Multiple Cause Vector Quantization

David A. Ross and Richard S. Zemel
Department of Computer Science
University of Toronto
{dross,zemel}@cs.toronto.edu

Abstract

We propose a generative model of high-dimensional data that improves on aspects of several previous models, including probabilistic latent semantic analysis and non-negative matrix factorization. Our key assumption is that the dimensions of the data can be separated into several disjoint subsets, or *multiple causes*, which take on values independently of each other. We also assume each cause is a *vector quantizer*, with a small number of discrete states. Given a set of training examples, our model learns the association of data dimensions with causes, as well as the states of each VQ. Inference and learning are carried out efficiently via variational algorithms. We present applications of this model to problems in image decomposition, collaborative filtering, and text classification.

1 Introduction

Many collections of data exhibit a common underlying structure: they consist of a number of parts or *causes*, each of which has a small number of discrete *states*. For example, in a collection of facial images, every image contains eyes, a nose, and a mouth (except under occlusion), each of which has a range of different appearances. A specific image can be described as a *composite sketch*: a selection of the appearance of each part, depending on the individual depicted.

In this paper, we describe a stochastic generative model for data of this type. This model is well-suited to decomposing images into parts, but also applies to domains such as text and collaborative filtering in which the parts correspond to latent features, each having several alternative instantiations. This representational scheme is powerful due to its combinatorial nature: while a standard clustering/VQ method containing N components can represent at most N items, if we divide the N into j-component VQs, we can represent $j^{N/j}$ items. MCVQ is also especially appropriate for high-dimensional data in which many values may be unspecified for a given input case.

2 Generative Model

In MCVQ we assume there are K causes, each of which is modeled by a vector quantizer with J states. To generate an observed data example of N dimensions, $X \in \Re^N$, we stochastically select one state for each VQ, and one VQ for each dimension. These selections are represented as binary latent variables, $S = \{s_{jk} : j \in k\}, R = \{r_{ik}\}$, for

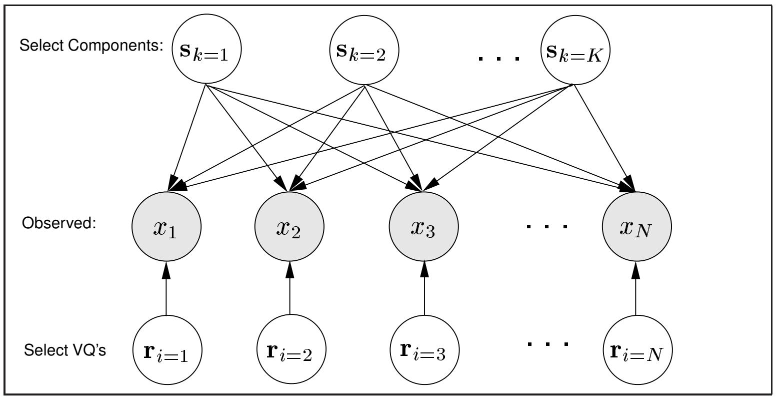

Figure 1: Graphical model representation of MCVQ. We let $\mathbf{r}_{i=1}$ represent all the variables $r_{i=1,k}$, which together select a VQ for x_1. Similarly, $\mathbf{s}_{k=1}$ represents all $s_{j,k=1}$, which together select a component of VQ 1.

$i = 1...N, k = 1...K$, and $j = 1...J$. The variable $s_{jk} = 1$ if and only if state j has been selected from VQ k. Similarly $r_{ik} = 1$ when VQ k has been selected for data dimension i. These variables can be described equivalently as multinomials, $\mathbf{s}_k \in 1...J, \mathbf{r}_i \in 1...K$.

The latent factors are sampled independently from each vector $\mathbf{s}_k$, according to its prior probability $\mathbf{a}_k$, and and the latent causes of each pixel are sampled independently from each vector $\mathbf{r}_i$, according to its prior probability $\mathbf{b}_i$. Given an instantiation of the selection variables, for each dimension i, a single cause from a single VQ determines the value of x_i. The K states, one per VQ, represent the multiple causes of the input vector.

Assuming each VQ state specifies the mean as well as the standard deviation of a Gaussian distribution, and the noise in the input dimensions is conditionally independent, we have:

$$\mathrm{P}(X|R, S, \theta) = \prod_i \prod_{k,j \in k} \mathcal{N}(x_i\,;\, \mu_{ijk},\, \sigma_{ijk})^{r_{ik}\, s_{jk}} \tag{1}$$

where $\theta = \{\mu_{ijk}, \sigma_{ijk}\}$. The graphical model representation of MCVQ is given in Fig. 1.

The resulting model can be thought of as a two-dimensional mixture model, in which $J * K$ possible causes exist for each input element. The selections of causes for the different input elements are joined along the J dimension and occur independently along the K dimension.

3 Learning and Inference

The joint distribution over the observed vector X and the latent variables is

$$\mathrm{P}(X, R, S|\theta) = \mathrm{P}(R|\theta)\mathrm{P}(S|\theta)\mathrm{P}(X|R, S, \theta) = \prod_{i,k,j \in k} a_{ik}^{r_{ik}} b_{jk}^{s_{jk}} \mathcal{N}(x_i\,;\,\theta)^{r_{ik} s_{jk}} \tag{2}$$

Given an input X, the posterior distribution over the latent variables is quite complicated, as they all become dependent.

We apply a variational EM algorithm to learn the parameters θ, and infer hidden variables given observations. We approximate the posterior distribution using a factored distribution:

$$Q(R, S|X, \theta) = \prod_{i,k} g_{ik}^{r_{ik}} \prod_{k,j \in k} m_{jk}^{s_{jk}}$$

The variational free energy, $\mathcal{F}(Q,\theta) = E_Q\big[-\log P(X,R,S|\theta) + \log Q(R,S|X,\theta)\big]$ is:

$$
\begin{aligned}
\mathcal{F}(Q,\theta) &= E_Q\big[\sum_{i,k} r_{ik}\log(g_{ik}/a_{jk}) + \sum_{k,j\in k} s_{jk}\log(m_{jk}/b_{jk}) + \sum_{i,k,j} r_{ik}s_{jk}\log\mathcal{N}(x_i\,;\theta)\big] \\
&= \sum_{k,j\in k} m_{jk}\log m_{jk} + \sum_{i,k} g_{ik}\log g_{ik} + \sum_{i,k,j} g_{ik}m_{jk}d_{ijk} \qquad (3)
\end{aligned}
$$

where $d_{ijk} = \log\sigma_{ijk} + \frac{(x_i-\mu_{ijk})^2}{2\sigma^2_{ijk}}$, and we have assumed uniform priors for the selection variables. The negative of the free energy $-\mathcal{F}$ is a lower bound on the log likelihood of generating the observations. The variational EM algorithm improves this bound by iteratively improving $-\mathcal{F}$ with respect to Q (E-step) and to θ (M-step).

Let C be the set of training cases, and Q^c be the mean field approximation to the posterior distribution over hidden states given the training case (observation) $c \in C$. We further constrain this variational approach, forcing the $\{g^c_{ik}\}$ to be consistent for any observation X^c. Hence these gating variables, which govern the selection of a cause for a given observation dimension, are not dependent on the observation. This approach encourages the model to learn representations that conform to this constraint. That is, if there are several posterior distributions consistent with an observed data vector, it favors distributions over $\{\mathbf{r}_i\}$ that are consistent with other observed data vectors. Under this formulation, only the $\{m^c_{jk}\}$ parameters are updated during the E step for each observation c:

$$m^c_{jk} = \exp\Big(-\sum_i g_{ik}d^c_{ijk}\Big)/\sum_{\alpha=1}^{J}\exp\Big(-\sum_i g_{ik}d^c_{i\alpha k}\Big)$$

The M step updates the parameters, μ and σ, from each hidden state jk to each input dimension i, and the gating variables $\{g_{ik}\}$:

$$g_{ik} = \exp\Big(-\sum_{c,j} m^c_{jk}d^c_{ijk}\Big)/\sum_{\beta=1}^{K}\exp\Big(-\sum_{c,j} m^c_{j\beta}d^c_{ij\beta}\Big)$$

$$\mu_{ijk} = \sum_c m^c_{jk}x^c_i/\sum_c m^c_{jk} \qquad \sigma^2_{ijk} = \sum_c m^c_{jk}(x^c_i-\mu_{ijk})^2/\sum_c m^c_{jk}$$

The inferred values of the variational parameters specify a posterior distribution over the VQ states, which in turn imply a mixture of Gaussians for each input dimension. Below we use the mean of this mixture, $\hat{x^c_i} = \sum_{k,j\in k} m^c_{jk}g_{ik}\mu_{ijk}$, to measure the model's reconstruction error on case c.

4 Related models

MCVQ falls into the expanding class of unsupervised algorithms known as *factorial methods*, in which the aim of the learning algorithm is to discover multiple independent causes, or factors, that can well characterize the observed data. Its direct ancestor is Cooperative Vector Quantization [10, 3, 2], which models each data vector as a linear combination of VQ selections. Another part-seeking algorithm, non-negative matrix factorization (NNMF) [8], utilizes a non-negative linear combination of non-negative basis functions. MCVQ entails another round of competition, from amongst the VQ selections rather than the linear combination of CVQ and NNMF, which leads to a division of input dimensions into separate causes. The contrast between these approaches mirrors the development of the competitive mixture-of-experts algorithm which grew out of the inability of a cooperative, linear combination of experts to decompose inputs into separable experts.

MCVQ also resembles a wide range of generative models developed to address image segmentation [9, 4, 7]. These are generally complex, hierarchical models designed to focus on a different aspect of this problem than that of MCVQ: to dynamically decide which pixels belong to which objects. The chief obstacle faced by these models is the unknown pose (primarily limited to position) of an object in an image, and they employ learned object models to find the single object that best explains each pixel. MCVQ adopts a more constrained solution with respect to part locations, assuming that these are consistent across images, and instead focuses on the assembling of input dimensions into parts, and the variety of instantiations of each part. The constraints built into MCVQ limit its generality, but also lead to rapid learning and inference, and enable it to scale up to high-dimensional data.

Finally, MCVQ also closely relates to sparse matrix decomposition techniques, such as latent semantic analysis (LSA), and particularly probabilistic LSA, or the *aspect model* [5], a latent variable model which associates an unobserved class variable, the aspect z, with each observation. Observations consist of co-occurrence statistics, such as counts of how often a specific word w occurs in document d. It resembles MCVQ in that each input element (e.g. w, d) is represented as a convex combination of latent variables. Yet a given latent variable refers to a block within the input element-vector (word-document) matrix, while a latent variable in MCVQ pertains to a subset of elements (words), and proposes values (counts) that could be found in different vectors (documents). The aspect model is not a proper generative model of input vectors, since each vector is only represented as a set of labels for a particular vector in the training set, and there is no natural way to examine the probability of some unseen vector [1]. Below we present some initial experiments examining whether MCVQ can match the successful application of the aspect model to information retrieval and collaborative filtering problems, after evaluating it on image data.

5 Experimental Results

5.1 Parts-based Image Decomposition: Shapes and Faces

The first dataset used to test our model consisted of 11×11 gray-scale images, as pictured in Fig. 2a. Each image in the set contains three shapes: a box, a triangle, and a cross. The horizontal position of each shape is fixed, but the vertical position is allowed to vary, uniformly and independently of the positions of the other shapes. A model containing 3 VQs, 5 states each, was trained on a set of 100 shape images. The learned representation, pictured in Fig. 2b, clearly shows the specialization of each VQ to one of the shapes.

The training set was selected so that none of the examples depict cases in which all three shapes are located near the top of the image. Despite this handicap, MCVQ is able to learn the full range of shape positions, and can accurately reconstruct such an image (Fig. 2c). In contrast, standard unsupervised methods such as Vector Quantization (Fig. 3a) and Principal Component Analysis (Fig. 3b) produce holistic representations of the data, in which each basis vector tries to account for variation observed across the entire image. Non-negative matrix factorization does produce a parts-based representation (Fig. 3c), but captures less of the data's structure. Unlike MCVQ, NNMF does not group related parts, and its generative model does not limit the combination of parts to only produce valid images.

As an empirical comparison, we tested the reconstruction error of each of the aforementioned methods on an independent test set of 629 images. Since each method has one or more free parameters (e.g. the # of principal components) we chose to relate models with

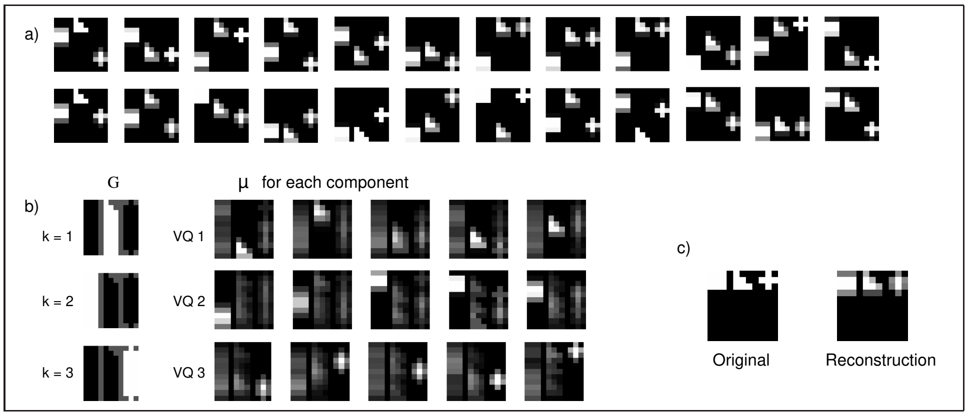

Figure 2: a) A sample of 24 training images from the Shapes dataset. b) A typical representation learned by MCVQ with 3 VQs and 5 states per VQ. c) Reconstruction of a test image: original (left) and reconstruction (right).

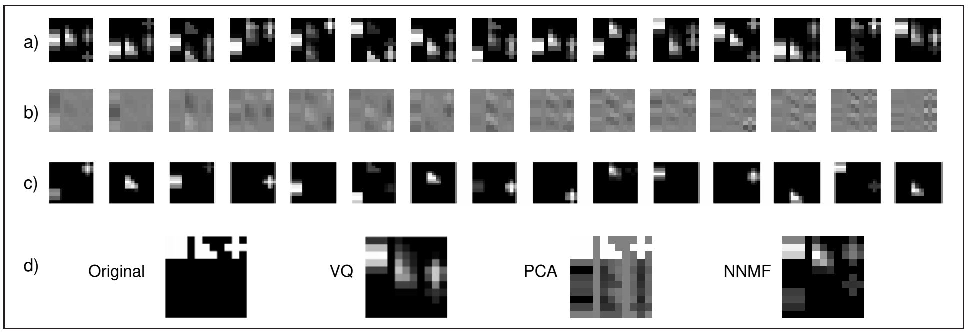

Figure 3: Other methods trained on shape images: a) VQ, b) PCA, and c) NNMF. d) Reconstruction of a test image by the three methods (cf. Fig. 2c).

similar description lengths[1]. Using a description length of about 5.9×10^5 bits, and pixel values ranging from -1 to 1, the average root mean squared reconstruction error was 0.21 for MCVQ (3 VQs), 0.22 for PCA, 0.30 for MCVQ (6 VQs), 0.35 for NNMF, and 0.49 for VQ.

As a more interesting visual application, we trained our model on a database of face images (www.ai.mit.edu/cbcl/projects).The dataset consists of 19×19 gray-scale images, each containing a single frontal or near-frontal face. A model of 6 VQs with 12 states each was trained on 2000 images, requiring 15 iterations of EM to converge. As with shape images, the model learned a parts-based representation of the faces.

The reconstruction of two test images, along with the specific parts used to generate each, is illustrated in Fig. 4. It is interesting to note that the pixels comprising a single part need not be physically adjacent (e.g. the eyes) as long as their appearances are correlated. We again compared the reconstruction error of MCVQ with VQ, PCA, and NNMF. The training and

[1] We define description length to be the number of bits required to represent the model, plus the number of bits to encode all the test examples using the model. This metric balances the large model cost and small encoding cost of VQ/MCVQ with the small model cost and large encoding cost of PCA/NNMF.

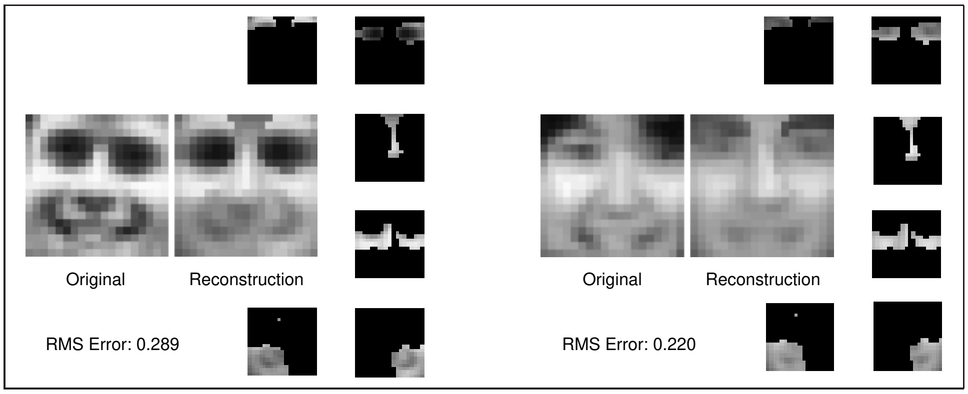

Figure 4: The reconstruction of two test images from the Faces dataset. Beside each reconstruction are the parts—the most active component in each of six VQs—used to generate it. Each part $j \in k$ is represented by its gated prediction $(g_{ik} * m_{jk})$ for each image pixel i.

testing sets contained 1800 and 629 images respectively. Using a description length of 1.5×10^6 bits, and pixel values ranging from -1 to 1, the average r.m.s. reconstruction error was 0.12 for PCA, 0.20 for NNMF, 0.23 for MCVQ (both 3 and 6 VQs), and 0.28 for VQ.

5.2 Collaborative Filtering

The application of MCVQ to image data assumes that the images are normalized, i.e., that the head is in a similar pose in each image. Normalization can be difficult to achieve in some image contexts; however, in many other types of applications, the input representation is more stable. For example, many information retrieval applications employ bag-of-words representations, in which a given word always occupies the same input element.

We test MCVQ on a collaborative filtering task, utilizing the EachMovie dataset, where the input vectors are ratings by users of movies, and a given element always corresponds to the same movie. The original dataset contains ratings, on a scale from 1 to 6, of a set of 1649 movies, by 74,424 users. In order to reduce the sparseness of the dataset, since many users rated only a few movies, we only included users who rated at least 75 movies and movies rated by at least 126 users, leaving a total of 1003 movies and 5831 users. The remaining dataset was still very sparse, as the maximum user rated 928 movies, and the maximum movie was rated by 5401 users. We split the data randomly into 4831 users for a training set, and 1000 users in a test set. We ran MCVQ with 8 VQs and 6 components per VQ on this dataset. An example of the results, after 18 iterations of EM, is shown in Fig. 5.

Note that in the MCVQ graphical model (Fig. 1), all the observation dimensions are leaves, so an input variable whose value is not specified in a particular observation vector will not play a role in inference or learning. This makes inference and learning with sparse data rapid and efficient.

We compare the performance of MCVQ on this dataset to the aspect model. We implemented a version of the aspect model, with 50 aspects and truncated Gaussians for ratings, and used "tempered EM" (with smoothing) to fit the parameters[6]. For both models, we train the model on the 4831 users in the training set, and then, for each test user, we let the model observe some fixed number of ratings and hold out the rest. We evaluate the models by measuring the absolute difference between their predictions for a held-out rating and the user's true rating, averaged over all held-out ratings for all test users (Fig. 6).

The Fugitive 5.8 (6) **Terminator 2** 5.7 (5) **Robocop** 5.4 (5)	**Pulp Fiction** 5.5 (4) **The Godfather: Part II** 5.3 (5) **The Silence of the Lambs** 5.2 (4)	**Cinema Paradiso** 5.6 (6) **Touch of Evil** 5.4 (-) **Rear Window** 5.2 (6)	**The Shawshank Redemption** 5.5 (5) **Taxi Driver** 5.3 (6) **Dead Man Walking** 5.1 (-)
Kazaam 1.9 (-) Rent-a-Kid 1.9 (-) Amazing Panda Adventure 1.7 (-)	The Brady Bunch Movie 1.4 (1) Ready to Wear 1.3 (-) A Goofy Movie 0.8 (1)	Jean de Florette 2.1 (3) Lawrence of Arabia 2.0 (3) Sense Sensibility 1.6 (-)	Billy Madison 3.2 (-) Clerks 3.0 (4) Forrest Gump 2.7 (2)
Best of Wallace & Gromit 5.6 (-) **The Wrong Trousers** 5.4 (6) **A Close Shave** 5.3 (5)	**Tank Girl** 5.5 (6) **Showgirls** 5.3 (4) **Heidi Fleiss: Hollywood Madam**5.2 (5)	**Mediterraneo** 5.3 (6) **Three Colors: Blue** 4.9 (5) **Jean de Florette** 4.9 (6)	**Sling Blade** 5.4 (5) **One Flew Over the Cuckoo's Nest** 5.3 (6) **Dr. Strangelove** 5.2 (5)
Robocop 2.6 (2) Dangerous Ground 2.5 (2) Street Fighter 2.0 (-)	Talking About Sex 2.4 (5) Barbarella 2.0 (4) The Big Green 1.8 (2)	Jaws 3-D 2.2 (-) Richie Rich 1.9 (-) Getting Even With Dad 1.5 (-)	The Beverly Hillbillies 2.0 (-) Canadian Bacon 1.9 (4) Mrs. Doubtfire 1.7 (-)

Figure 5: The MCVQ representation of two test users in the EachMovie dataset. The 3 most conspicuously high-rated (bold) and low-rated movies by the most active states of 4 of the 8 VQs are shown, where conspicuousness is the deviation from the mean rating for a given movie. Each state's predictions, μ_{ijk}, can be compared to the test user's true ratings (in parentheses); the model's prediction is a convex combination of state predictions. Note the intuitive decomposition of movies into separate VQs, and that different states within a VQ may predict very different rating patterns for the same movies.

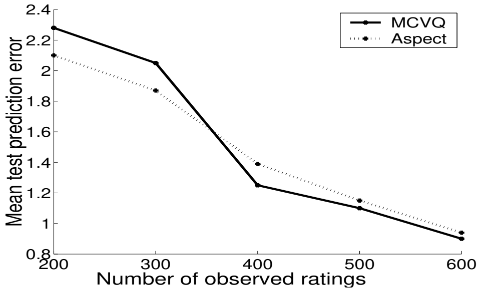

Figure 6: The average absolute deviation of predicted and true values of held-out ratings is compared for MCVQ and the aspect model. Note that the number of users per x-bin decreases with increasing x, as a user must rate at least $x+1$ movies to be included.

5.3 Text Classification

MCVQ can also be used for information retrieval from text documents, by employing the bag-of-words representation. We present preliminary results on the NIPS corpus (available at www.cs.toronto.edu/˜roweis/data.html), which consists of the full text of the NIPS conference proceedings, volumes 1 to 12. The data was pre-processed to remove common words (e.g. the), and those appearing in fewer than five documents, resulting in a vocabulary of 14,265 words. For each of the 1740 papers in the corpus, we generated a vector containing the number of occurrences of each word in the vocabulary. These vectors were normalized so that each contained the same number of words. A model of 8 VQs, 8 components each, was trained on the data, converging after 15 iterations of EM. A sample of the results is shown in Fig. 7.

When trained on text data, the values of $\{g_{ik}\}$ provide a segmentation of the vocabulary into subsets of words with correlated frequencies. Within a particular subset, the words can be positively correlated, indicating that they tend to appear in the same documents, or negatively correlated, indicating that they seldom appear together.

6 Conclusion

We have presented a novel method for learning factored representations of data which can be efficiently learned, and employed across a wide variety of problem domains. MCVQ combines the cooperative nature of some methods, such as CVQ, NNMF, and LSA, that

Predictive Sequence Learning in Recurrent Neocortical Circuits R. P. N. Rao & T. J. Sejnowski				The Relevance Vector Machine Michael E. Tipping			
afferent	**ekf**	**latent**	**ltp**	**svms**	**hme**	**similarity**	**extraction**
lgn	**niranjan**	**som**	**gerstner**	**svm**	**svr**	**classify**	**net**
interneurons	**freitas**	**detection**	**zador**	**margin**	**svs**	**classes**	**weights**
excitatory	**kalman**	**search**	**soma**	**kernel**	**hyperparameters**	**classification**	**functions**
membrane	**wp**	**data**	**depression**	**risk**	**kopf**	**class**	**units**
query	critic	mdp	spline	jutten	chip	barn	mdp
documents	stack	pomdps	tresp	pes	ocular	correlogram	pomdps
chess	suffix	prioritized	saddle	cpg	retinal	interaural	littman
portfolio	nuclei	singh	hyperplanes	axon	surround	epsp	prioritized
players	knudsen	elevator	tensor	behavioural	cmos	bregman	pomdp

Figure 7: The representation of two documents by an MCVQ model with 8 VQs and 8 states per VQ. For each document we show the states selected for it from 4 VQs. The bold (plain) words for each component are those most conspicuous by their above (below) average predicted frequency.

use multiple causes to generate input, with competitive aspects of clustering methods. It also gains combinatorial power by splitting the input into subsets, and can readily handle sparse, high-dimensional data. One direction of further research involves extending the applications described above, including applying MCVQ to other dimensions of the NIPS corpus such as authors to find groupings of authors based on word-use frequency. An important theoretical direction is to incorporate Bayesian learning for selecting the number and size of each VQ.

References

[1] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. In T.K. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, Cambridge, MA, 2001.

[2] Z. Ghahramani. Factorial learning and the EM algorithm. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*. MIT Press, Cambridge, MA, 1995.

[3] G. Hinton and R.S. Zemel. Autoencoders, minimum description length, and Helmholtz free energy. In G. Tesauro J. D. Cowan and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann Publishers, San Mateo, CA, 1994.

[4] G.E. Hinton, Z. Ghahramani, and Y.W. Teh. Learning to parse images. In S.A. Solla, T.K. Leen, and K.R. Muller, editors, *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge, MA, 2000.

[5] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[6] T. Hofmann. What people (don't) want. In *European Conference on Machine Learning*, 2001.

[7] N. Jojic and B.J. Frey. Learning flexible sprites in video layers. In *CVPR*, 2001.

[8] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

[9] C. Williams and N. Adams. DTs: Dynamic trees. In M.J. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*. MIT Press, Cambridge, MA, 1999.

[10] R.S. Zemel. *A Minimum Description Length Framework for Unsupervised Learning*. PhD thesis, Dept. of Computer Science, University of Toronto, Toronto, Canada, 1993.