Optimal Combination of SOM Search in Best-Matching Units and Map Neighborhood*

Mats Sjöberg and Jorma Laaksonen

Helsinki University of Technology TKK,
Department of Information and Computer Science,
P.O. Box 5400, FI-02015 TKK, Finland

Abstract. The distribution of a class of objects, such as images depicting a specific topic, can be studied by observing the best-matching units (BMUs) of the objects' feature vectors on a Self-Organizing Map (SOM). When the BMU "hits" on the map are summed up, the class distribution may be seen as a two-dimensional histogram or discrete probability density. Due to the SOM's topology preserving property, one is motivated to smooth the value field and spread out the values spatially to neighboring units, from where one may expect to find further similar objects. In this paper we study the impact of using more map units than just the single BMU of each feature vector in modeling the class distribution. We demonstrate that by varying the number of units selected in this way and varying the width of the spatial convolution one can find an optimal combination which maximizes the class detection performance.

1 Introduction

In many crucial information processing applications, such as high-level indexing and querying on multimedia data, it has proven to be very useful to have models of semantically related classes, i.e. meaningful subsets of the full dataset under study [1]. When a Self-Organizing Map (SOM) [2] is trained on a large dataset, mapping the data vectors of some semantic class to their best-matching units (BMUs) produces a distribution characterizing that particular class in the context of the full dataset. For example, when studying a database of animal images, one could map the class of objects depicting lions on a SOM trained from color features extracted from all the images. The SOM may then be used for example in an image retrieval task for detecting images of lions in a new batch of unannotated images.

The rest of this paper is organized as follows: Section 2 describes modeling of class distributions with BMUs, Section 3 smoothing in the spatial and feature domains. In Section 4 an image retrieval experiment is shown, and finally conclusions are drawn in Section 5.

* The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under *grant agreement* n° 216529, Personal Information Navigator Adapting Through Viewing, PinView. Mats Sjöberg has been supported by a grant from the Nokia Foundation.

J.C. Príncipe and R. Miikkulainen (Eds.): WSOM 2009, LNCS 5629, pp. 281–289, 2009.

2 Modeling Class Distributions with BMUs

For any database of objects, feature vectors can be extracted for analyzing the properties of the objects. If the features are selected properly they should be of moderate dimensionality, while still preserving semantically important information of the objects and their distribution. Figure 1 (left), visualizes how the original very-high-dimensional pattern space is first projected to a lower-dimensional feature space, the vectors of which are then used in training a SOM. The dark areas in the figure illustrate how a class of objects might be projected, ideally to a compact distribution in the feature space if the discriminative properties of the class are well represented in the feature extraction process.

If the best-matching units of the objects of a specific semantic class are marked with a positive impulse, the "hits" on a SOM surface form a sparse value field. When these values are summed up and properly normalized, the formed distribution can be seen as a two-dimensional discrete probability density that characterizes the object class. Such distributions were studied in an earlier article [3] in the context of our content-based retrieval system PicSOM [4], and information-theoretic measures were proposed for evaluating their properties.

Due to the topography-preserving property of the SOM, we can now expect to find more similar objects in the map areas with many nearby hits. In order to spread the values to such neighboring units the value field is, in the PicSOM system, low-pass filtered with a tapered kernel. This facilitates finding new unannotated objects of the same class, and also aids in visual inspection of the map distribution. It also serves to emphasize areas with many hits close-by and deemphasize areas with only a few sporadic hits. A visual example is shown in Figure 1 (right) where a class of video frames depicting scenes with "explosion or fire" have been mapped to a SOM trained from Color Layout feature vectors. Areas occupied by objects of the concept in question are shown with gray shades. Clearly the hits from this class seem to be concentrated into the bottom right corner of the map.

These class-conditional distributions or class models can be considered as estimates of the true distributions of the semantic concepts in question, not on the original feature spaces, but on the discrete two-dimensional grids defined by the used SOMs. Thereby, instead of modeling probability densities in the

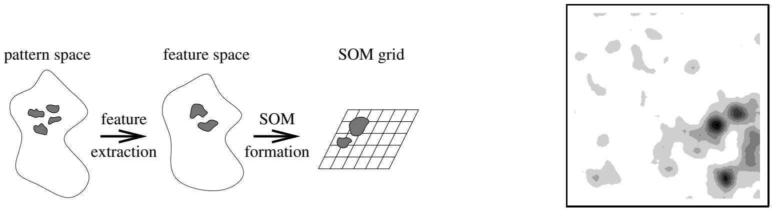


Fig. 1. Left: Stages in creating a class model from the very-high-dimensional pattern space through the high-dimensional feature space to the two-dimensional SOM grid. Right: An example of image class model "explosion or fire" on a Color Layout SOM.

high-dimensional feature spaces, the PicSOM system is essentially performing kernel-based estimation of discrete class densities over the SOM grid. Depending on the variance of the kernel function, these kernels will overlap and weight vectors close to each other will partially share each other's probability mass. As an example, the most representative objects of a given semantic class can be obtained by locating those SOM units, and the objects mapped to these units, that have the highest responses on the estimated class distribution.

In this paper, we study the use of more than just one BMU when mapping the members of a semantic class to a SOM. We sort all the model vectors of the map in ascending order of the distance to the input vector and apply a weighting kernel to this set, giving the highest weight to the best-matching unit, and decreasing weights according to the list rank. By varying the width of this kernel we can choose the number of nearest units selected for each input vector. We call this number the "BMU depth". For example, for BMU depth equal to three, we select the second and third best-matching units (generally with decreasing weights) in addition to the normal BMU. Thus, we use both spatial SOM surface smoothing and smoothing in the BMU depth, i.e. we spread the "hit" values both in the SOM grid and feature space domains.

To compare, the WEBSOM system [5] for interactive browsing of large text document databases, used only the BMU depth approach, not spatial smoothing. An idea similar to ours was explored in [6], where the cluster structure of the data could be visualized on different levels of detail by varying the smoothing parameter (equivalent to our BMU depth). Another related concept is to force the map convolution to follow the form of the U-matrix, i.e. the convolution span is inversely proportional to the distance between the SOM units [7]. The advantage of the proposed approach over U-matrix based weighting is computational simplicity; instead of tuning the convolution separately for each unit we need only select a small set of best-matching units. Finding BMUs is very fast, especially in the PicSOM system that implements the tree-structured SOM variant [8] which does BMU search in logarithmic time.

3 Smoothing in the SOM and Feature Spaces

In this paper we introduce smoothing in the feature space domain in combination with the traditional spatial SOM surface smoothing. Instead of only using the single best-matching unit, we order the list of SOM model vectors by increasing distance from the input vector. Such ordered lists can be generated off-line for each database object storing only a restricted set of the best matches.

Let us assume that we have a set R of training set objects j whose membership value r_j in the studied object class is known. Then

$$r_j = \begin{cases} +\rho_+ & \text{, if } j \text{ is a member of the class} \\ 0 & \text{, if } j\text{'s membership in the class is unknown ,} \\ -\rho_- & \text{, if } j \text{ is not a member of the class} \end{cases} \tag{1}$$

where ρ_+ and ρ_- are properly selected non-negative weights for the member and non-member samples, respectively. In PicSOM, the values of ρ_+ and ρ_- have been inverses of the number of positive and negative samples and consequently $\sum_j r_j = 0$.

A membership score for any point $\mathbf{x}$ can then be estimated as a sum of kernel functions $h_j(\cdot)$ centered in the locations of the points $\mathbf{x}_j$ with known membership assessments:

$$r(\mathbf{x}) = \sum_{j \in \mathrm{R}} r_j h_j(\mathbf{x} - \mathbf{x}_j). \tag{2}$$

In the PicSOM system, the kernel functions $h_j(\mathbf{x} - \mathbf{x}_j)$ have been replaced by the use of a function $h(\cdot)$ that can be calculated from the difference between the BMU coordinates on the SOM surfaces. Let $\mathbf{b}(\mathbf{x}) = \big(b_x(\mathbf{x}),\ b_y(\mathbf{x})\big)$ denote the discrete two-dimensional coordinates of the best-matching unit of $\mathbf{x}$. One should note that the values of the BMU function $\mathbf{b}(\mathbf{x}_j)$ can be calculated and tabulated offline for each object j as soon as the SOM has been trained. The membership value estimate for $\mathbf{x}$ can thus be written as

$$\begin{aligned} r(\mathbf{x}) &= \sum_{j \in \mathrm{R}} r_j h\big(b_x(\mathbf{x}) - b_x(\mathbf{x}_j),\ b_y(\mathbf{x}) - b_y(\mathbf{x}_j)\big) \\ &= \sum_{j \in \mathrm{R}} r_j g\big(b_x(\mathbf{x}) - b_x(\mathbf{x}_j)\big)\ g\big(b_y(\mathbf{x}) - b_y(\mathbf{x}_j)\big)\ . \end{aligned} \tag{3}$$

The latter notation follows from the practice of using separable and symmetric kernels $h(\cdot)$. Now the extent and shape of the scalar function $g(\cdot)$ determines the effect of the SOM surface smoothing. In PicSOM we have used a simple triangular kernel with different widths.

In order to take the BMU smoothing into the formulation, one needs to extend the BMU function $\mathbf{b}(\mathbf{x}_j)$ with the BMU depth index k to be $\mathbf{b}_k(\mathbf{x}_j) = \big(b_{x,k,j},\ b_{y,k,j}\big)$, where $k = 1, \ldots, k_{\max}$. Now we have

$$r(\mathbf{x}) = \sum_{j \in \mathrm{R}} r_j \sum_{k=1}^{k_{\max}} f(k) g\big(b_x(\mathbf{x}) - b_{x,k,j}\big)\ g\big(b_y(\mathbf{x}) - b_{y,k,j}\big)\ . \tag{4}$$

Function $f(k)$ determines the extent of smoothing in the BMU order. Note that the BMUs $b_{x,k,j}$ and $b_{y,k,j}$ of the objects j in the database can be calculated and tabulated offline.

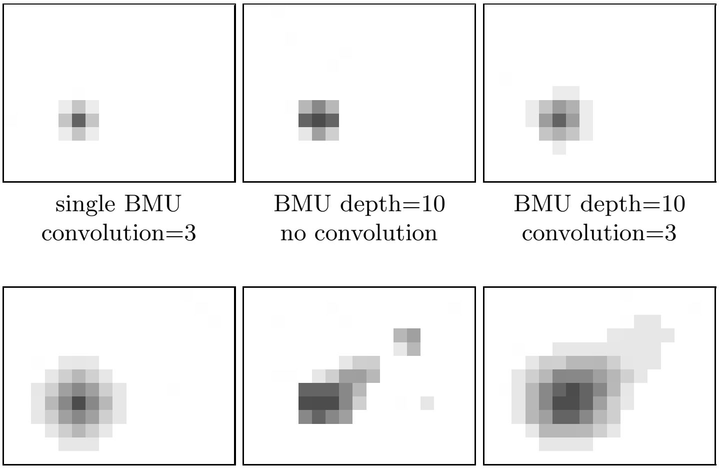
A linear kernel $f(\cdot)$ has been used in our experiments, i.e. the weight decreases linearly with the rank in the ordered list. We have also tried several other shapes of $f(\cdot)$, including Gaussian and one-per-rank, but the linear kernel worked best overall. In our experiments, the most important parameter turned out to be the width of the kernel, not the particular type.

Figure 2 illustrates the smoothing in the two domains separately and combined. The images depict a small neighborhood of a SOM surface trained with Scalable Color features, and a single image of an airplane mapped to its BMU.

The first column shows this single BMU convolved on the map surface with two kernel widths: 3 and 7. This illustrates the traditional approach in PicSOM, where only the map topology is taken into account. The second column shows the same BMU, but now using a BMU depth of 10 or 30, and no map convolution. The values are thus spread to the 10, respectively 30, nearest units in the feature space. It can be readily observed that the two cases on the first row are very similar. Not surprisingly, the nearest units are located closely around the best-matching unit. A map convolution width of 3 encompasses roughly the same amount of units. The difference is that the selection in the first column is done based on the map grid neighborhood, and in the second column on the feature space neighborhood.

On the second row of images in Figure 2 we can see a difference, when the topology of the feature space stretches the BMU depth distribution to the upper right, while the center-symmetric regular map convolution does not take this into account. A similar effect could be achieved with the method of tuning the map convolution to the U-matrix distances [7]. The proposed method, however, is computationally much simpler.

In Figure 2 the last column shows the result of combining the two first columns, i.e. first the values are smoothed in the BMU depth domain, and then the result is smoothed in the SOM surface domain. This combined approach turned out to give the best results in our concept detection experiments.



single BMU
convolution=3

BMU depth=10
no convolution

BMU depth=10
convolution=3

single BMU
convolution=7

BMU depth=30
no convolution

BMU depth=30
convolution=7

Fig. 2. The first column shows a single BMU with SOM convolutions of increasing width. The single impulse is marked with black and decreasing values with shades of gray, with white indicating zero. The second column shows the hits with increasing BMU depth, without SOM convolution. The last row shows the combination of both the BMU depth and the SOM convolution.

4 Image Retrieval Experiment

We experimented with SOMs trained on several different features extracted from a set of images from the Pascal Visual Object Class (VOC) 2007 Challenge[1]. The VOC dataset includes several predefined object classes including images annotated according to class membership. The classes are: *aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train* and *TV/monitor*. The full dataset of 9963 images is divided roughly evenly into training and test sets.

4.1 Features

The features used were *Color SIFT, Edge Fourier, Edge Histogram, IPLD* and *Scalable Color*. These were selected from a larger set of features because they were the 5 best performing ones (both with and without variable BMU depth). The *Color SIFT* feature is a 256-bin histogram of Opponent-SIFT (opponent color space) features calculated from interest points detected with the Harris-Laplace algorithm [9]. *Edge Fourier* is a 16×16 FFT of a Sobel edge image, *Edge Histogram* is a histogram of five edge types in 4×4 subimages. The IPLD feature is based on 256-bin histograms of interest point features. The interest points were detected using a combined Harris-Laplace and Difference-of-Gaussian detector, and SIFT features [10] were calculated for each interest point. The Scalable Color is a Haar transform of the quantized HSV color histogram. Both Edge Histogram and Scalable Color are implemented following the MPEG-7 standard [11].

4.2 Performance Measures

Given a training set of example objects belonging to a specific class, one can now calculate the membership score of novel objects from a test set by using Eq. (4) as implemented in the PicSOM algorithm. If the correct answers are known the quality of the SOM model can be measured by standard information retrieval performance measures, such as precision and recall.

In this paper we have opted for the use of non-interpolated average precision (AP) as the performance measure. AP is formed by calculating the precision after each retrieved relevant object. The final measure is obtained by averaging these precisions over the total number of relevant objects, when the precision is defined to be zero for all non-retrieved relevant objects. This measure can be said to incorporate both precision and recall in a single number [12].

4.3 Experiment

The convolution width on the map surface was varied from 1 (a single impulse) to 20 units. This was deemed a realistic interval due to the 64×64 size of the maps. In the feature space, the convolution width, or BMU depth, $k_{\max}$

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

was varied from 1 to 100. For each run the average precision was calculated. We selected the parameters (convolution width, or both convolution width and BMU depth) with the best average precision results separately for each class and feature combination, with a single BMU and with variable BMU depth. The average precision was calculated as the average over a 6-fold cross validation in the training set, i.e. each of the six subsets were in turn left out and used for validation. Typically, the optimal map convolution was wider when using a single BMU. This is not surprising, since using a greater BMU depth $k_{\max}$ in Eq. (4) spreads the values to more units, and a smaller convolution width is needed to reach the same amount of units. The median spatial convolution width changed from 7 to 4 when introducing BMU depth.

The optimal BMU depths found are summarized in Table 1 for each class–feature combination. As can be seen the optimal $k_{\max}$ varies quite a lot, and for some cases the optimum is one, i.e. the same as the baseline algorithm which does not use more than one BMU. The feature-wise medians are shown in bold face at the bottom of each column. The class-wise medians are at the end of each row, and 25 in the bottom right corner is the median over the entire table. Some classes vary quite a lot, while some clearly seem to prefer a high (bus, dog, person) or low (aeroplane, horse) BMU depth.

Table 2 summarizes the percentage changes in average precision as measured in the test set when introducing variable BMU depths. Where the optimal BMU depth was one, i.e. there was no improvement above the baseline (in the training set), the table cell has been intentionally left empty. It can be seen that in most cases the result is an improvement in performance, however in some instances there is a small decrease. In some situations, for example the sheep class and

Table 1. Optimal BMU depths $k_{\max}$ for each class and feature: Color SIFT (cSIFT), Edge Fourier (EF), Edge Histogram (EH), IPLD and Scalable Color (SC). Medians of each row and column are shown in bold face. A priori probabilities of classes are shown in parentheses.

class	**cSIFT**	**EF**	**EH**	**IPLD**	**SC**	
aeroplane (4.47%)	1	5	10	1	10	**5**
bicycle (5.07%)	20	5	80	1	5	**5**
bird (6.24%)	1	10	45	10	10	**10**
boat (3.65%)	10	20	30	70	5	**20**
bottle (5.04%)	50	25	50	10	55	**50**
bus (3.81%)	20	50	60	55	100	**55**
car (15.42%)	1	15	75	5	55	**15**
cat (6.79%)	30	15	40	10	100	**30**
chair (11.21%)	30	1	5	80	65	**30**
cow (2.74%)	45	60	1	20	10	**20**
dining table (5.12%)	40	60	40	50	5	**40**
dog (8.66%)	35	95	70	75	85	**75**
horse (5.75%)	5	20	25	5	5	**5**
motorbike (4.84%)	30	45	25	5	40	**30**
person (42.08%)	100	20	45	65	65	**65**
potted plant (5.29%)	25	5	25	1	70	**25**
sheep (1.96%)	10	50	5	90	5	**10**
sofa (7.30%)	45	10	1	15	30	**15**
train (5.24%)	35	1	90	20	30	**30**
tv/monitor (5.36%)	5	10	5	75	50	**10**
	27.5	**17.5**	**35**	**17.5**	**35**	**25**

Table 2. Average precision changes in percent for each class and feature combination, given in percentage. Averages of each row and column are shown in bold face. A priori probabilities of classes are shown in parentheses.

class	cSIFT	EF	EH	IPLD	SC	
aeroplane (4.47%)		2.22	3.26		-0.85	**0.93**
bicycle (5.07%)	-1.84	-7.80	4.60		5.84	**0.16**
bird (6.24%)		0.12	-7.05	-5.77	-0.34	**-2.61**
boat (3.65%)	7.40	2.59	4.88	3.86	3.82	**4.51**
bottle (5.04%)	4.80	0.35	1.32	0.38	-10.27	**-0.69**
bus (3.81%)	25.45	4.84	-3.35	-0.49	-0.01	**5.29**
car (15.42%)		-4.06	-1.07	0.19	-3.42	**-1.67**
cat (6.79%)	2.09	-1.72	1.44	-5.97	2.03	**-0.43**
chair (11.21%)	-4.45		-0.08	-5.62	4.25	**-1.18**
cow (2.74%)	22.73	-4.23		1.21	-0.50	**3.84**
dining table (5.12%)	-0.70	-5.38	-4.95	11.37	2.97	**0.66**
dog (8.66%)	-7.23	-0.13	5.95	-1.22	2.69	**0.01**
horse (5.75%)	-1.87	2.40	-7.41	18.67	9.36	**4.23**
motorbike (4.84%)	8.57	-2.92	-7.76	0.00	5.21	**0.62**
person (42.08%)	1.03	0.52	1.08	0.46	1.02	**0.82**
potted plant (5.29%)	1.57	0.54	4.41		4.53	**2.21**
sheep (1.96%)	9.82	10.77	26.98	5.55	3.95	**11.41**
sofa (7.30%)	-2.86	-0.92		-0.86	1.84	**-0.56**
train (5.24%)	4.26		-6.53	35.48	0.00	**6.64**
tv/monitor (5.36%)	-1.06	-2.97	-14.29	2.44	-0.56	**-3.29**
	3.39	**-0.29**	**0.07**	**2.98**	**1.58**	**1.55**

the Edge Histogram features, there is a dramatic improvement. The overall improvement is 1.55%. If we select the best single feature for each class the mean average precision increases from 0.2358 to 0.2402, i.e. a 1.86% increase.

It must be emphasized that the parameters of the methods were optimized in the training set, which is separate from the test set. This means that the results should indeed give a realistic indication of the generalization ability of the two different methods. If we optimized the performance directly with the test set, we would get an even more significant performance increase, but this scenario is not realistic as the parameters can easily "overlearn" some features of the dataset and thus not be generally applicable.

5 Conclusions

We have proposed a class density estimation method that takes into account the nearest SOM units of projected data vectors both in the feature space and in the SOM grid. In the baseline approach previously used in the PicSOM system the value field on the SOM grid was convolved after projecting an object class to its best-matching units. This is now preceeded by a convolution in the "BMU domain", i.e. in the set of nearest SOM units in the original feature space.

The distribution formed on the SOM surface can be seen as a two-dimensional discrete probability density, and can be used to find unannotated objects which are similar to the modeled class. We have demonstrated that the proposed approach can improve the accuracy when using the PicSOM technique to retrieve objects belonging to the same semantic class in an image database. However, the approach can be more generally applied to any kind of retrieval scenario.

The initial results presented in this paper are promising, however not as conclusive as we had hoped. There is no satisfactory general rule of picking the optimal BMU depths for different class and feature combinations. It thus remains as an open research question what properties of the semantic class and the feature extraction method could explain the optimal value of the $k_{\max}$ parameter.

References

1. Hauptmann, A.G., Christel, M.G., Yan, R.: Video retrieval based on semantic concepts. Proceedings of the IEEE 96(4), 602–622 (2008)
2. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer Series in Information Sciences, vol. 30. Springer, Berlin (2001)
3. Laaksonen, J., Koskela, M., Oja, E.: Class distributions on SOM surfaces for feature extraction and object retrieval. Neural Networks 17(8-9), 1121–1133 (2004)
4. Laaksonen, J., Koskela, M., Oja, E.: PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing 13(4), 841–853 (2002)
5. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive text document collection. IEEE Transactions on Neural Networks 11(3), 574–585 (2000)
6. Pampalk, E., Rauber, A., Merkl, D.: Using smoothed data histograms for cluster visualization in self-organizing maps. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 871–876. Springer, Heidelberg (2002)
7. Koskela, M., Laaksonen, J., Oja, E.: Implementing relevance feedback as convolutions of local neighborhoods on self-organizing maps. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 981–986. Springer, Heidelberg (2002)
8. Koikkalainen, P.: Progress with the tree-structured self-organizing map. In: 11th European Conference on Artificial Intelligence, European Committee for Artificial Intelligence (ECCAI) (August 1994)
9. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluation of color descriptors for object and scene recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA (June 2008)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
11. ISO/IEC: Information technology - Multimedia content description interface - Part 3: Visual, 15938-3:2002(E) (2002)
12. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)