Self-Organizing Maps, Vector Quantization, and Mixture Modeling

Tom Heskes

***Abstract*—Self-organizing maps are popular algorithms for unsupervised learning and data visualization. Exploiting the link between vector quantization and mixture modeling, we derive expectation–maximization (EM) algorithms for self-organizing maps with and without missing values. We compare self-organizing maps with the elastic-net approach and explain why the former is better suited for the visualization of high-dimensional data. Several extensions and improvements are discussed. As an illustration we apply a self-organizing map based on a multinomial distribution to market basket analysis.**

***Index Terms*—Expectation–maximization (EM) algorithms, market basket analysis, missing values, mixture modeling, self-organizing maps, vector quantization.**

I. Introduction

SELF-ORGANIZING maps are popular tools for clustering and visualization of high-dimensional data [1], [2]. The well-known Kohonen learning algorithm can be interpreted as a variant of vector quantization with additional lateral interactions [3], [4]. The addition of lateral interaction between units introduces a sense of topology, such that neighboring units represent inputs that are close together in input space [5]. Self-organizing maps are therefore also referred to as topology-preserving maps.

Self-organizing maps without lateral interaction are standard vector quantizers. Vector quantization, especially in the "soft" annealed variants [6]–[9] is very similar to mixture modeling. In fact, although they seem to pursue a different goal, algorithms for vector quantization and mixture modeling in some circumstances end up being exactly the same.

In this article, we will explore the links between self-organizing maps, vector quantization, and mixture modeling in order to derive new interpretations and algorithms. We will start in Section II with the vector quantization interpretation of self-organizing maps. Using the free energy interpretation of [10], we show in Section III that the so-called batch-map algorithm of [1], [11] is in fact a standard expectation maximization (EM) algorithm. Adapting the mixture modeling interpretation of vector quantization, we will see in Section IV that we can interpret self-organizing maps as mixture models with additional regularization. This interpretation is used in Section IV-B to compare self-organizing maps with elastic nets, which are explicitly defined as mixture models with regularization. The mixture modeling interpretation naturally leads to EM algorithms for self-organizing maps in the case of missing values in Section V. The mixture model for the standard Kohonen algorithm is a mixture of Gaussians. In some situations, other probability distributions (and thus quantization errors) may be more appropriate. As an example, we will in Section VI apply self-organizing maps to market basket analysis, where we use a quantization error derived from a multinomial distribution.

Manuscript received November 20, 2000; revised April 4, 2001.

The author is with RWCP Theoretical Foundation SNN, University of Nijmegen, Nijmegen 6252 EZ, The Netherlands (e-mail: tom@mbfys.kun.nl).

Publisher Item Identifier S 1045-9227(01)05543-6.

II. Self-Organizing Maps and Vector Quantization

A. Quantization Errors

To derive an error function for the self-organizing map, we will follow the vector quantization interpretation given in, among others, [3].

A self-organizing map consists of a set of nodes r with corresponding weight vectors $\mathbf{w}_r$. The quantization error of the node with weight $\mathbf{w}_r$ given a particular input $\mathbf{x}^\mu$ reads

$$D(\mathbf{x}^\mu, \mathbf{w}_r) = \tfrac{1}{2}\|\mathbf{x}^\mu - \mathbf{w}_r\|^2.$$

Given a set of inputs $\mathcal{X}$ and weights $\mathcal{W}$, let p_r^μ denote the probability that input $\mathbf{x}^\mu$ is assigned to the node with weight $\mathbf{w}_r$. It is constrained by $\sum_r p_r^\mu = 1$ and $p_r^\mu \geq 0$. Even if we assign input μ to node r, there is a confusion probability h_{rs} that input μ is instead quantized by the weight vector $\mathbf{w}_s$ corresponding to node s. h_{rs} corresponds to the lateral-interaction strength and defines the underlying manifold: usually it is a decreasing function of the distance between nodes r and s on a two-dimensional grid. Given the data $\mathcal{X}$, the goal is now to find the probability assignments $\mathcal{P}$ and weights $\mathcal{W}$ minimizing the error

$$F_{\text{quantization}}(\mathcal{P}, \mathcal{W}) = \sum_\mu \sum_r p_r^\mu \sum_s h_{rs} D(\mathbf{x}^\mu, \mathbf{w}_s).$$

B. The Free Energy

An annealed variant of the self-organizing map is obtained if we add an entropy term of the form

$$F_{\text{entropy}}(\mathcal{P}) = \sum_\mu \sum_r p_r^\mu \log\left[\frac{p_r^\mu}{q_r}\right]$$

where q_r can be interpreted as prior probability assignments. The usual choice is $q_r = 1/K$ with K the number of nodes, but for later purposes we will consider here the general situation. This entropy term favors probability assignments that are similar to q_r, i.e., maximize the entropy for homogeneous q_r. Annealed vector quantization has been introduced in [6] and applied to self-organizing maps in, e.g., [12]–[14].

The final "free energy functional" now follows from a weighted combination of the quantization and entropy term

$$F(\mathcal{P}, \mathcal{W}) = \beta F_{\text{quantization}}(\mathcal{P}, \mathcal{W}) + F_{\text{entropy}}(\mathcal{P}). \quad (1)$$

β plays the role of an inverse temperature: the larger β, the smaller the influence of the entropy term. Formulation of the optimization criterion in terms of this free energy functional will be very convenient in the derivation of EM algorithms later on.

C. The Standard Error Function

The dependency on the assignments $\mathcal{P}$ can be removed by computing the optimal assignments $\mathcal{P}(\mathcal{W})$ given a particular set of weights $\mathcal{W}$

$$p_r^\mu(\mathcal{W}) = \frac{q_r \exp\left[-\beta \sum_t h_{rt} D(\mathbf{x}^\mu, \mathbf{w}_t)\right]}{\sum_s q_s \exp\left[-\beta \sum_t h_{st} D(\mathbf{x}^\mu, \mathbf{w}_t)\right]}. \quad (2)$$

With $D(\mathbf{x}, \mathbf{w})$ continuous in $\mathbf{w}$, these assignments are unique. Substitution into (1) then yields

$$E(\mathcal{W}) \equiv \min_{\mathcal{P}} F(\mathcal{P}, \mathcal{W}) = -\sum_\mu \log \sum_r q_r \exp\left[-\beta \sum_t h_{rt} D(\mathbf{x}^\mu, \mathbf{w}_t)\right]. \quad (3)$$

This error function (with $q_r = 1/K$) corresponds to an annealed version of a closely related variant of Kohonen's original self-organizing map algorithm [5]. The (small) differences are discussed in [12] and [13] and mainly have to do with a slightly different choice for the winning unit (see also below).

It can be shown (e.g., following a proof in [10]) that any (locally) optimal solution of the free energy $F(\mathcal{W}, \mathcal{P})$ in terms of both $\mathcal{W}$ and $\mathcal{P}$ corresponds to a (locally) optimal solution of the error function $E(\mathcal{W})$ in terms of only $E(\mathcal{W})$ in terms of only $\mathcal{W}$ and vice versa. In other words, we can exchange the two optimization criteria, as we will do throughout the article.

III. EM Algorithm Without Missing Values

A. Expectation and Maximization

The free energy functional (1) allows for an extremely straightforward derivation of an EM algorithm. Both the expectation and the maximization step can be seen as minimizing this same functional [10].

The *expectation step* in the full EM algorithm follows by minimizing $F(\mathcal{P}, \mathcal{W})$ with respect to the assignments $\mathcal{P}$, given the current set of parameters $\mathcal{W}$. We immediately obtain (2).

The *maximization step* in the full EM algorithm follows by minimizing $F(\mathcal{P}, \mathcal{W})$ with respect to the parameters $\mathcal{W}$, given the current set of assignments $\mathcal{P}$. For sum-squared $D(\mathbf{x}, \mathbf{w})$, we easily find

$$\mathbf{w}_s(\mathcal{P}) = \frac{\sum_\mu \sum_r p_r^\mu h_{rs} \mathbf{x}^\mu}{\sum_\mu \sum_r p_r^\mu h_{rs}}. \quad (4)$$

B. The Batch-Map Algorithm

This EM algorithm (in the limit $\beta \to \infty$) is referred to as the batch-map algorithm in [1] and [11]. The batch-map algorithm corresponding to Kohonen's original learning (see, e.g., [1]) differs from the EM algorithm discussed here in two aspects: it corresponds to the limit $\beta \to \infty$ and has no neighbor averaging in the E-step. The limit $\beta \to \infty$ is just a special case of the analysis in this article and contains no further peculiarities except that some of the proofs may be technically more involved because of discontinuities. The simpler E-step can be interpreted as an approximation to the one derived in (2) which does involve neighbor averaging. The simpler E-step is faster, but the connection with a global error function like (3) is lost. This makes it difficult to check and proof convergence. Maps obtained through application of both winner mechanisms are roughly the same (for example, tested on WEBSOM [15], Prof. Kohonen, private communication). The constraint $\sum_s h_{rs} = 1$ facilitates a probabilistic interpretation of the lateral interactions, but has further no consequences.

C. Simple Speed-Ups

The EM algorithms presented here are the standard versions. There are many different ways to speed them up. Especially attractive and relatively simple is the "accelerated" version. The idea is to take the new weight vectors w_r^{new} "beyond" the optimal $w_r(\mathcal{P})$ given in (4)

$$w_r^{\text{new}} = \eta w_r(\mathcal{P}) + (1 - \eta) w_r^{\text{old}}$$

with $1 \leq \eta < 2$. The same can be done for the probabilities $\mathcal{P}$ in the E-step. Here we might take, for all μ

$$\log p_r^{\text{new}} \propto \eta \log p_r(\mathcal{W}) + (1 - \eta) \log p_r^{\text{old}}$$

where the proportionality constant follows from the normalization $\sum_r p_r^{\text{new}} = 1$. This logarithmic averaging seems to work a little better than simple linear averaging and explicitly constrains the probabilities to positive numbers. By applying the same reasoning as in [16], it can be shown that accelerated EM is locally contractive (converges to a local minimum if starting sufficiently close to this minimum) for $\eta < 2$. In practice, $\eta \approx 1.3$ seems to work fine and speeds up the convergence of the EM algorithm considerably (roughly a factor of two).

IV. A Mixture-Modeling Interpretation

A. Self-Organizing Maps are Regularized Mixture Models

There is a close link between vector quantization and mixture modeling. Saying that a particular $\mathbf{w}_r$ is a good quantizer for a pattern $\mathbf{x}^\mu$ because of a low quantization error $D(\mathbf{x}^\mu, \mathbf{w}_r)$ is similar to stating that some probability $G(\mathbf{x}^\mu | \mathbf{w}_r)$ of finding $\mathbf{x}^\mu$ given $\mathbf{w}_r$ is quite high. The obvious choice for this probability in the case of a sum-squared error $D(\mathbf{x}, \mathbf{w})$ is a Gaussian.

Let us first consider the case of no lateral interaction, i.e., $h_{rs} = \delta_{rs}$. As a mixture model we take

$$P(\mathbf{x}|\mathcal{W}) = \sum_r q_r G(\mathbf{x}|\mathbf{w}_r)$$

with

$$G(x|w) = \sqrt{\frac{\beta}{2\pi}} \mathrm{e}^{-\beta(x-w)^2/2}$$

and

$$G(\mathbf{x}|\mathbf{w}) = \prod_{\alpha} G(x_\alpha|w_\alpha).$$

Through simple substitution it is easy to show that, with this particular choice of mixture model and up to irrelevant constants, we have

$$L(\mathcal{W}) \equiv \sum_{\mu} \log P(\mathbf{x}^\mu|\mathcal{W}) = -E(\mathcal{W}) \tag{5}$$

i.e., the optimization criterion for annealed vector quantization corresponds to a maximum likelihood procedure for a mixture of Gaussians.

With lateral interaction, the link is not so obvious. To simplify the term in the exponent in (3), we need the "bias-variance decomposition"

$$\sum_s h_{rs} D(x, w_s) = D(x, \tilde{w}_r) + \sum_s h_{rs} D(\tilde{w}_r, w_s) \tag{6}$$

with

$$\tilde{w}_r = \sum_s h_{rs} w_s.$$

The essence here is that the average error on the righthand side can be decomposed into an error of an average weight $\tilde{w}_r$ and a variance term *independent* of the input x. The variance $V_r(\mathcal{W}) = \sum_s h_{rs} D(\tilde{\mathbf{w}}_r, \mathbf{w}_s)$ measures to what extent the weights vary around node r.

The decomposition is used to proof that self-organizing maps can be interpreted as mixture models with an added regularization term. If we take the mixture model

$$P(\mathbf{x}|\mathcal{W}) = \sum_r \tilde{q}_r(\mathcal{W}) G(\mathbf{x}|\tilde{\mathbf{w}}_r) \tag{7}$$

with

$$\tilde{q}_r(\mathcal{W}) \equiv \frac{q_r \mathrm{e}^{-\beta V_r(\mathcal{W})}}{\sum_s q_s \mathrm{e}^{-\beta V_s(\mathcal{W})}}$$

and compute the loglikelihood, we obtain, after some rewriting, the self-organizing map error (3), except for a term independent of the patterns $\mathcal{X}$. That is, neglecting irrelevant constants independent of $\mathcal{W}$, we have

$$E(\mathcal{W}) = -L(\mathcal{W}) + E_{\text{regularization}}(\mathcal{W}) \tag{8}$$

with $L(\mathcal{W})$ the loglikelihood as in (5) and the regularization term

$$E_{\text{regularization}}(\mathcal{W}) \equiv -\sum_{\mu} \log \sum_r q_r \mathrm{e}^{-\beta V_r(\mathcal{W})}. \tag{9}$$

B. Self-Organizing Maps and Elastic Nets

The self-organizing map error (8) has a striking correspondence with the error function for elastic nets. In the elastic-net approach [17], [18], topology is introduced by adding a penalty term to an (annealed) vector quantization error, i.e., the goal is to minimize an error function of the form

$$E_{\text{elastic}}(\mathcal{W}) = -\sum_{\mu} \log \sum_r q_r \exp\left[-\beta D(\mathbf{x}^\mu, \tilde{\mathbf{w}}_r)\right] + \sum_{r,s} h_{rs} \left\|\tilde{\mathbf{w}}_r - \tilde{\mathbf{w}}_s\right\|^2. \tag{10}$$

The standard choice for elastic nets is again $q_r = 1/K$.

From an efficiency point of view, the EM algorithm for self-organizing maps derived in Section III is the clear winner: incorporation of tolopology only requires extra summations over nodes, limited to the width of the lateral interaction h_{rs}. On the other hand, the additive penalty term in the elastic-net approach [see (10)] makes that the M-step requires the solution of a set of K linear equations [18].

From a modeling point of view the differences are more subtle. The important difference between the two is that fixed $q_r = 1/K$ in (10), the standard choice in the elastic-net approach, corresponds to fixed marginals $P(r|\mathcal{W})$. This yields a tendency to make all nodes equally important. In the self-organizing map approach, with $q_r = 1/K$ in (3), one can still have nodes with a low marginal $P(r|\mathcal{W})$, namely those with a high local variance $V_r(\mathcal{W})$. These variances are similar to what in the literature on self-organizing maps is called the "U-matrix" [2]. The U-matrix is often visualized as a surface on the two-dimensional topology of the self-organizing map and indicates clusters, with different clusters separated by barriers. In mixture-modeling terms these barriers correspond to nodes with a low marginal $P(r|\mathcal{W})$, which can focus on interpolating between different clusters (see also the application in Section VI). An elastic-net algorithm does not have this flexibility and therefore seems less suited for the visualization of high-dimensional data.

The regularization term (9) aims at low variances, that is, small differences between weight vectors of neighboring nodes. This is the term that explains the self-organizing property of self-organizing maps: it implements the tendency for neighboring nodes to represent similar input patterns. Note that the regularization term scales with the number of patterns $N = \sum_\mu$. It can, therefore, not be truly interpreted as resulting from a kind of Bayesian prior as, e.g., in the model described in [19], since such a term would become less and less important with growing N.

V. EM Algorithm With Missing Values

A. The Derivation

The basic idea in EM algorithms for mixture models is to extend the distribution $P(\mathbf{x}|\mathcal{W})$ to a joint distribution $P(\mathbf{x}, r|\mathcal{W})$, where the states of the nodes r are considered hidden. The extra set of parameters $\mathcal{P}$ is introduced to represent the probabilities of these states. Another important application of EM is to learning with truly missing (input) values. The combination of

both missing inputs and mixture models is pursued in [20]. The probabilistic interpretation of self-organizing maps derived in this paper allows for a similar combination. We consider the standard situation $q_r = 1/K$.

We assume that for each pattern μ some inputs are known, indicated by the lower index k, and some may be missing, indicated by m. We should in fact write k^μ and m^μ, but for the sake of clarity we will leave it at k and m. From the definition of the error $E(\mathcal{W})$ in vector-quantization terms, as in (3), it is not so obvious how to incorporate missing values. The link (8), where the vector-quantization error is decomposed in a loglikelihood term $-L(\mathcal{W})$ and a regularization term $E_{\text{regularization}}(\mathcal{W})$, offers a solution. The regularization term is independent of the data $\mathcal{X}$ and thus unaffected by the presence of missing values. The loglikelihood term, on the other hand, can only look at the known components and thus becomes

$$L(\mathcal{W}) = \sum_\mu \log P(\mathbf{x}_k^\mu|\mathcal{W})$$

with

$$P(\mathbf{x}_k|\mathcal{W}) = \int dx_m P(\mathbf{x}|\mathcal{W})$$

and $P(\mathbf{x}|\mathcal{W})$ from (7). Following [10] and similar to the above link between the error (3) and the free energy (1), the free energy functional corresponding to the error $E(\mathcal{W}) = -L(\mathcal{W}) + E_{\text{regularization}}(\mathcal{W})$ can be written

$$\begin{aligned} F(\mathcal{P}, \mathcal{W}) = &-\sum_r \int dx_m p_r^\mu(\mathbf{x}_m) \log P(\mathbf{x}_k^\mu, \mathbf{x}_m|\tilde{\mathbf{w}}_r) \\ &+ \sum_r \int dx_m p_r^\mu(\mathbf{x}_m) \log p_r^\mu(\mathbf{x}_m) \\ &+ E_{\text{regularization}}(\mathcal{W}) \end{aligned} \tag{11}$$

with for each μ a joint distribution $p_r^\mu(\mathbf{x}_m)$ over both the state of the nodes and the missing inputs.

The E-step follows by minimizing free energy (11) with respect to these distributions for given $\mathcal{W}$. We easily derive

$$p_r^\mu(\mathbf{x}_m|\mathcal{W}) = p_r^\mu(\mathcal{W}) G(\mathbf{x}_m|\tilde{\mathbf{w}}_{rm}) \tag{12}$$

where we have defined

$$p_r^\mu(\mathcal{W}) = \frac{q_r \exp\left[-\beta \sum_t h_{rt} D(\hat{\mathbf{x}}_r^\mu, \mathbf{w}_t)\right]}{\sum_s q_s \exp\left[-\beta \sum_t h_{st} D(\hat{\mathbf{x}}_s^\mu, \mathbf{w}_t)\right]} \tag{13}$$

with

$$\hat{x}_{r\alpha}^\mu = \begin{cases} x_\alpha^\mu, & \text{if } \alpha \text{ known for } \mu \\ \tilde{w}_{r\alpha}, & \text{if } \alpha \text{ missing in } \mu. \end{cases} \tag{14}$$

In other words, the E-step in the case of missing values yields (12) with $G(\mathbf{x}_m|\tilde{\mathbf{w}}_{rm})$ a Gaussian probability distribution over the missing inputs given the current average weight $\tilde{\mathbf{w}}_{rm}$, and $p_r^\mu(\mathcal{W})$ equivalent to (2) for the case of no missing values with missing $\mathbf{x}_m^\mu$ replaced by $\tilde{\mathbf{w}}_r$.

The M-step is based on the minimization of the free energy (11) with respect to the parameters $\mathcal{W}$ for fixed $\mathcal{P}$. After straightforward manipulations we obtain

$$\mathbf{w}_s(\mathcal{P}) = \frac{\sum_\mu \sum_r p_r^\mu h_{rs} \hat{\mathbf{x}}_r^\mu}{\sum_\mu \sum_r p_r^\mu h_{rs}} \tag{15}$$

with $\hat{x}_{r\alpha}^\mu$ as in (14) and p_r^μ from (13), i.e., the term $G(\mathbf{x}_m|\tilde{\mathbf{w}}_{rm})$ drops out. The M-step with missing values is equivalent to the M-step (4) without missing values, using the same substitution as in the E-step for the missing $\mathbf{x}_m^\mu$. This is what we might have expected from the start, except that it is important to realize that the parameter used for filling in the unknown input x_α is the average $\tilde{w}_{r\alpha}$, and not the original $w_{r\alpha}$. We can use the procedures described in Section III-C to accelerate the algorithms.

B. An Illustration

We illustrate the use of an EM algorithm for self-organizing maps with missing values on a simple toy problem. The training set consists of 500 points in three dimensions, all close to the plane $y = z$, but with 50% of all values missing (not shown). The self-organizing map has 8×12 nodes with lateral interactions $h_{rs} \propto \exp[-d_{rs}/2\sigma^2]$ where d_{rs} refers to the node distance on a two-dimensional grid. The parameters are $\sigma = 0.5$ and $\beta = 100$. As can be seen in Fig. 1 the map still manages to unfold and represents the data quite well.

VI. Other Probability Models

A. The Exponential Family

The analysis presented in this paper focused on sum-squared error $D(x, w)$ and corresponding Gaussian probability $G(x|w)$. It can be easily extended to quantization errors that can be derived from probability distributions in the exponential family. For distributions of the form

$$G(x|w) = \exp\left[c(w)T(x) + d(w) + S(x)\right]$$

the quantization error is the deviance

$$\begin{aligned} D(x, w) &= -\log G(x|w) + \log G(x|x) \\ &= [c(x) - c(w)]T(x) + d(x) - d(w). \end{aligned}$$

$c(w)$ is called the canonical link, $T(x)$ the sufficient statistic. Examples include the Gamma distribution, multinomial, and Poisson. The bias-variance decomposition (6), and thus the correspondence of self-organizing maps to regularized mixture modeling, still holds if we define the average weight by averaging the canonical links (see, e.g., [21])

$$c(\tilde{w}_r) = \sum_s h_{rs} c(w_s).$$

Furthermore, if the sufficient statistic is linear, i.e., $T(x) = x$ as for most common distributions, we still have the simple form (4) for the M-step. The EM algorithm for missing values stays the

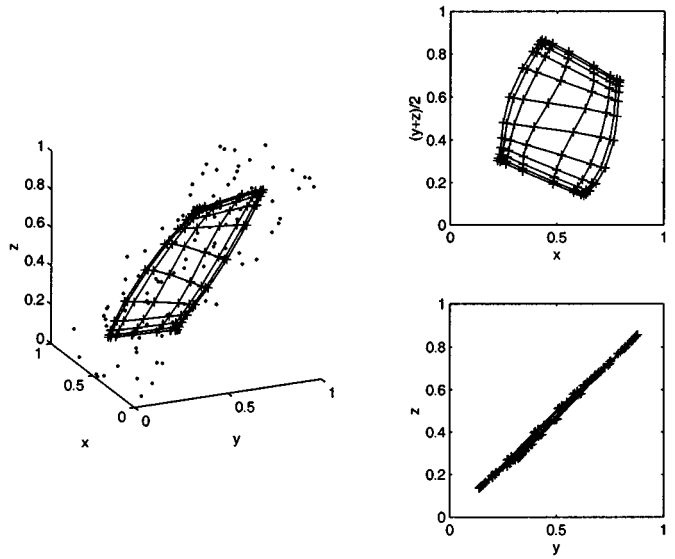

Fig. 1. Self-organizing map learned on data with missing values. The data set and the unfolded map are shown on the left-hand side. Fifty percent of all values are missing (not shown). Projections on the plane $y = x$ and along the x-direction on the right-hand side.

same for continuous distributions where the annealing parameter β can be interpreted as a dispersion parameter and for all other distributions if $\beta = 1$. There hardly seems to be a reason to restrict self-organizing maps to sum-squared errors. Depending on the format of the data and the underlying assumptions, more appropriate quantization errors can be chosen, perhaps even different ones for different dimensions.

B. An Example: Market Basket Analysis

In market basket analysis, we are given a whole list of transactions. Each transaction corresponds to a joint set of products purchased by a customer at the same time. Our goal here[1] is to use the information in the transactions to map the products onto a two-dimensional sheet such that neighboring products are "similar." The essence is to define a proper measure of similarity. We follow reasoning similar to the distributional clustering of words based on their (joint) occurrence in documents (see, e.g., [24] and [25]). The basic idea is that similar products have similar conditional probabilities of buying other products.

Let us consider the "base" product i, which is purchased in n_i of the transactions. We define the vector $\mathbf{w}$ of conditional probabilities with components w_j. If we use $\mathbf{w}$ to model product i, w_j refers to the conditional probability that the client buys product j given that he or she buys product i. Given w_j and n_i, the probability to find n_{ij} joint purchases of both products i and j is the multinomial

$$P(n_{ij}|n_i, w_j) = \binom{n_i}{n_{ij}} w_j^{n_{ij}} (1 - w_j)^{n_i - n_{ij}}.$$

Defining $x_{ij} = n_{ij}/n_i$ and with some rewriting, it is easy to see that this multinomial is a member of the exponential family. Applying the procedure explained in the previous section to go from a probability distribution to a distance measure, we obtain

$$D(\mathbf{w}, \mathbf{x}_i|n_i) = n_i \sum_{j \neq i} x_{ij} \log \left[\frac{x_{ij}}{w_j}\right] + (1 - x_{ij}) \log \left[\frac{1 - x_{ij}}{1 - w_j}\right].$$

This is a kind of cross-entropy error, in which $x_{ij} = n_{ij}/n_i$ plays the role of the observed relative frequency, which we try to model with the conditional probability w_j. The distance is weighted by the number of occurrences n_i, which has the effect that products that are bought more frequently will have a stronger effect on the formation of the self-organizing map. Note that in the above notation product i plays the role of an example (denoted μ beforehand).

C. A Supermarket Product Map

We applied the self-organizing map trained with EM on a set of supermarket basket data concerning 193 639 transactions. All products are summarized in 199 product groups, i.e., all information needed to build the self-organizing map is contained in the 199×199 matrix with relative frequencies x_{ij} and the 199 dimensional vector n_i. We used a two-dimensional grid of 60 by 40 nodes with interactions of the form $\exp\left[-d_{rs}/2\sigma^2\right]$ for $\sigma = 1$, annealing parameter $\beta = 0.005$, and $\eta = 1.3$ to accelerate the EM algorithm.

The resulting map is visualized in Fig. 2. The lines connect the different nodes. Since the number of nodes is much higher than the number of products to cluster, this is a typical example of a so-called "emergent" self-organizing map [2]. The height is proportional to the logarithm of the probability $P(r|\mathcal{W})$ for each node r. The square markers indicate the "winning" nodes with the highest probability $P(r|\mathcal{W}, \mathbf{x}_i)$ for some product i. The names of the corresponding products are written below the markers.[2] Winning nodes are within valleys, surrounded by hills with hardly any probability mass. Although not perfect, it can be seen that products one would expect to be similar are indeed grouped together. See, for example, the large cluster of household products.

VII. Discussion

In this article we have explored the links between self-organizing maps, vector quantization, and mixture modeling. We derived EM algorithms with and without missing values, of which the batch-map algorithm [11] is a special case, together with simple ways to accelerate them. Furthermore, we generalized the self-organizing maps to different errors and corresponding probability distributions. The other way around, we derived a general procedure to add topology to mixture models. The specific market basket application is an example of clustering contextual information. In the standard approach, see, e.g., the work on semantic maps in [26], counts or histograms are treated as or

[1]This is somewhat different from the standard goal in market basket analysis to find informative rules of the form "if a custom buys product A, he or she will buy B with probability p." See, e.g., [22] and [23].

[2]To avoid text overlap, the names of the products are sometimes well below the marker they belong to. The position of the names should therefore be only taken crudely.

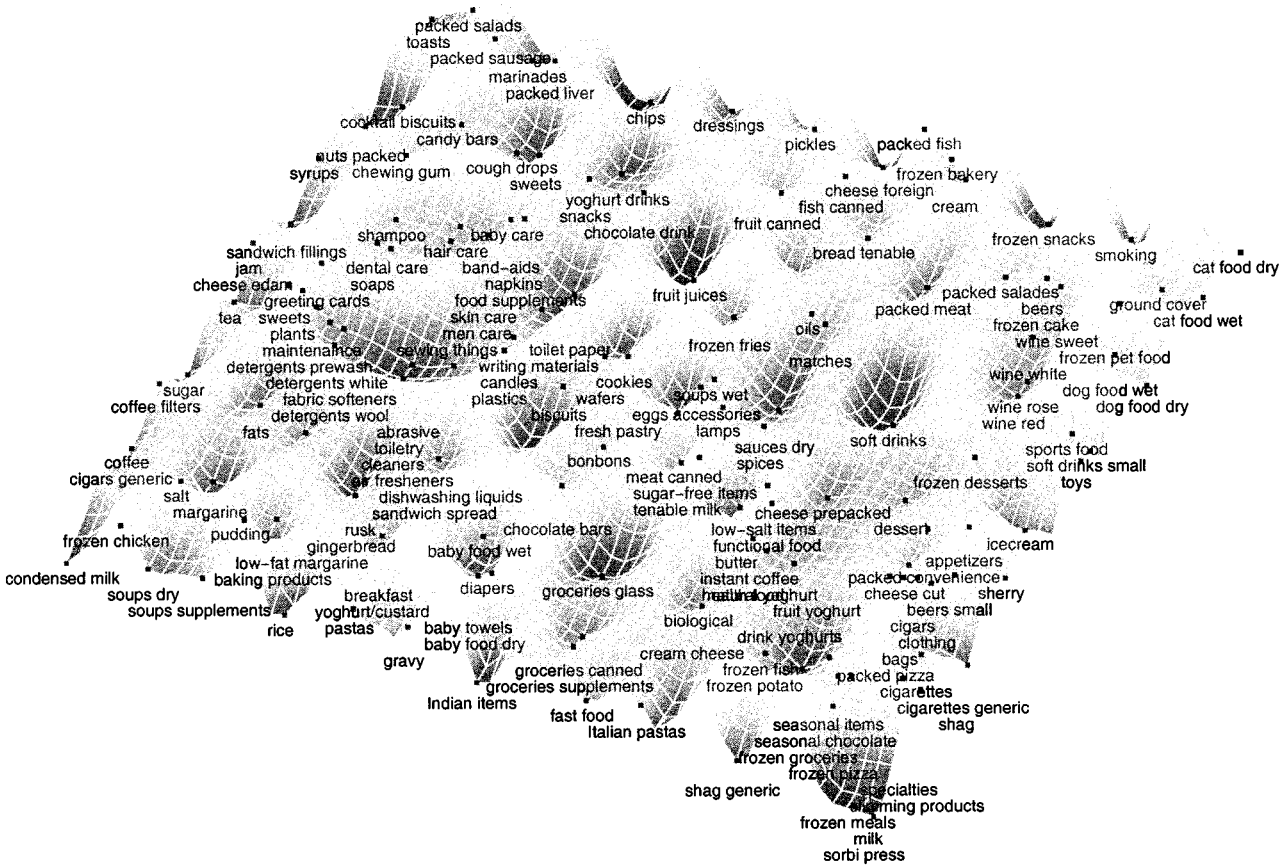

Fig. 2. Self-organizing map based on market basket data. 199 product groups are clustered based on their co-occurrence frequencies with other products. Lines link the nodes on a two-dimensional grid. Heights visualize the marginal probabilities of the nodes. Markers indicate winning nodes, with the corresponding product names written below them.

transformed into vectors in an Euclidean space to fit the description of the standard "sum-squared" self-organizing map. The approach taken here seems more appropriate from a statistical point of view. The only price we have to pay is the computation of logarithms and exponentials; all other algorithmic aspects are largely the same. Other options, mainly focusing on speed-ups for massive databases, are discussed in [15].

The self-organizing map algorithm is often considered a heuristic approach with severe limitations. In [27], some of these limitations are listed and contrasted with the generative topographic mapping (GTM) algorithm. The GTM algorithm builds upon a generative mixture model where the centers in data space are nonlinear functions of the position of the nodes on the topological maps, parameterized by a set of weights. Three of the six problems with self-organizing maps solved by the GTM are derived from the lack of an energy function for the original Kohonen learning rule. They disappear in the formulation chosen here, which differs from the original one only by a slight change in the definition of the winning unit [12]. Two other problems relate to the choice of the lateral interaction and how this choice affects neighborhood preservation. It is argued in the literature that in combination with annealing the final solution is rather insensitive to the exact choice of the (small and fixed) neighborhood interaction [13]. However, if neighborhood preservation is fundamental for the problem at hand, the GTM, with hard-wired rather than emergent topological preservation, may be preferable, even although the user has to specify the nonlinear functions mapping the node positions to the data space. The sixth limitation is the lack of a probability model, the basis of the GTM. In this paper, we have shown how to add self-organizing properties to a mixture model and how the resulting optimization criterion can still be interpreted in terms of probability modeling with regularization.

ACKNOWLEDGMENT

The author would like to thank W. Wiegerinck and A. Wanders for their assistance in the presentation of this work.

REFERENCES

[1] T. Kohonen, "The self-organizing map," *Neurocomput.*, vol. 21, pp. 1–6, 1998.

[2] A. Ultsch, "Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series," in *Kohonen Maps*, E. Oja and S. Kaski, Eds. Amsterdam, The Netherlands: Elsevier, 1999, pp. 33–45.

[3] S. Luttrell, "Self-organization: A derivation from first principles of a class of learning algorithms," in *Proc. Int. Joint Conf. Neural Networks*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1989, vol. 2, pp. 495–498.

[4] ——, "A Bayesian analysis of self-organizing maps," *Neural Comput.*, vol. 6, pp. 767–794, 1994.

[5] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.

[6] K. Rose, E. Gurewitz, and G. Fox, "Statistical mechanics of phase transitions in clustering," *Phys. Rev. Lett.*, vol. 65, pp. 945–948, 1990.

[7] ——, "Vector quantization by deterministic annealing," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1249–1257, 1992.
[8] E. Yair, K. Zeger, and A. Gersho, "Competitive learning and soft competition for vector quantizer design," *IEEE Trans. Signal Processing*, vol. 40, pp. 294–309, 1992.
[9] J. Buhmann and H. Kühnel, "Vector quantization with complexity costs," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1133–1145, 1993.
[10] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. Jordan, Ed. Dordrecht, The Netherlands: Kluwer, 1998, pp. 355–368.
[11] Y. Cheng, "Convergence and ordering of Kohonen's batch map," *Neural Comput.*, vol. 9, pp. 1667–1676, 1997.
[12] T. Heskes and B. Kappen, "Error potentials for self-organization," in *Proc. Int. Conf. Neural Networks*. New York: IEEE, 1993, vol. 3, pp. 1219–1223.
[13] T. Graepel, M. Burger, and K. Obermayer, "Self-organizing maps: Generalizations and new optimization techniques," *Neurocomput.*, vol. 21, pp. 173–190, 1998.
[14] T. Heskes, "Energy functions for self-organizing maps," in *Kohonen Maps*, E. Oja and S. Kaski, Eds. Amsterdam, The Netherlands: Elsevier, 1999, pp. 303–315.
[15] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Trans. Neural Networks*, vol. 11, pp. 574–585, 2000.
[16] B. Peters and H. Walker, "An iterative procedure for obtaining maximum-likelihood estimates of the parameters for a mixture of normal distributions," *SIAM J. Appl. Math.*, vol. 35, pp. 362–378, 1987.
[17] R. Durbin and D. Willshaw, "An analogue approach to the traveling salesman problem using an elastic net method," *Nature*, vol. 326, pp. 689–691, 1987.
[18] A. Yuille, P. Stolorz, and J. Utans, "Statistical physics, mixtures of distributions, and the EM algorithm," *Neural Comput.*, vol. 6, pp. 334–340, 1994.
[19] A. Utsugi, "Hyperparameter selection for self-organizing maps," *Neural Comput.*, vol. 9, pp. 623–635, 1997.
[20] Z. Ghahramani and M. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems 6*, J. Cowan, G. Tesauro, and J. Alspector, Eds. San Mateo, CA: Morgan Kaufmann, 1994, pp. 120–127.
[21] J. Hansen and T. Heskes, "General bias/variance decomposition with target independent variance of error functions derived from the exponential family of distributions," in *Proc. 15th Int. Conf. Pattern Recognition*, A. Sanfeliu, J. J. Villanueva, M. Vanrell, R. Alguézar, A. K. Jain, and J. Kittler, Eds., 2000, vol. 2, pp. 207–210.
[22] M. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York: Wiley, 1997.
[23] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations," in *Proc. ACN SIGMOD/PODS'97*, 1997, pp. 265–276.
[24] F. Periera, N. Tishby, and L. Lee, "Distributional clustering of English words," in *Proc. Assoc. Comput. Linguistics*, 1993, pp. 183–190.
[25] T. Hofmann, J. Puzicha, and M. Jordan, "Learning from dyadic data," in *Advances in Neural Information Processing Systems 11*, M. Kearns, S. Solla, and D. Cohn, Eds. Cambridge, MA: MIT Press, 1999, pp. 466–472.
[26] H. Ritter and T. Kohonen, "Self-organizing semantic maps," *Biol. Cybern.*, vol. 61, pp. 241–254, 1989.
[27] C. Bishop, M. Svensén, and C. Williams, "GTM: A principled alternative to the self-organizing map," in *Advances in Neural Information Processing Systems 9*, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 354–360.

Tom Heskes received the M.Sc. and the Ph.D. degrees in physics both from the University of Nijmegen, The Netherlands, in 1989 and 1993, respectively.

After a year postdoctoral work at the Beckman Institute, Champaign-Urbana, IL, he rejoined the Dutch Foundation for Neural Networks (SNN) in 1994. Since 1997, he has run the company SMART Research BV. His research interests include theoretical and practical aspects of neural networks and related techniques, in addition to applications.

Dr. Heskes is a member of the editorial board of *Neurocomputing* and served as a referee for many international journals on neural networks and alike.