Implementing Relevance Feedback as Convolutions of Local Neighborhoods on Self-Organizing Maps

Markus Koskela, Jorma Laaksonen, and Erkki Oja

Laboratory of Computer and Information Science, Helsinki University of Technology
P.O.BOX 5400, 02015 HUT, Finland
{markus.koskela,jorma.laaksonen,erkki.oja}@hut.fi

Abstract. The Self-Organizing Map (SOM) can be used in implementing relevance feedback in an information retrieval system. In our approach, the map surface is convolved with a window function in order to spread the responses given by a human user for the seen data items. In this paper, a number of window functions with different sizes are compared in spreading positive and negative relevance information on the SOM surfaces in an image retrieval application. In addition, a novel method for incorporating location-dependent information on the relative distances of the map units in the window function is presented.

1 Introduction

The data organization provided by the Self-Organizing Map (SOM) [1] can be utilized in searching for interesting data items. Due to the topology-preservation property of the SOM, neighboring map units contain similar feature vectors. If we already know that certain map units contain data items which are in some manner similar to the item we are interested in, a natural strategy is to focus the further search in the neighborhoods of these map units. This kind of setting arises, e.g. in iterative multi-round information retrieval where, on each query round, the user marks the retrieved items as relevant or nonrelevant to the query. The system then uses this information in estimating what the user is looking for. This kind of iterative refinement of a query is known as *relevance feedback* in information retrieval literature [2].

Content-based image retrieval (CBIR) has been a subject of recent intensive research effort. It differs considerably from textual information retrieval as, unlike text that consists of words, images do not consist of such basic building blocks which could directly be utilized in retrieval applications. Instead, the retrieval is based on visual features extracted from the images and alternative retrieval paradigms must be used. One common approach is *query by example*, where the user specifies her object of interest by giving or pointing out examples of interesting or relevant images. Relevance feedback is essential here, as the systems are normally not capable of returning the desired image on the first query round [3]. A CBIR system implementing relevance feedback tries to learn the optimal correspondence between the high-level concepts people use and the low-level features obtained from the images. The user thus does not need to explicitly specify weights for different features as the weights are formed implicitly by the system. This is desirable, as it is generally a difficult task to give low-level features such weights which would coincide with human perception of images.

2 PicSOM

The PicSOM [4, 5] image retrieval system is a framework for research on methods for content-based image retrieval. The methodological novelty of PicSOM is to use several parallel Self-Organizing Maps trained with separate data sets. After training the SOMs, their map units are connected with the images of the database. This is done by locating the best-matching map unit (BMU) for each image. Also, among the images which have a common BMU, the best-matching one is used as a visual label for that unit. As a result, the different SOMs impose different similarity relations on the images and the system is able to adapt to different kinds of retrieval tasks. The spreading of the positive and negative responses on the SOMs has been an integral part of the system from the beginning, but it has not been thoroughly examined until now.

Instead of the standard SOM version, PicSOM uses a special form of the algorithm, the Tree Structured Self-Organizing Map (TS-SOM) [6]. The hierarchical TS-SOM structure is useful for two purposes. First, it reduces the complexity of training large SOMs by exploiting the hierarchy in finding the BMU for an input vector. Second, the hierarchical representation of the image database produced by a TS-SOM can be utilized in browsing the images in the database.

The PicSOM home page including a working demonstration of the system for public access is located at *http://www.cis.hut.fi/picsom*.

3 Relevance Feedback with Self-Organizing Maps

The basic assumption in the PicSOM method is that images similar according to specific visual features are located near each other on the SOM surfaces. Therefore, we are motivated to spread the relevance information given by the user to the shown images also to the neighboring units. This is done as follows. All relevant images are first given equal positive weight inversely proportional to the number of relevant images. Likewise, nonrelevant images receive negative weights that are inversely proportional to their total number. The overall sum of these relevance values is thus zero. For each SOM layer, the values are then mapped from the images to their BMUs where they are summed. Finally, the resulting sparse value fields on the SOM surfaces are low-pass filtered to produce qualification values for each SOM unit and its associated images. This process is illustrated in Figure 1.

Content descriptors that fail to coincide with the user's conceptions mix positive and negative values in nearby map units. Therefore, they produce lower qualification values than those descriptors that match the user's expectations and

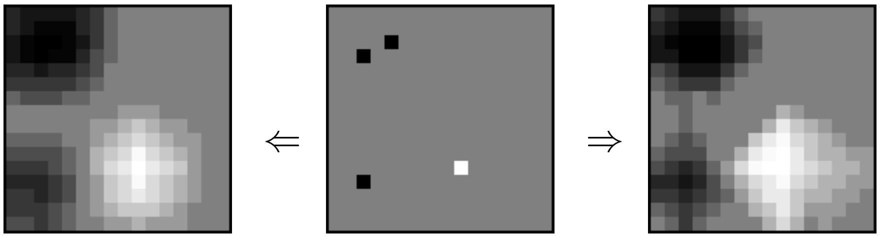

Fig. 1. An example of how positive and negative map units, shown with white and black marks on the middle figure, are convolved with shift-invariant (leftmost figure) and location-dependent (rightmost figure) window functions.

impression of image similarity. As a consequence, the different content descriptors and the TS-SOMs formed from them do not need to be explicitly weighted as the system automatically takes care of weighting their opinions.

On each SOM, we first search for a fixed number, say 100, of unseen images with the highest qualification values. After removing duplicates, the second stage of processing is carried out. Now, the qualification values of all images in this combined set are summed up on all SOMs. 20 images with the highest total qualification values have then been used in the experiments as the result of the query round.

3.1 Shift-Invariant Window Functions

Spreading of the response values can be performed by convolving the sparse value fields with a tapered (or rectangular) window or kernel function. The one-dimensional convolution of signal $x[n]$ and window $w[n]$ is a basic signal processing operation defined as

$$y[n] = x[n] * w[n] = \sum_{k=-M}^{M} x[n-k]w[k] \; . \tag{1}$$

On SOM surfaces the convolutions have to be two-dimensional. Due to computational reasons this has been implemented as one-dimensional horizontal convolution followed by one-dimensional vertical convolution. This can be done because the convolution kernels we have used have been separable and shift-invariant. The following window functions have been used in the experiments:

$$w_r[n] = 1 \qquad \text{(rectangular)} \tag{2}$$

$$w_t[n] = \frac{M-|n|}{M} \qquad \text{(triangular)} \tag{3}$$

$$w_g[n] = e^{-(\frac{n}{\alpha})^2} \qquad \text{(truncated Gaussian)} \tag{4}$$

$$w_x[n] = e^{-\frac{|n|}{\beta}} \qquad \text{(truncated exponential)} \tag{5}$$

The truncated Gaussian and exponential windows require a parameter controlling the decay of the window. Here, α and β have been selected so that $w_g[\pm\frac{M}{2}] = w_x[\pm\frac{M}{4}] = \frac{1}{2}$.

The length of the window, $N = 2M + 1$, is the predominant parameter of any window function. With small N, the search expands only to the immediate neighbors of the relevant items. As N grows the search area widens. As the computational complexity of the convolution is linearly dependent on the window length, it is beneficial to be able to use as small windows as possible.

3.2 Location-Dependent Window Functions

Information on the distances between neighboring SOM codebook vectors in the feature space has earlier been used mainly in visualization [7, 1]. If the relative distance of two SOM units is small, they can be regarded as belonging to the same cluster and, therefore, the relevance response should easily spread between the neighboring map units. Cluster borders, on the other hand, are characterized by large distances and the spreading of responses should be less intensive.

For each neighboring pair of map units according to 4-neighborhood, say i and j, the distance in the original feature space is calculated and then scaled so that the average neighbor distance is equal to one. The normalized distances d_{ij} are then used for calculating location-dependent convolutions with two alternative methods, illustrated in Figure 2. The "path" method uses dynamic programming to solve the minimum path length along the 4-neighborhood grid between two arbitrary map units i and j. Given a maximum allowed distance M, we can calculate and tabularize the between-node distances d_{ij} for non-neighboring map units. Then the two-dimensional convolution functions were formed from equations (2–5) by setting $n = d_{ij}$.

In the alternative "sum" method, a computationally faster solution is obtained by performing one-dimensional location-dependent convolution first horizontally with kernel values obtained again from equations (2–5) with $n = d_{ij}$. The result of the horizontal convolution was then similarly convolved with vertical one-dimensional location-dependent kernels. As the order of the successive one-dimensional convolutions now matters, the original impulse-valued SOM surface was convolved again, now first vertically and then horizontally, and the two

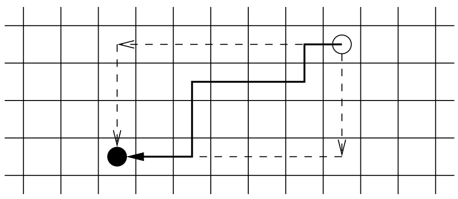

Fig. 2. An illustration of the two methods for calculating the location-dependent convolutions on the SOM grid. In the "path" method, the minimum path $\circ \rightarrow \bullet$ is solved with dynamic programming. In the "sum" method, horizontal and vertical one-dimensional location-dependent convolutions are calculated in both orders and then averaged.

slightly-different convolution results were averaged. In preliminary experiments it was observed that the difference of the two methods is not substantial in this setting and that the "path" method is computationally not feasible with large window sizes. Therefore, only the "sum" method was used in the experiments.

4 Experiments

We used an image database containing 59 995 images from the Corel Gallery 1 000 000 product. From the database images, we have created manually five ground truth image classes: **faces** (1115 images, *a priori* probability 1.85%), **cars** (864 images, 1.44%), **planes** (292 images, 0.49%), **sunsets**, (663 images, 1.11%), and **horses**, (486 images, 0.81%). As image features, we used a subset of MPEG-7 [8] content descriptors for still images, viz. *Scalable Color*, *Dominant Color*, *Color Structure*, *Color Layout*, *Edge Histogram*, and *Region Shape*.

The image queries are always started with one reference image that belongs to the image class in question. Therefore, initial browsing is not needed and we can limit the search exclusively to the 256×256-sized bottommost TS-SOM levels. From each of the above classes, 20 random images were selected to the set of reference images and an image query was then run for each of these images. Then we get the final results by averaging the results of the 100 individual runs.

For performance evaluation, we used the τ measure [5] which coincides with the question "how large portion of the whole database needs to be browsed through until, on the average, the searched image will be found". The τ measure can be obtained for a relevance feedback system by simulating the responses of a human user. This can be done by examining each output of the system and marking the returned images either as relevant or non-relevant according to whether they belong to a ground truth image class $\mathcal{C}$. From this data, we calculate the average number of shown images needed before a hit occurs. The τ measure for $\mathcal{C}$ is then obtained by dividing the average number of shown images by the size of the database. The measure yields a value in the range $\tau \in [\frac{\rho_\mathcal{C}}{2}, 1 - \frac{\rho_\mathcal{C}}{2}]$ where $\rho_\mathcal{C}$ is the *a priori* probability of the class $\mathcal{C}$. For values $\tau < 0.5$, the performance of the system is thus better than random picking of images and, in general, the smaller the τ value the better the performance.

The resulting values of the τ measure with different window functions are shown in Figure 3. First, it can be seen that a small window length is sufficient. Best results are obtained with $2 \leq M \leq 4$. Second, the window function should be tapered, as the rectangular window clearly performs worse than the others. Otherwise the shape of the window seems not to be a significant factor. Third, the results with shift-invariant and location-dependent window functions are quite similar. This is probably due to the relatively large size of the maps compared to the size of the database. In many cases using smaller SOMs is preferable and then location dependency is likely to be more important as the images are mapped more densely to map units and the convolutions are more likely to cross cluster borders.

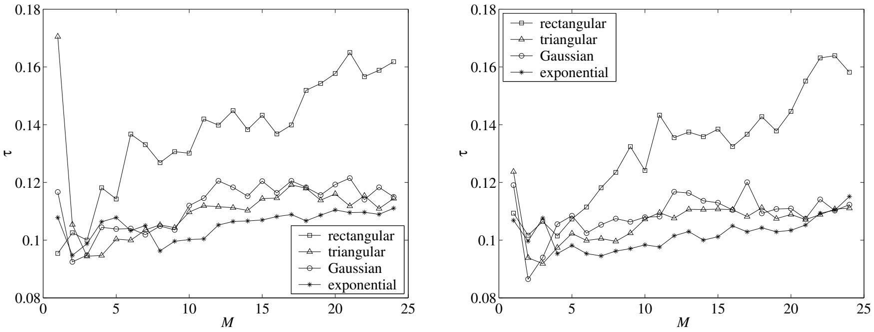

Fig. 3. Average τ values with different shift-invariant (left figure) and location-dependent (right figure) window functions of varying width M.

5 Conclusions

In this paper, experiments on implementing relevance feedback on multiple SOMs were presented. The SOM surfaces are convolved with a window function in order to spread the relevance feedback responses provided by a human user. Different shapes and sizes of the convolution kernel have been studied. In addition, a method for combining location-dependent distance information of the map units in the window function was presented. Here, using the location-dependent window functions did not improve the results. Still, they may prove out to be useful with smaller SOMs. Also, it was seen that using small window sizes suffices, which is computationally advantageous in actual implementations.

References

1. Kohonen, T.: Self-Organizing Maps. Third edn. Volume 30 of Springer Series in Information Sciences. Springer-Verlag (2001)
2. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. Computer Science Series. McGraw-Hill (1983)
3. Lew, M.S., ed.: Principles of Visual Information Retrieval. Springer-Verlag (2000)
4. Laaksonen, J.T., Koskela, J.M., Laakso, S.P., Oja, E.: PicSOM - Content-based image retrieval with self-organizing maps. Pattern Recognition Letters **21** (2000) 1199–1207
5. Laaksonen, J., Koskela, M., Laakso, S., Oja, E.: Self-organizing maps as a relevance feedback technique in content-based image retrieval. Pattern Analysis & Applications **4** (2001) 140–152
6. Koikkalainen, P., Oja, E.: Self-organizing hierarchical feature maps. In: Proc. IJCNN-90, International Joint Conference on Neural Networks, Washington, DC. Volume II., Piscataway, NJ, IEEE Service Center (1990) 279–285
7. Ultsch, A., Siemon, H.P.: Kohonen's self organizing feature maps for exploratory data analysis. In: Proc. INNC'90, Int. Neural Network Conf., Dordrecht, Netherlands, Kluwer (1990) 305–308
8. MPEG: Overview of the MPEG-7 standard (version 5.0) (2001) ISO/IEC JTC1/SC29/WG11 N4031.