Query by Semantic Example

Nikhil Rasiwasia[1], Nuno Vasconcelos[1], and Pedro J. Moreno[2]

[1] Statistical Visual Computing Lab,
University of California, San Diego
nikux@ucsd.edu, nuno@ece.ucsd.edu
[2] Google, Inc. 1440 Broadway, 21st Floor New York, NY USA
pedro@google.com

Abstract. A solution to the problem of image retrieval based on query-by-semantic-example (QBSE) is presented. QBSE extends the idea of query-by-example to the domain of semantic image representations. A semantic vocabulary is first defined, and a semantic retrieval system is trained to label each image with the posterior probability of appearance of each concept in the vocabulary. The resulting vector is interpreted as the projection of the image onto a semantic probability simplex, where a suitable similarity function is defined. Queries are specified by example images, which are projected onto the probability simplex. The database images whose projections on the simplex are closer to that of the query are declared its closest neighbors. Experimental evaluation indicates that 1) QBSE significantly outperforms the traditional query-by-visual-example paradigm when the concepts in the query image are known to the retrieval system, and 2) has equivalent performance even in the worst case scenario of queries composed by unknown concepts.

1 Introduction

Content-based image retrieval (CBIR), has been the subject of a significant amount of computer vision research in the recent past [6]. Two main retrieval paradigms have evolved over the years: one based on visual queries, here referred to as *query-by-visual-example* (QBVE), and the other based on text, here denoted as *semantic retrieval*. Under the QBVE paradigm, each image is decomposed into a number of low-level visual features (e.g. a color histogram) and image retrieval is formulated as the search for the best database match to the feature vector extracted from a user-provided query image. It is, however, well known that strict visual similarity is, in most cases, weakly correlated with the measures of similarity adopted by humans for image comparison. This motivated the more ambitious goal of designing retrieval systems with support for semantic queries [4]. The basic idea is to annotate images with semantic keywords, enabling users to specify their queries through a natural language description of the visual concepts. Because manual image labeling is a labor intensive process, the goal of semantic retrieval generated significant interest in the problem of the automatic extraction of semantic descriptors, by the application of machine learning algorithms. Early efforts targeted the extraction of specific semantics, more recently there has been an effort to solve the problem in greater generality, through techniques capable of learning relatively large semantic vocabularies from informally annotated training image collections with resort to unsupervised [1, 2, 5] and weakly supervised learning [9].

When compared to QBVE, semantic retrieval has the advantages of 1) image similarity at a higher level of abstraction, and 2) support for the natural language queries. However, the performance of semantic retrieval systems tends to degrade for semantic classes that were not identified as potentially interesting during training, and can lead to less intuitive interaction with retrieval systems (especially during query refinement) than QBVE. In this work, we show that it is possible to combine the advantages of the two formulations by extending the query-by-example paradigm to the semantic domain. We refer to the combination of the two paradigms as query-by-semantic-example (QBSE), and compare its performance to QBVE. Our results indicate that QBSE can perform significantly better for queries composed of concepts known to the semantic retrieval system, and achieves equivalent performance in the worst case scenario of queries composed by concepts outside of the semantic vocabulary.

2 Motivation

2.1 Generalization

In terms of generalization, the performance of QBVE and semantic retrieval systems can be quite distinct. On one hand, natural language queries enable a much higher level of *query abstraction*, and therefore exhibit much better generalization along *the dimension of image similarity*. For example, a query for "sky" will return scenes of both daytime (where sky is mostly blue) and sunsets (where sky tends to be orange) with equal ease. QBVE can be quite limited in this respect, since most concepts of interest exhibit a great diversity of visual appearance. It is usually quite difficult to design a set of visual features that captures all the relevant dimensions of image variability (e.g. that sky can be both blue or orange). On the other hand, semantic retrieval can be quite brittle, due to the need to pre-learn appearance models for all visual concepts of interest [8]. Learning large vocabularies is a difficult task, which requires large corpuses of manually labeled data that are usually not available. In result, it is not uncommon to find scenes for which the most obvious semantic classes are not even defined in the supported semantic vocabulary. For these queries, the performance of semantic retrieval systems can degrade quite dramatically. Other problems include the fact that many scenes do not have a unique interpretation[3] (e.g. a picture of a lake may evoke the "fishing" descriptor for fishing aficionados, the "wind-surfing" label for fans of this sport, and the simple "lake" characterization for most other users), and the fact that it is possible to miss images that use different synonyms in their descriptions (e.g. when faced with a query for "sea", the retrieval system must assign a non-zero relevance to classes

[3] It is commonly said that "a picture is worth a thousand words".

such as “ocean”, “shore”, “waves”, “coast”, or “beach”). None of these problems affect QBVE, which places very few constraints on the supported queries and, therefore, generalizes much better along the *dimension of query diversity*.

2.2 User Interaction

A second metric of retrieval system performance where QBVE and semantic retrieval differ significantly, is that of user-interaction. Natural language queries are the easiest form of *query specification* for most naive users. By definition, QBVE requires an example similar to the desired image, which is typically not easy to find. Furthermore, due to the different measures of similarity implemented by users and retrieval system, there can be a significant difference in the retrieval efficiency achieved by power and naive users. Successful interaction with a QBVE system typically requires some ability, by the user, to “think” in terms of low-level properties such as color or texture. On the other hand, QBVE systems tend to enable a more intuitive *user interaction*. This is particularly true when the desired image is not immediately found, and there is a need for *query refinement*. The refinement of a natural language query is usually not trivial, and can be particularly challenging when the supported semantic vocabulary is small. In QBVE systems, interaction proceeds by 1) visual inspection of the top results to the current query and 2) selection of a number of examples for the subsequent query. This builds on the ability of the human visual system to quickly scan through a screen of images and select those that are most like the image of interest. Furthermore, assuming that the database is large enough, there is never a shortage of subsequent examples with which to refine the query.

2.3 Query by Semantic Example

From the discussion above, it follows that QBVE and semantic retrieval are, in many respects, *complementary*. While semantic retrieval generalizes better along the dimension of image similarity, QBVE supports a much broader query diversity. While the former enables easier query specification, the latter allows more intuitive interaction. In fact, the advantages of each paradigm are not mutually exclusive: *while those of semantic retrieval are indisputably connected to the semantic representation, the limitations of this paradigm are mostly due to the desire for an unambiguous query specification, as a short natural language description*. Let us assume, for an instant, that instead of a few keywords, 1) the user specifies the query as *a vector of weights for all the keywords in the semantic vocabulary* supported by the retrieval system, and 2) each weight represents the *relevance, to the query, of the associated keyword*.

Clearly, because the representation is still of semantic level, none of the advantages of semantic retrieval are compromised. On the other hand, most of its limitations are eliminated. First, synonyms are no longer a problem, since all the semantic classes that could be relevant receive a non-zero weight. Second, even if the semantic class of interest is not part of the semantic vocabulary, there may still be various semantic concepts that are relevant for the query. For example, "fishing" images are likely to be returned in response to a query composed of fishing-related terms, e.g. "lake", "boats", "nets", "water", and "people", even if an appearance model for the fishing class has never been explicitly learned. Finally, it may suffice to specify the weights qualitatively, by *equating the relevance weights with the probability of the concept appearing in the desired image* Given that concepts of small weight will penalize potential false-positives (e.g. a zero weight for "beach" scenes will filter out a large number of possible false-positives for the "fishing" query), it may not be necessary to specify the concept probabilities with great accuracy.

2.4 Implementation

It is obviously not feasible to ask a user to explicitly provide all the probabilities required to make this type of query practical. The user can, nevertheless, provide these probabilities *indirectly*, through the adoption of the query-by-example paradigm. The basic idea is to, as in QBVE systems, let users specify the query in visual terms, by providing query images. These images are then classified by the semantic retrieval system which returns a vector of probabilities, where each component is the posterior probability of a semantic concept satisfying the query. This probability vector is then compared to the set of similar probability vectors previously computed for each of the images in the database, in the standard query-by-example fashion. Note that, because from the user point of view the interaction really occurs at the visual level, this shares all the user-interaction advantages of QBVE. In fact, the combination of query by example with the semantic representation even allows a combination of the interaction modes, e.g. user starts with a traditional natural language query and switches to QBSE for query refinement.

An interesting interpretation QBSE, is that of *query-by-example on a semantic feature space*. The space is the simplex of posterior concept probabilities, and each image is represented as a point in this simplex, as illustrated by Fig. 1 a). Image similarity is measured by evaluating distances in this space. When the user selects a query image, the computation of the posterior probabilities for that image can be seen as a (highly non-linear) *projection* of the image into this semantic space. Each probability can be thought of as a *semantic feature*. Features (semantic concepts) that are not part of the semantic vocabulary define directions that are orthogonal to the semantic space. While it is impossible to recover their values exactly, they can still be approximated by their closest projection in the space. The traditional specification of the query by a short natural language description can also be mapped to the space: it is equivalent to the adoption of a binary probability vector where a few concepts are assigned non-zero posterior probabilities and all other probabilities are set to zero. This restricts the area populated by the images to the sides, and most frequently the corners, of the simplex, as illustrated by Figure 1 b). Under QBSE, images can be projected onto the entire simplex, enabling a much richer representation.

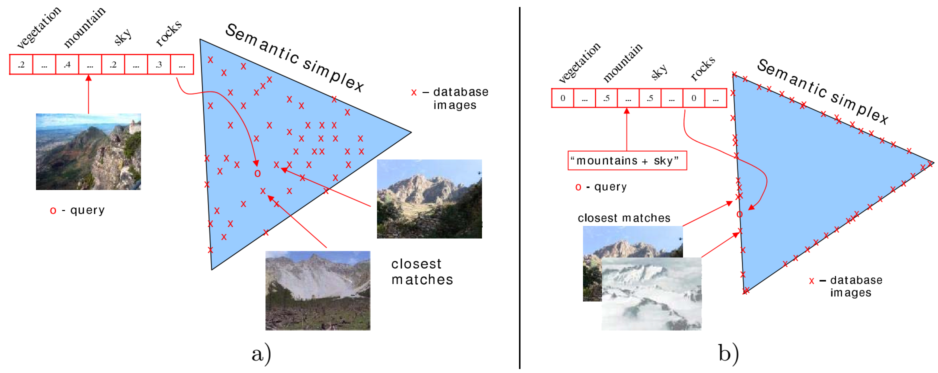

Fig. 1. Semantic image retrieval. a) Under QBSE the user provides a query image, posterior probabilities (given the image) are computed for all concepts, and the image represented by the concept probability distribution. b) Under the traditional semantic retrieval paradigm, the user specifies a short natural language description, and only a small number of concepts are assigned a non-zero posterior probability.

3 Query by Semantic Example

3.1 Definitions

The starting point for the design of a QBSE retrieval system is the combination of an image database $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_D\}$ and a vocabulary $\mathcal{L} = \{w_1, \ldots, w_L\}$ of semantic labels or keywords w_i. All database images are annotated with a caption composed of words from $\mathcal{L}$, i.e the caption $\mathbf{c}_i$ that (in the judgment of a human labeler) best describes image I_i is available for all i. Note that $\mathbf{c}_i$ is a binary L-dimensional vector such that $\mathbf{c}_{i,j} = 1$ if the i^{th} image was annotated with the j^{th} keyword in $\mathcal{L}$. The training set $\mathcal{D} = \{(\mathcal{I}_1, \mathbf{c}_1), \ldots, (\mathcal{I}_D, \mathbf{c}_D)\}$ of image-caption pairs is said to be weakly labeled if the absence of a keyword from caption $\mathbf{c}_i$ does not necessarily mean that the associated concept is not present in $\mathcal{I}_i$. This is usually the case in practical scenarios, since each image is likely to be annotated with a small caption that only identifies the semantics deemed as most relevant to the labeler. We assume weak labeling in the remainder of this work.

The design of a QBSE retrieval systems requires two main components. The first is a semantic image labeling system that, given a novel image $\mathcal{I}$, produces a vector of posterior probabilities $\pi = (\pi_1, \ldots, \pi_L)^T$ for the concepts in $\mathcal{L}$. This can be seen as a feature transformation, from the space of image measurements $\mathcal{X}$ to the L-dimensional probability simplex $\mathcal{S}_L$, i.e. a mapping $\mathbf{\Pi} : \mathcal{X} \rightarrow \mathcal{S}_L$ such that $\mathbf{\Pi}(\mathcal{I}) = \pi$. Each image can, therefore, be seen as a point π in $\mathcal{S}_L$, i.e. the probability distribution of a multinomial random variable defined on the space of semantic concepts. We will refer to this representation as the *semantic multinomial* (SMN) that characterizes the image. The second component is a query-by-example function on $\mathcal{S}_L$. This is a function that, given the SMN that characterizes a query image, returns the most similar SMN among those derived from all database images, i.e. $f : \mathcal{S}_L \rightarrow \{1, \ldots, D\}$ such that

$f(\pi) = \arg\max_i s(\pi, \pi_i)$ where π is the query SMN, π_i the SMN that characterizes the i^{th} database image, and $s(\cdot, \cdot)$ an appropriate similarity function. Given that SMNs are probability distributions, a natural similarity function is the Kullback-Leibler divergence

$$s(\pi, \pi') = KL(\pi || \pi_i) = \sum_{i=1}^{L} \pi_i \log \frac{\pi'_i}{\pi_i}, \tag{1}$$

which we adopt in this work. We next present our implementation of $\mathbf{\Pi}$.

3.2 Image Labeling

The mapping $\mathbf{\Pi}$ can be implemented with any semantic labeling system that produces posterior probabilities for the concepts in $\mathcal{L}$ given an image $\mathcal{I}$. We build on our previous work in the area, by adopting the weakly supervised method of [9], briefly reviewed in the remainder of this section. This method formulates semantic image labeling as an L-ary classification problem. Images are represented as bags of localized measurements $I = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathcal{X}$ is a vector of image measurements (or *visual features*), and semantic labeling is achieved through the introduction of 1) a random variable W, which takes values in $\{1, \dots, L\}$, so that $W = i$ if and only if $\mathbf{x}$ is a sample from the concept w_i, and 2) a set of class-conditional distributions $P_{\mathbf{X}|W}(\mathbf{x}|i), i \in \{1, \dots, L\}$ for visual features given the semantic class.

For all i, the semantic class density $P_{\mathbf{X}|W}(\mathbf{x}|i)$ is learned from a training set $\mathcal{D}_i$ of images labeled with the annotation w_i, using a *hierarchical estimation* procedure first proposed, in [7], for image indexing. This procedure is itself composed of two steps. First, a Gaussian mixture model is learned for each image in $\mathcal{D}_i$, using the classical expectation-maximization (EM) algorithm.This originates a sequence of mixture density estimates $P_{\mathbf{X}|L,W}(\mathbf{x}|l, i) = \sum_k \pi_{i,l}^k \mathcal{G}(\mathbf{x}, \mu_{i,l}^k, \Sigma_{i,l}^k)$, where $\pi_{i,l}^k$ is a probability mass function such that $\sum_k \pi_{i,l}^k = 1$, $\mathcal{G}(\mathbf{x}, \mu, \Sigma)$ a Gaussian density of mean μ and covariance Σ, and L a hidden variable that indicates the image number. Omitting, for brevity, the dependence of the mixture parameters on the semantic class i, and assuming that each mixture has K components, this produces $D_i K$ mixture components of parameters $\{\pi_j^k, \mu_j^k, \Sigma_j^k\}, j = 1, \dots, D_i$, $k = 1, \dots, K$. The second step is an extension of the EM algorithm, which clusters the Gaussian components into a T-component mixture, where T is the desired number of components at the semantic class level. Denoting by $\{\pi_c^t, \mu_c^t, \Sigma_c^t\}, t = 1, \dots, T$ the parameters of the class mixture, this algorithm iterates between the following steps.
E-step: compute

$$h_{jk}^t = \frac{\left[\mathcal{G}(\mu_j^k, \mu_c^t, \mathbf{\Sigma}_c^t) e^{-\frac{1}{2} trace\{(\mathbf{\Sigma}_c^t)^{-1} \mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k N} \pi_c^t}{\sum_l \left[\mathcal{G}(\mu_j^k, \mu_c^l, \mathbf{\Sigma}_c^l) e^{-\frac{1}{2} trace\{(\mathbf{\Sigma}_c^l)^{-1} \mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k N} \pi_c^l}, \tag{2}$$

where N is a user-defined parameter (see [7] for details).
M-step: set

$$(\pi_c^t)^{new} = \frac{\sum_{jk} h_{jk}^t}{PK} \tag{3}$$

$$(\mu_c^t)^{new} = \sum_{jk} w_{jk}^t \mu_j^k, \text{where } w_{jk}^t = \frac{h_{jk}^t \pi_j^k}{\sum_{jk} h_{jk}^t \pi_j^k} \tag{4}$$

$$(\mathbf{\Sigma}_c^t)^{new} = \sum_{jk} w_{jk}^t \left[\mathbf{\Sigma}_j^k + (\mu_j^k - \mu_c^t)(\mu_j^k - \mu_c^t)^T \right]. \tag{5}$$

Notice that the number of parameters in each image mixture is orders of magnitude smaller than the number of feature vectors in the image itself. Hence the complexity of estimating the class mixture parameters is negligible when compared to that of estimating the individual mixture parameters for all images in the class. It follows that the overall training complexity is equivalent to that required to train a QBVE retrieval system based on the minimum probability of error cost [10].

4 Experimental evaluation

In this section we present results of an evaluation of QBSE on a number of databases. The goal is to answer two main questions. The first is how well QBSE performs, comparatively to QBVE, in the standard scenario where the queries are from classes which belong to the semantic space on which the system was trained. The second deals with generalization, namely how well QBSE performs on images from classes outside this space.

4.1 Experimental protocol

In all experiments the semantic feature space was learned from the Corel database used in [3, 5]. This database, henceforth called *Corel50*, consists of 5,000 images from 50 Corel Stock Photo CDs, divided into a training set of 4,500, and a test set of 500 images. Each CD includes 100 images of the same topic, and each image is labeled with 1-5 semantic concepts. Overall there are 371 keywords in the data set, leading to a 371-dimensional semantic simplex. In terms of image representation, all images were normalized to size 181×117 or 117×181 and converted from RGB to the YBR color space. Image observations were derived from 8×8 patches obtained with a sliding window, moved in a raster fashion. A feature transformation was applied to this space by computing the 8×8 discrete cosine transform (DCT) of the three color components of each patch. The parameters of the semantic class mixture hierarchies were learned in the subspace of the resulting 192-dimension feature space composed of the first 21 DCT coefficients from each channel. For all experiments, the SMN associated with each image was computed with these semantic class distributions.

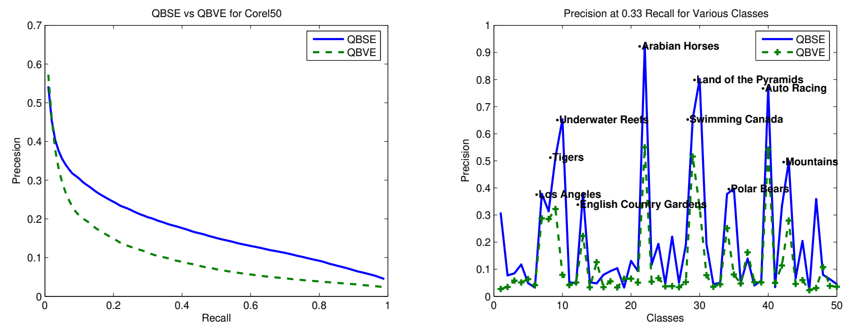

Fig. 2. Left: average PR for QBSE and QBVE on *Corel50*. Right: Precision for 50 classes of the *Corel50* database.

To evaluate retrieval performance, we relied on the standard precision/recall (PR) curves and carried out tests on three databases *Corel50, Flickr18* and *Corel15*. In all cases there is a clear ground truth regarding which images are relevant to a given query (e.g., images labeled as belonging to the same Topic on *Corel50* data set.). The first set of experiments were done using the 500 test images of *Corel50* as the *query database* and the 4500 training images as the *retrieval database*. The closest match in the retrieval database was found for each image in the query database, PR measured, and averaged over all queries. Note that, in this experiment, the query images belong to the semantic classes that the system was trained to recognize, i.e. they are in the semantic simplex. This is the usual evaluation scenario for semantic image retrieval [3, 5]. To analyze the generalization ability of QBSE, we have also used two completely new image databases. The first, *Flickr18*, was built with 1,800 images from 18 classes downloaded from `www.flickr.com`. These were classified according to the manual annotations provided by the online users. The second, *Corel15*, consisted of 1,500 images from another These were classified based on the CD themes, which were non-overlapping with the semantic class learned from *Corel50*. For both databases, 20% of randomly selected images served as the *query database* and the remaining 80% as the *retrieval database*.

4.2 Performance within the semantic simplex

Figure 2 a) presents the PR curves obtained on *Corel50* with QBVE and QBSE. It can be seen that the precision of QBSE is significantly higher than that of QBVE at most levels of recall. QBVE performs well at low-levels of recall, confirming its well known ability to generalize along the *dimension of query diversity*, i.e. to find most images that are *visually similar* to the query. However, its performance is dramatically inferior to that of QBSE, which is able to generalize much more broadly along the *dimension of image similarity*. Figure 2 presents a comparison of the relative performance for individual classes, namely the precision at 0.33 recall. It is clear that QBSE outperforms QBVE for almost all

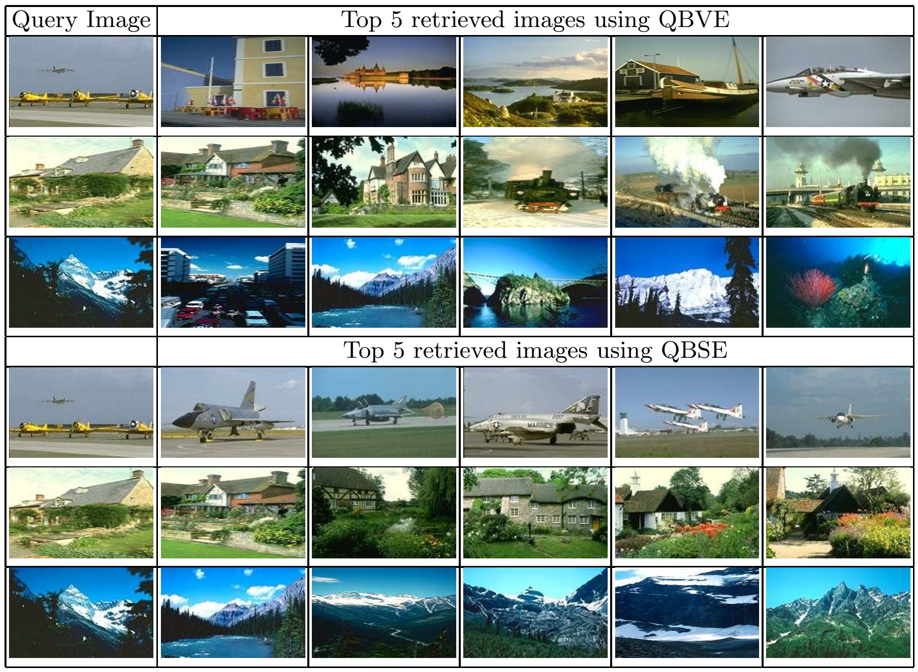

Fig. 3. QBVE and QBSE retrieval from *Corel50*. The first column shows the query image and columns $2-6$ the top 5 database matches.

classes. In 5 classesthe absolute precision gain is greater than 0.30. The benefits of QBSE are illustrated in Fig. 3, where we present the results for some queries under both QBVE and QBSE. Note, for example, that for the query image containing *yellow airplanes* and a large area of *blue sky*, QBVE tends to retrieve images with *yellowish* foregrounds, against a the backdrop of *blue*, that have little connection to the *airplane* theme. Due to its higher level of abstraction, QBSE is successfully able to generalize the main semantic concepts of *airplanes, ground* and *sky*.

4.3 Semantic simplex mismatch

One question which is always of relevance for semantic retrieval systems is that of how well they generalize for image classes not seen during training. QBVE is obviously not affected by this problem, and provides a good comparative benchmark. To address this question, we tested QBSE on two other image sets (*Flickr18* and *Corel15*) with a significant number of semantic classes that are not covered by *Corel50*. Note that this is true for both the *query* and *retrieval* databases constructed. While there is a semantic space associated with these databases, and this space necessarily has some overlap with that of *Corel50* (e.g., all databases contain images with "sky"), these two datasets were explicitly

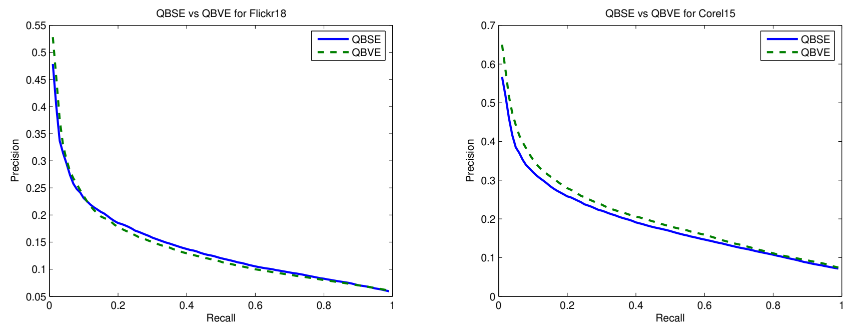

Fig. 4. PR curves for QBSE and QBVE on *Flickr18* (left) and *Corel15* (right).

constructed to minimize this overlap insofar as possible. Figure 4 presents the PR curves obtained in *Flickr18* and *Corel15*. It can be seen that, in both cases, the performance of QBSE is equivalent to that of QBVE. This indicates that QBSE has good generalization: in the worst case its performance drops to the levels that were possible with visual similarity.

Acknowledgment This research was supported by a gift from Google Inc., NSF Career award IIS-0448609, and NSF award IIS-0534985.

References

1. K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003.
2. D. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the* 26^{th} *Intl. ACM SIGIR Conf.*, pages 127–134, 2003.
3. V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.
4. R. Picard. Digital Libraries: Meeting Place for High-Level and Low-Level Vision. In *Proc. Asian Conf. on Computer Vision*, December 1995, Singapore, USA.
5. S.L.Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004.
6. A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval: the end of the early years. In *PAMI*, 22(12):1349–1380, 2000.
7. N. Vasconcelos. Image Indexing with Mixture Hierarchies. In *CVPR.*, Kawai, Hawaii, 2001.
8. S. Sclaroff, M. L. Cascia, S. Sethi, and L. Taycher. Unifying textual and visual cues for content-based image retrieval on the world wide web. *Computer Vision and Image Understanding*, 75(1-2):8698, 1999.
9. G. Carneiro and N. Vasconcelos. Formulating Semantics Image Annotation as a Supervised Learning Problem. In *CVPR*, San Diego, 2005.
10. N. Vasconcelos. Minimum Probability of Error Image Retrieval., In *IEEE Transactions on Signal Processing* Vol. 52, NO. 8, 2004