. Short-term learning using feedback from a single user in a single search session can be used to deal with different perceptions of objects.

Recently, a large amount of work (Veltkamp and Tanase (1999); Smeulders et al. (2000); Lew et al. (2006); Crucianu et al. (2004); Datta et al. (2008)) explored the use of user feedback as training data. Feedback is used to label data points as positive or negative for the training purposes. The problem with this approach is that at each iteration, the user selects the most relevant image, which may not necessarily be very similar to the ideal target image. Images predicted to be positive examples by discriminative methods are usually selected for presentation in each round. This might hinder progress in the search significantly as parts of the search space with images incorrectly predicted as negative are ignored.

3. Problem Definition and Approach

In this section, we describe the main components of our algorithm.

3.1 Comparative Feedback

Following Auer and Leung (2009), we consider a model in which the search engine supports the user in finding an image that matches her query sufficiently well. In each iteration, the search engine presents a set of images to the user and the user selects the most relevant image from this set. In each iteration, k images from the database $\mathscr{D}$ are presented to the user. The protocol is described below.
For each iteration $i = 1, 2, \dots$ of the search:

- The search engine calculates a set of images $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,k} \in \mathscr{D}$ and presents them to the user.
- If one of the presented images matches the user's query, then the search terminates.
- Otherwise the user chooses one of the images $\mathbf{x}_i^*$ as most relevant according to a distribution $D\{\mathbf{x}_i^* = \mathbf{x}_{i,j} \mid \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,k}; \mathbf{t}\}$, where $\mathbf{t}$ denotes the ideal target image for the user's query.

3.2 The User Model

We assume that the choice of one of the presented images is a random process, where more relevant images are more likely to be chosen. This models some source of noise in the user's selection process. Following Auer and Leung (2009), we assume a similarity measure $S(\mathbf{x}_1, \mathbf{x}_2)$ between images $\mathbf{x}_1, \mathbf{x}_2$, which also measures the relevance of an image $\mathbf{x}$ compared to an ideal target image $\mathbf{t}$ by $S(\mathbf{x}, \mathbf{t})$. Let $0 \leq \lambda \leq 1$ be the uniform noise in the user's choice. The probability of choosing image $\mathbf{x}_{i,j}$ is given by:

$$D\{\mathbf{x}_i^* = \mathbf{x}_{i,j} \mid \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,k}; \mathbf{t}\} = (1 - \lambda) \frac{S(\mathbf{x}_{i,j}, \mathbf{t})}{\sum_{j=1}^{k} S(\mathbf{x}_{i,j}, \mathbf{t})} + \frac{\lambda}{k} \tag{1}$$

Assuming a distance function $d(\cdot, \cdot)$, a possible choice for the similarity measure $S(\cdot, \cdot)$ is:

$$S(\mathbf{x}, \mathbf{t}) = \exp\{-a d(\mathbf{x}, \mathbf{t})^2\} \tag{2}$$

with parameter $a > 0$. Thus, the user's response depends on the squared distance of the selected images from the ideal target image. As a consequence, the accuracy of the user's response will deteriorate if the presented images are far from the ideal target image. In all the experiments, we use squared Euclidean norm

$$d(\mathbf{x}, \mathbf{t}) = \parallel \mathbf{x} - \mathbf{t} \parallel \tag{3}$$

as the distance measure between image $\mathbf{x}$ and the target image $\mathbf{t}$.

3.3 The Algorithm

We describe an exploration strategy which is an attempt to solve the search problem. The algorithm maintains weights $m(\mathbf{x})$ on the images $\mathbf{x}$ in the database $\mathscr{D}$ and calculates the images to be presented to the user according to these weights. The algorithm is motivated by the Dirichlet Process (DP). We call it the Dirichlet Sampling (DS) search algorithm.

The algorithm uses a DP to represent its estimates of the likelihood that different images are relevant. As the DP is conjugate to the multinomial, updating based on the user feedback is straightforward (see Sections 3.3.2 and 3.3.4 below). Furthermore, the DP can be used to generate a set of candidate images to present to the user at each iteration through taking a set of random samples (see Section 3.3.5). There is a slight complication here in that the samples of a DP are distributions over the images with the actual images chosen for presentation being the most probable in the sample distributions.

3.3.1 The Dirichlet Process

Let us begin with the definition of Dirichlet distribution (Sjolander et al. (1996); Teh; Ferguson (1973); Blackwell and MacQueen (1973); Neal (1992); Rasmussen (2000)). The Dirichlet distribution is a multi-parameter generalisation of the Beta distribution and defines a distribution over distributions. Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ be a multinomial probability distribution on the discrete space $\mathscr{X} = \{\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_n\}$ with $\mathbf{x}$ a random variable in the space $\mathscr{X}$. The Dirichlet distribution on Θ is given by the following formula:

$$P(\Theta \mid \alpha, M) = \frac{\Gamma(\alpha)}{\prod_{i=1}^{n} \Gamma(\alpha m_i)} \prod_{i=1}^{n} \theta_i^{\alpha m_i - 1} \tag{4}$$

where $M = \{m_1, m_2, \ldots, m_n\}$ is the base measure defined on $\mathscr{X}$ and is the mean value of Θ, and α is a precision parameter that specifies how concentrated the distribution is around M. α can be regarded as the number of (pseudo-) measurements observed to obtain M. The greater the value of α, the more the distribution is concentrated around M. We refer to this distribution as $Dir(\alpha m_1, \ldots, \alpha m_n) = Dir(\alpha M)$

Consider a possibly continuous input space $\mathscr{X}$. A Dirichlet Process (DP) on $\mathscr{X}$ is a distribution over distributions on $\mathscr{X}$ with samples being measures on $\mathscr{X}$. For a random distribution G to be distributed according to a DP, its marginal distributions on partitions of the input space have to be Dirichlet distributed. G is Dirichlet Process distributed with base measure M and precision parameter α, written $G \sim DP(\alpha, M)$ if

$$(G(\mathscr{X}_1), \ldots, G(\mathscr{X}_n)) \sim Dir(\alpha M(\mathscr{X}_1), \ldots, \alpha M(\mathscr{X}_n)) \tag{5}$$

for every measurable partition $\mathscr{X}_1, \ldots, \mathscr{X}_n$ of $\mathscr{X}$.

3.3.2 The Posterior Distribution

We are interested in using the Dirichlet distribution to learn a distribution from observations based on a multinomial noise model. If the target distribution on $\mathscr{X}_1, \ldots, \mathscr{X}_n$ is $\mu = (\mu_1, \ldots, \mu_n)$, we will observe $\mathscr{X}_i$ with probability μ_i. Let $z_1, \ldots, z_l$ s.t. $z_i \in \{\mathscr{X}_1, \ldots, \mathscr{X}_n\}$ be a sequence of independent draws from μ. We are interested in the posterior distribution G given the observations of $z_1, \ldots, z_l$. Let $\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_n$ be a finite measurable partition of $\mathscr{X}$, and let n_k be the number of observed values in $\mathscr{X}_k$, that is $n_k = \mid \{i : z_i = \mathscr{X}_k, i = 1, \ldots, l\} \mid$. Then,

$$(G(\mathscr{X}_1), \ldots, G(\mathscr{X}_n)) \sim Dir(\alpha M(\mathscr{X}_1) + n_1, \ldots, \alpha M(\mathscr{X}_n) + n_n) \tag{6}$$

The posterior DP has updated precision parameter $\alpha^* = \alpha + l$ and the base measure $M^* = \frac{\alpha M + \sum_{i=1}^{n} n_i l_i}{\alpha + l}$, where l_i is a unit vector with entry 1 in the i^{th} component.

The posterior base measure distribution is a weighted average between the prior base measures M and the empirical distribution $\frac{\sum_{i=1}^{n} n_i l_i}{n}$. The weight associated with the prior base distribution is proportional to α, while the empirical distribution has weight proportional to the number of observations n. Thus, as the number of observations grows larger and larger, $n \gg \alpha$, the posterior is dominated by the empirical distribution.

3.3.3 Application to the Image Selection Problem

Let $\mathscr{D}$ be a dataset of n images $(\mathbf{x}_i)_{i=1,\ldots,n}$. Let $M = m_1, m_2, \ldots, m_n$ be the base measure defined on $\mathscr{D}$. Initially, we set $m_i = \frac{1}{n}$ for $i = 1, \ldots, n$.

Let $\mathbf{x}_i^* \in \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,k}\}$ be the image chosen by the user at iteration i from among the k presented images $\{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,k}\}$. We suppress the index i to simplify the exposition. We need to define the user model that defines how the image $\mathbf{x}_i^*$ is chosen. The simplest possibility is that we view the set of images $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ as partitioning the complete space of images into sets $\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_k$ with $\mathscr{X}_j = \{\mathbf{x} : d(\mathbf{x}_j, \mathbf{x}) < d(\mathbf{x}_{j'}, \mathbf{x}), j' \neq j\}$. In the event of ties a random assignment to one of the minimum distance partitions is made. Now, if we take the "true" response probability as

$$m_i^* \propto \exp(-ad(\mathbf{x}_i, \mathbf{t})^2) \tag{7}$$

where $\mathbf{t}$ is the target image, then the user model should choose partition $\mathscr{X}_j$ with probability

$$P(\mathscr{X}_j) = \sum_{i:\mathbf{x}_i \in \mathscr{X}_j} m_i^* \tag{8}$$

When we consider the user's choice, we need to update the base measures and the precision parameter. The standard posterior update tells us how to do this for the partition probabilities, but not for the single images. Since we are not able to distinguish between the images in a partition, the natural choice is to update

```
**if** $x_i \in \mathscr{X}_j$ **then**
    $m_i \leftarrow \frac{\alpha m_i + |\mathscr{X}_j|^{-1}}{\alpha + 1}$
**else**
    $m_i \leftarrow \frac{\alpha m_i}{\alpha + 1}$
**end if**
$\alpha \leftarrow \alpha + 1$
```

Note that at each iteration the partition of the images will change, but that the update will correctly compute the posterior measure over the complete set of images.

3.3.4 Alternative Noise Model

There is a slight mismatch between the noise model proposed by Auer and Leung (2009) and the Dirichlet prior we have adopted. This is because the probability of response is computed from the distance of the images that define the partitions of the images rather than the probability of the partitions in a target distribution td, that is

$$P(\mathscr{X}_j) \propto \sum_{i:\mathbf{x}_i \in \mathscr{X}_j} \mathrm{td}_i. \tag{9}$$

We will perform experiments with this alternative Dirichlet noise model where we take the target distribution to be defined as

$$\mathrm{td}_i \propto \exp\{-ad(\mathbf{x}_i, \mathbf{t})^2\}. \tag{10}$$

This will make it possible to measure convergence of the algorithm by computing the probability of the target distribution in the posterior distribution as a function of the number of iterations. By comparing this convergence for the two noise models (the more realistic noise model and the Dirichlet noise model) we will be able to assess the cost of this mismatch between model and application.

3.3.5 Balancing Exploration versus Exploitation

The final ingredient in the overall search algorithm is the way in which the images to be presented to the user should be chosen at each iteration. This involves a trade-off between presenting images that appear promising based on best current estimates of the mean given by the posterior measure $(m_i)_{i=1,\dots,n}$ (exploitation) and trying areas where our current estimate could be too pessimistic (exploration). The strategy we adopt to solve this problem is to draw k samples from the posterior distribution and select the image $\mathbf{x}_j$ that has the highest probability in the jth sample, $j = 1, \dots, k$. Note that we should not use the individual m_j of the images when drawing these samples since the image selected will be a proxy for the approximately n/k images in its partition and it is the partition that we wish to choose. This problem is overcome by multiplying each of the αm_i by n/k before drawing each of the samples. We therefore summarise the selection algorithm as

```
for j = $1, \dots, k$ do
    r ← randg($m * \alpha * n/k$)
    [val, i] ← max(r)
    ind[j] ← i
end for
Output: Array ind gives indices of selected images
```

One way of viewing this strategy is to select images to display with probability proportional to the probability that the partition they define contains the target. This algorithm randomly selects images from the dataset according to their weights m. Images with higher weights are more likely to be relevant and thus more likely to be presented to the user.

4. Experiments

In this section, we test the performance of our algorithm.

4.1 Questions to Be Explored

The aim of the experiments is to investigate the following issues:

1. Scaling of the algorithm with size of image set with ratio of target set.
2. Scaling of the algorithm with target size for fixed image set size.
3. Convergence of probability of target in posterior distribution.
4. Comparison with previous work.

We begin with a subsection describing the data that will be used in all of the experiments and is the same as that used by Auer and Leung (2009). The next subsection describes a series of experiments that we have performed with the Dirichlet search (DS) algorithm proposed in this paper. The third subsection describes experiments with Auer and Leung (2009)'s algorithm referred to as the AL algorithm in the rest of this section. These include scaling and convergence properties. The final subsection highlights comparisons between the two algorithms.

4.2 Setup of Experiments

For the experiments, we used the VOC2007 dataset (Everingham et al. (2007)) with 23 categories and 9963 images. The dataset was built for the PASCAL Visual Object Classes Challenge 2007. The goal of the challenge was to recognise objects from several classes in realistic scenes. The 23 classes (and subclasses) are: person (person, foot, hand, head), animal (bird, cat, cow, dog, horse, sheep), vehicle (aeroplane, bicycle, boat, bus, car, motorbike, train) and indoor (bottle, chair, dining table, potted plant, sofa, tv monitor). Each of the 9963 images in the dataset is annotated by a bounding box and class label for each object from the 23 classes present in the image. Multiple objects from multiple classes may be present in an image.

In our experiments, we used the 23 high-level features to describe the images (Auer and Leung (2009)). For an image the feature value for an object class is the size (as calculated from the bounding box) of the largest object from this class in the image. If no object from a particular class is present in the image, then the feature value is 0. For an easier analysis, we set $k = 2$ so that only 2 images are presented to the user in each iteration. The number of search iterations is expected to be significantly reduced for larger k. All reported results are averaged over 1000 searches for randomly selected target images from the dataset.

4.3 DS Algorithm - Experimental Results

In the first set of experiments, we use the user model proposed in Auer and Leung (2009) (described in more detail in Section 3.2). In all the experiments described in this section, the α parameter in the DS algorithm was set to 100. (We also tested the impact of the α parameter in the DS algorithm on the search results, however, its influence was insignificant.) The a and λ parameters in the user model were set to 8 and 0.1, respectively. Initially, following Auer and Leung (2009), we set the search to terminate when the target image is presented to the user. In this scenario, the average number of iterations was 264. In real-life scenario, however, the user might terminate the search as soon as he is presented with an image which is very close to his ideal target image. Hence, in the next set of experiments, we calculated the Euclidean distance of each image from the target image. Each iteration of the algorithm terminates when the user is presented with the target image or: (1) one of the 4 images closest to the target, (2) one of the 9 images closest to the target. The average number of iterations was significantly reduced to 110 and 84, respectively. The results of these experiments are summarised in Table 1 below. Table 1 shows that for a fixed size image dataset, the average number of iterations gets smaller as the size of the image target set gets bigger.

Target size	Average number of iterations
1	264
5	110
10	84

Table 1: Scaling of the DS algorithm with target size for a fixed image set size, with the noise model proposed by Auer and Leung (2009)

In the next set of experiments, we used the alternative Dirichlet user model described in Section 3.3. The a parameter in the noise model was set to 0.5 in all the experiments described below. As previously, we tested the scaling properties of the algorithm by increasing the size of the image target set, while keeping the size of the image database constant. The results of the experiments are summarised in Table 2 below.

Target size	Average number of iterations
1	330
5	99
10	60

Table 2: Scaling of the DS algorithm with target size for a fixed image set size, with the Dirichlet noise model

As Table 2 shows, the DS algorithm retains its scaling properties also with the alternative Dirichlet noise model - for a fixed size image dataset, the average number of iterations gets smaller as the size of the image target set grows. However, the DS algorithm performs differently depending on the noise model used. For searches that terminate when the user is presented with the ideal target image, the algorithm performs significantly worse with the Dirichlet noise model. When the size of the target set is increased, however, the algorithm performs better with the Dirichlet noise model, which indicates that the Dirichlet noise model allows the algorithm to find the region with the target faster than the exponential noise model.

In order to test the compatibility of the DS algorithm with the proposed multinomial user model, we calculated the posterior probability of the target in posterior distribution after each iteration of the search algorithm. The target set consisted of 10 images. The experiment indicates that the Dirichlet noise model is indeed an appropriate noise model for the DS algorithm. As one might expect, the probability of the target increases with each iteration, which is illustrated in Figure 1.

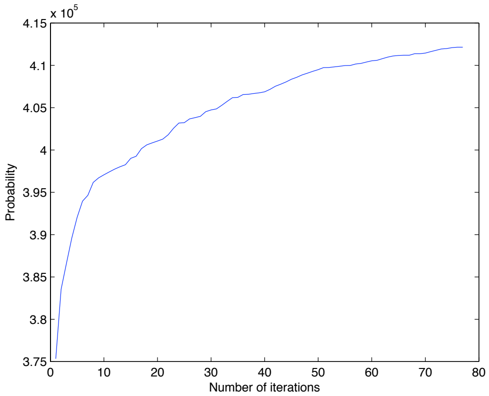


Figure 1: Convergence of probability of target in posterior distribution with the user Dirichlet model.

4.4 AL Algorithm - Description and Experimental Results

In this section, we describe the search algorithm proposed by Auer and Leung (2009). So far this has been the best performing algorithm that incorporates both noisy user feedback and aspects of reinforcement learning. The algorithm proposes the following weighting scheme that demotes all apparently less relevant images by a constant discount factor $0 \leq \beta < 1$. Let $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$ be the weights of the images in the dataset. Initially, all $w_i = 1$. Let $\mathbf{x}_i^* \in \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,k}\}$ be the image chosen by the user as the most relevant at iteration i. If the search has not terminated, then all the images $\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,k}$ are not sufficiently relevant and thus their weights are set to 0. All the images closer to some $\mathbf{x}_{i,j}$ rather than to $\mathbf{x}_i^*$ are demoted by the discount factor β. Formally, Auer and Leung (2009) use the following update of the weights:

1. Initialise all $w_j = 1$
2. For each iteration $i = 1, 2, \ldots$ of the search:
   - For all $\mathbf{x}_j$ set:

$$w_{i+1} = \begin{cases} w_i & \text{if} \quad d(\mathbf{x}_i^*, \mathbf{x}) = \min_j d(\mathbf{x}_{i,j}, \mathbf{x}) \\ \beta * w_i & \text{otherwise} \end{cases}$$

   - Set $w_{i+1} = 0$ for all $j = 1, \ldots, k$

The images presented to the user are selected by the random sampling algorithm, which randomly selects (without repetition) images from the dataset according to their weights.

As in the experiments involving the DS algorithm, the values of a and λ in the user model were kept constant at 8 and 0.1, respectively. Initially, we tested the influence of the β parameter on the performance of the AL algorithm. The value of β varied from 0.1 to 0.9. The search terminated only when the user was presented with the ideal target image. As Figure 2 shows, the average number of iterations to complete the search ranges from 204 to 354, depending on the value of β. The algorithm performs best when $\beta = 0.6$.

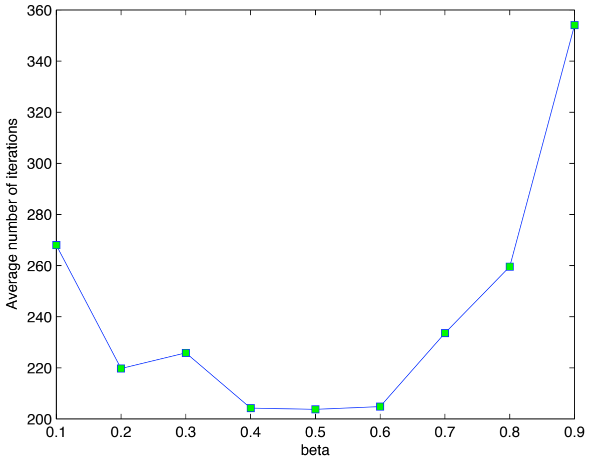
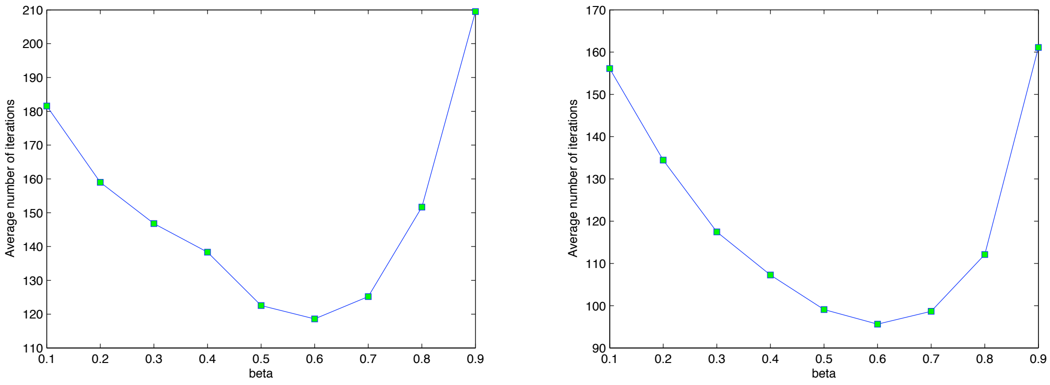


Figure 2: Average number of search iterations of the AL with varying β

The experiments also indicate that when the value of β ranges from 0.1 to 0.4, the use of the AL algorithm often leads to the weights all the images being equally small. In such circumstances, finding the correct target image is extremely difficult. Consequently, in about 2 % of searches when β ranges from 0.1 to 0.4, the user is effectively forced to conduct an exhaustive search of the entire database in order to find the target image.

We also tested the scalability of the algorithm with regards to the size of the target set, while keeping the size of the image dataset constant. The results of these experiments are presented in Figure 3. When the target set consisted of 5 points, the average number of iterations ranged from 118 to 209, depending on the value of β. However, when the target set consisted of 10 points, the average number of iterations ranged from 95 to 161, depending on the value of β. The algorithm performed best when $\beta = 0.6$. In spite of increasing the size of the target image set, when $0.1 \leq \beta \leq 0.4$ about 2% of queries required an exhaustive search of the entire image dataset in order for the search to terminate.

In the weighting scheme proposed in Auer and Leung (2009) only the weights of the less relevant images are exponentially discounted. In the DS algorithm, we reduce the weights of the less relevant images and at the same time we increase the weights of the images that are likely to be more relevant. Additionally, in the AL algorithm, weights of the images presented to the user are set to 0. Effectively, at each iteration of the search algorithm we are reducing the set of images from which the images that are to be presented to the user are selected. We consider this to be a strategy that will only have an effect for relatively small databases and/or small values of t, the neighbourhood size of the target that is accepted.



(a) (b)

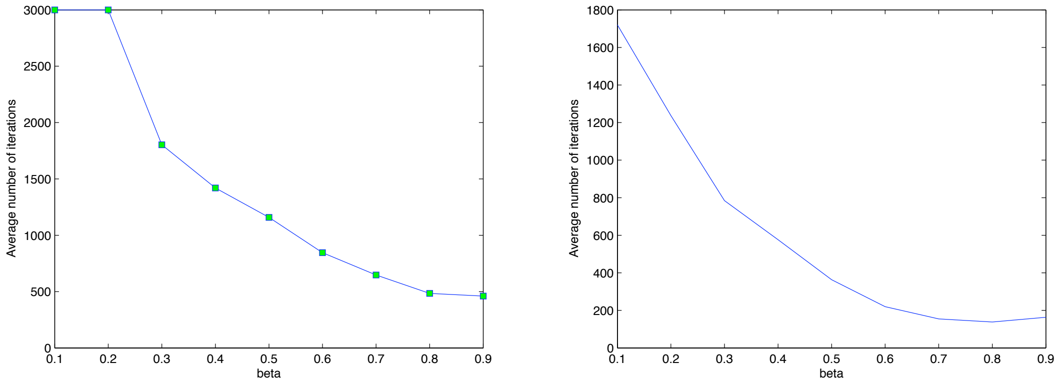
Figure 3: Average number of search iterations of the AL algorithm with varying β. The search terminates when the user is presented with the target image or (a) one of the 4 images closest to the target (b) one of the 9 images closest to the target image.

The reason for this is best illustrated by considering a fixed ratio of the size n of the database and t: $n/t = r$. Now if we scale up n keeping r fixed we would expect the number of images shown to remain constant and hence the number that had their weights set to zero would be the same. This number would be a decreasingly small fraction of the size n of the database and so would no longer have any influence on the algorithm. Our results confirm that the performance of the AL algorithm is worse if we set the weights to zero and that the effect of this reduces as the database size grows for fixed ratio n/t.

In the next set of experiments, we test the influence of setting the weights to 0 on the performance of the AL algorithm. The weights of the images presented to the user are never set to 0; instead they either remain unchanged or are reduced by the β factor. We set the search algorithm to terminate when the number of iterations reached 3000 or when the user is presented with the ideal image. The results of the experiments are presented in Figure 4.

The algorithm performs dramatically worse in comparison to the situation when the weights of the images seen by the user are set to 0. The lowest average number of iterations is now 460 as compared to the 204 averaged in the previous experiments. For smaller values of β, the algorithm never terminates before reaching 3000 iterations. The average number of iterations as well as the average number of exhaustive searches falls as the value of β grows. However, the number of exhaustive searches is never smaller than 10% of the total number of iterations.

As in previous experiments, the convergence rate improves if the size of the target set is increased from 1 to 10. The average number of iterations varies between 1720 and 139, while the percentage of exhaustive searches ranges from 3 to 16%, depending on the value of β. Thus, the AL algorithm shows the same scaling properties with respect to the target size irrespective of whether the weights seen by the user are set to 0 or not. However, not setting the weights to 0 has a detrimental effect on the performance of the algorithm.



(a)

(b)

Figure 4: Average number of search iterations of the AL algorithm with varying β when the weights are not set to zero. The size of the target set consists of (a) 1 image, (b) 10 images.

4.5 Comparison of the DS Algorithm and the AL Algorithm

In this section we compare the performance of the two algorithms. In general, the DS algorithm terminates the search faster than the AL algorithm. The AL algorithm outperforms the DS algorithm only in one experimental setup - when the weights in the AL algorithm are set to 0 once presented to the user while the target set consists of only one image (once the set of the target images increases the performance of the algorithm deteriorates). However, as mentioned earlier, one might expect the user to terminate the search once he is shown an image that is close enough to his ideal target. Thus, we might view the cases where the target set is larger than 1 to be more representative of a real-life image search scenario. Further, the performance of the AL algorithm deteriorates dramatically if the weights of the images are never set to 0. As discussed earlier in Section 4.4, this renders the AL algorithm not particularly well suited for large datasets as even setting the weights to 0 in such a scenario will not speed up the search.

In the next set of experiments, we compared the scaling properties of the two algorithms, i.e. we calculated the average number of iterations as the size of the image dataset increases along with the size of the target set. We started with a dataset consisting of 1000 images selected randomly from the original dataset of 9963 images and one image as the target. Next, we performed the search on the smaller dataset with the two algorithms. We gradually increased the dataset by 1000 images, while increasing the size of the target set by one image. As Fig. 5 shows, the performance of the DS algorithm is not affected by the size of the dataset. The average number of iterations oscillates between 50 and 60, irrespective of the size of the image dataset. In the case of the AL algorithm, the average number of iterations rises as the size of the dataset grows.

In all the experiments reported so far, the user is always presented with only two images at each iterations, which requires a rather high number of iterations to complete the search. However, one might expect a significant improvement in performance once the number

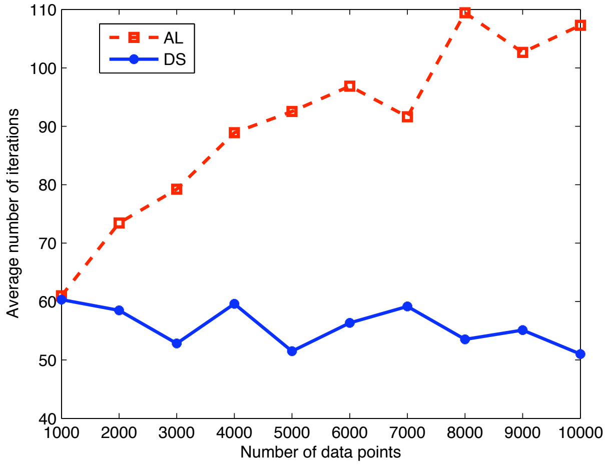


Figure 5: Comparison of the scaling properties of the AL algorithm and the DS algorithm.

of images presented to the user at each iteration is increased. Thus, in the last set of experiments, we compare the performance of the two algorithms as the value of k increases. In all the experiments reported below, the value of β in the AL algorithm was set to 0.6. The results of the experiments are presented in Table 3. We report the average number of iterations required to terminate the search by the two algorithms with respect to the size of the target set and the value of k.

	k	=2	k	= 5	k	=10
Target	AL	DS	AL	DS	AL	DS
1	845	330	431	123	228	71
5	448	99	92	46	43	24
10	219	60	51	28	22	16

Table 3: Comparison of the performance the AL and DS algorithm as the value of k increases.

As Table 3 shows, the number of iterations required to complete the search decreases as the number of images presented to the user increases. Table 3 also show that the DS algorithm proposed in this paper significantly outperforms the AL algorithm in all the experiments.

5. Conclusions

We have presented a new approach to content-based image retrieval based on multinomial relevance feedback. We model the knowledge of the system using a Dirichlet process. The model suggests an algorithm for generating images for presentation that trades exploration and exploitation. The model also enables us to make predictions about the scaling of the algorithm and convergence properties. We have verified that these predictions are borne out in our experiments. The experiments have also explored the degree of misfit between the model and the real retrieval protocol. These experiments show that there is still a good fit between model and measured performance. Furthermore the experiments confirm that the new approach outperforms earlier work using a more heuristic strategy.

References

P. Auer and A.P. Leung. Relevance feedback models for content-based image retrieval. In *Multimedia Analysis, Processing and Communications*. Springer, 2009.

D. Blackwell and J.B. MacQueen. Ferguson distributions via polya urn schemes. *Annals of Statistics*, 1:353 – 355, 1973.

E. Chang, S. Tong, K. Goh, and C. Chang. Support vector machine concept-dependent active learning for image retrieval. *IEEE Transactions on Multimedia*, 2005.

Y. Chen, X.S. Zhou, and T.S. Huang. One-class svm for learning in image retrieval. In *Proceedings of ICIP*, pages 34 – 37, 2001.

M. Crucianu, M. Ferecatu, and N. Boujemaa. Relevance feedback for image retrieval: a short survey. In *State of the art in audiovisual content-based retrieval, information universal access and interaction, including data models and languages*. European Network of Excellence (FP6), 2004.

R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 2008.

M. Everingham, L. van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 results. `http://www.pascal-network.org/challenges/VOC/voc2007/workshop`, 2007.

T.S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209 – 230, 1973.

J. Fournier and M. Cord. Long-term similarity learning in content-based image retrieval. In *Proceedings of ICIP*, pages 441– 444, 2002.

P.H. Gosselin, M. Cord, and S. Philipp-Foliguet. Active learning methods for interactive image retrieval. *IEEE Transactions on Image Processing*, 2008.

X. He, O. King, W.Ma, M. Li, and H. Zhang. Learning a semantic space from user's relevance feedback for image retrieval. *IEEE Trans. Circuits Syst. Video Techn.*, pages 39 – 48, 2002.

M. Koskela and J. Laaksonen. Using long-term learning to improve efficiency of content-based image retrieval. In *Proceedings of PRIS*, pages 72–79, 2003.

M.S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: state of the art and challenges. In *TOMCCAP*, pages 1 – 19, 2006.

J. Linenthal and X. Qi. An effective noise-resilient long-term semantic learning approach to content-based image retrieval. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'08)*, 2008.

R.M. Neal. Bayesian mixture modeling. In *Proceedings of the Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis*, pages 197 – 211, 1992.

C.E. Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems (12)*, 2000.

J. Rocchio. Relevance feedback in information retrieval. In J. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313 – 323. Prentice–Hall, 1971.

Y. Rui and T.S. Huang. Optimizing learning in image retrieval. In *Proceedings of CVPR*, pages 1236 –1236, 2000.

K. Sjolander, K. Karplus, M. Brown, R.Hughey, A. Krogh, I.S. Mian, and Haussler D. Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Computational Applied Bioscience*, 12(4):327 – 45, 1996.

A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1349 – 1380, 2000.

D. Tao and X. Tang. Nonparametric discriminant analysis in relevance feedback for content-based image retrieval. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 1013 – 1016, 2004.

D. Tao, X. Tang, X. Li., and X. Wu. Asymmetric bagging and random subspace for support vector machines –based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088 – 1099, 2006.

D. Tao, X. Li, and S.J. Maybank. Negative samples analysis in relevance feedback. *IEEE Trans. Knowl. Data Eng.*, 19(4):568 – 580, 2007.

Y.W. Teh. Dirichlet processes. `http://www.gatsby.ucl.ac.uk/ ywteh/research/npbayes/dp.pdf`.

S. Tong and E.Y. Chang. Support vector machine active learning for image retrieval. In *Proceedings of ACM Multimedia*, pages 107 – 118, 2001.

R.C. Veltkamp and M. Tanase. Content-based image retrieval systems: a survey. In *State-of-the-Art in Content-Based Image and Video Retrieval*, pages 97 – 124. 1999.

M. Wacht, J. Shan, and X. Qi. A short-term and long-term learning approach for content-based image retrieval. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, pages 389 – 392, 2006.

C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, pages 260 – 268, 2002.