The Rate Adapting Poisson Model for Information Retrieval and Object Recognition

Peter V. Gehler PGEHLER@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics, Spemannstrasse 38, 72076 Tübingen, Germany

Alex D. Holub HOLUB@VISION.CALTECH.EDU
Department Of Electrical Engineering, California Institute of Technology, MC 136-93 Pasadena, CA 91125 USA

Max Welling WELLING@ICS.UCI.EDU
Bren School of Information and Computer Science, University of California Irvine, CA 92697-3425 USA

Abstract

Probabilistic modelling of text data in the bag-of-words representation has been dominated by directed graphical models such as pLSI, LDA, NMF, and discrete PCA. Recently, state of the art performance on visual object recognition has also been reported using variants of these models. We introduce an alternative undirected graphical model suitable for modelling count data. This "Rate Adapting Poisson" (RAP) model is shown to generate superior dimensionally reduced representations for subsequent retrieval or classification. Models are trained using contrastive divergence while inference of latent topical representations is efficiently achieved through a simple matrix multiplication.

1. Introduction and Context

The dominant paradigm for modelling histogram data is the extraction of latent semantic structure, often referred to as topics. Text data for example can be represented as word counts for a given dictionary, a representation referred to as bag of words. For image data there exists an analogue, the so-called bag of features representation, which can be thought of as count data of visual words. Latent variable models determine a mapping from such count data to a compressed latent representation. This representation can subsequently be used to improve document retrieval and classification performance.

The simplest models assign each document to a single cluster a priori. However, it has been recognized that *distributed* latent representations are superior. For instance, a simple singular value decomposition of the count matrix, known as "latent semantic indexing" (LSI), is quite successful in extracting semantic structure (Deerwester et al., 1990). A probabilistic extension of this idea was introduced by Hofmann (1999) as "probabilistic latent semantic indexing" (PLSI). By realizing that PLSI is not a proper generative model at the level of documents, a further extension, latent Dirichlet allocation, was introduced by Blei et al. (2003). As pointed out by Buntine and Jakulin (2004), the basic architecture of LDA is known under various names such as add-mixtures, grade of memberships model, multiple aspect model and multinomial PCA. These authors also extend LDA to a still broader class of models known as "discrete PCA".

These models can be characterized along a number of dimensions. Firstly, they represent a subset of the of the class of *directed* graphical models, or approximations thereof. Directed models share certain properties, such as the phenomenon of explaining away (given an observation on a child node, its parents become dependent) and easy ancestral sampling. As shown in (Buntine, 2002; Girolami & Kaban, 2003) the growth of the number of parameters with the number of training documents for PLSI can be understood as variational EM learning of an LDA model, where for each training document the true posterior is replaced with a point estimate. This insight also relates non-negative matrix factorization (Lee & Seung, 1999) to the Gamma-Poisson model (Buntine & Jakulin, 2004) in a similar manner. More sophisticated approximations to the intractable inference problem have also been studied in the literature: a structured mean field approximation (Blei & Jordan, 2004), expectation propagation (Minka & Lafferty, 2002) and a collapsed Gibbs sampler (Griffiths & Steyvers, 2002).

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006.

There is also another property to characterize these models. Models such as LDA, PLSI, Gamma-Poisson models and NMF (in fact all discrete PCA models and their variational approximations) combine topics in the *probability domain*. For instance, in LDA we generate a probability vector θ with $\sum_i \theta_i = 1$ from a Dirichlet distribution and linearly combine these probabilities using a stochastic matrix M. Each column of M represents a discrete distribution over words for topic j and a document is modelled as N_{doc} samples from the linear combination $p_i = \sum_j M_{ij}\theta_j$. However, we can also take linear combinations in the *log-probability domain*. Exponential family PCA (EPCA) represents an example of this class of models. In fact we can think of EPCA exactly as a variational approximation (again using point estimates) of a model with conditional distributions in the exponential family and a flat (constant) prior. Special cases include PCA as the variational approximation of factor analysis (or probabilistic PCA)(Roweis, 1997) and the sparse coding algorithm of Olshausen and Field (1997) is a variational approximation of ICA.

Exponential family harmoniums introduced by Welling et al. (2004) can be understood as *undirected* probabilistic models which linearly combine topics in the *log-probability domain*. The undirected semantics of this model has interesting consequences. Most importantly, the latent variables are *conditionally independent* given the data, and vice versa. This is in stark contrast to the *marginal independence* of the latent variables in directed models. The implication is that the mapping from input space to latent space is given by a single matrix multiplication, possibly followed by a componentwise nonlinearity. For applications such as information retrieval and object recognition where speed if of the essence, this is a very useful property. We note that harmoniums also generate distributed latent representations.

An interesting explanation for the improved retrieval and classification results using harmoniums was given in Xing et al. (2005). These authors observe that harmoniums mix their topics using a very different mechanism than LDA. This has important consequences in particular for low count values. If a word appears only once in a document, LDA assumes a priori that this word is generated by a single topic, an assumption not made by harmoniums.

In some sense, the simpler inference in harmoniums is traded-off against more difficult learning due to the presence of an intractable normalization constant which depends on the parameters of the model. However, harmoniums are designed to take advantage of contrastive divergence learning (Hinton, 2002) which has shown to be an efficient algorithm that scales well to large problems.

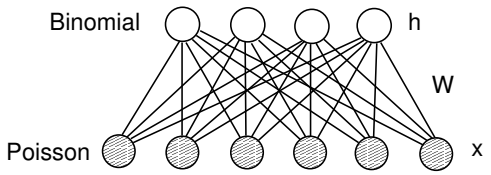

Figure 1. Markov random field representation of the RAP model. Top-layer nodes represent binomial hidden variables $\mathbf{h}$ while bottom-layer nodes represent Poisson visible variables $\mathbf{x}$.

2. The Rate Adapting Poisson Model

The rate adapting Poisson (RAP) model follows the general architecture of an exponential family harmonium (Welling et al., 2004). The RAP model is different from the "undirected probabilistic latent semantic indexing" (UPLSI) model presented in Welling et al. (2004) which uses a multinomial conditional distribution over the observed variables. This results in a large array W_{ia}^j of coupling parameters between topics and observed variables with a separate entry for every count-level a. This fact renders that model only practical for observations with very few states (e.g. binary). The RAP model is more economical in its use of parameters, coupling topics to counts using a conditional Poisson distribution involving a single matrix W_{ij}. This change has made the experiments in section 3 and 4 possible.

2.1. RAP: Generative Model

A harmonium can be specified by writing down two consistent conditional distributions in the exponential family. For RAP, we use conditional Poisson distributions for the observed count data and conditional binomial distributions for the latent topic variables,

$$p(\mathbf{x}|\mathbf{h}) = \prod_i \text{Pois}_{x_i}[\log(\lambda_i) + \textstyle\sum_j W_{ij}h_j] \quad (1)$$

$$p(\mathbf{h}|\mathbf{x}) = \prod_j \text{Bin}_{h_j}[\sigma(\log(\tfrac{p_j}{1-p_j}) + \textstyle\sum_i W_{ij}x_i); M_j] \quad (2)$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function, λ_i is the mean rate of the conditional Poisson distribution for word i, p_j is the "probability of success" and M_j the total number of "samples" for the conditional binomial distribution for topic j, $\mathbf{x}$ is the count vector, $\mathbf{h}$ a discrete topic vector and W the interaction between topics and counts. From these equations it can be seen that the value of the variables of the opposite layer shift the canonical parameters of the variables in the layer under consideration. It is due to this behavior that we named the model "rate adapting". Note also that all variables are *conditionally independent* given values for the variables in the opposite layer.

These two conditional distributions are consistent with the joint distribution over $\{\mathbf{x}, \mathbf{h}\}$ defined through $p(\mathbf{x}, \mathbf{h}) =$

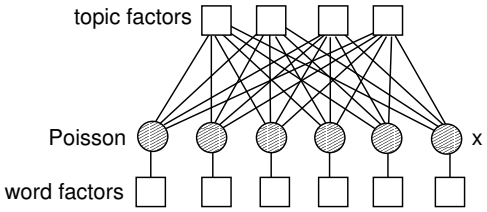

Figure 2. Factor graph representation of the marginalized RAP model. Square boxes indicate word and topic factors.

$\exp[f(\mathbf{x},\mathbf{h})]/Z$ with

$$f(\mathbf{x},\mathbf{h}) = \sum_i \log(\lambda_i)x_i - \log(x_i!) + \sum_j \log(\frac{p_j}{1-p_j})h_j - \log(h_j!) - \log((M_j - h_j)!) + \sum_{ij} W_{ij}x_i h_j \quad (3)$$

where we have not written any terms that do not explicitly depend on a random variable. The two-layer undirected architecture of this model is shown in figure 1.

Samples from the model can be obtained efficiently by Gibbs sampling because all variables in a layer can be sampled in parallel given the values for the variables of the opposite layer and vice versa.

To find the most likely variable assignments one can locate modes of the distribution by iterating the equations,

$$x_i^{\text{mode}} = \lfloor \exp(\log(\lambda_i) + \sum_j W_{ij}h_j) \rfloor \quad (4)$$

$$h_j^{\text{mode}} = \lfloor (M_j+1)\sigma(\log(\frac{p_j}{1-p_j}) + \sum_i W_{ij}x_i) \rfloor \quad (5)$$

The RAP model can also be represented as a factor graph (see figure 2) by marginalizing out the latent variables,

$$p(\mathbf{x}) \propto \exp[\sum_i (\log(\lambda_i)x_i - \log(x_i!)) + \sum_j M_j \log(1+\exp(\sum_i W_{ij}x_i - \beta_j))] \quad (6)$$

where we have abbreviated $\beta_j = -\log[p_j/(1-p_j)]$. We can read out the word-factors from this expression as $F_i(x_i) = \lambda_i^{x_i}/x_i!$ for each variable and the topic-factors $F_j(\mathbf{x}) = \exp(M_j \log(1+\exp(\sum_i W_{ij}x_i - \beta_j)))$. Note that the factors $F_j(\mathbf{x})$ are functions of all the variables $\mathbf{x}$ jointly. The nonlinearity for a topic-factor in the log domain is precisely given by the "hinge function" (see figure 3). Hence, a topic factor does not contribute to the probability distribution (i.e. $F_j(\mathbf{x}) = 1$) if the input count vector $\mathbf{x}$ is not well aligned with the topic vector $\mathbf{w}_j = \{W_i\}_j$. A threshold β_j determines what it means to be "well aligned":

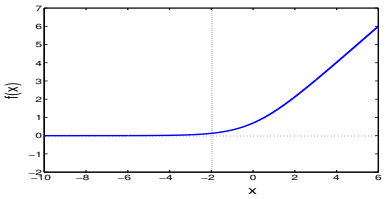

Figure 3. Hinge nonlinearity

if $\mathbf{w}_j^T\mathbf{x} \ll \beta_j$ then the factor does not contribute. On the other hand, if $\mathbf{w}_j^T\mathbf{x} \gg \beta_j$ then the hinge function is linear, and hence those factors modulate the log Poisson rate λ_i as follows,

$$\log(\lambda_i) \leftarrow \log(\lambda_i) + \sum_j \mathbb{I}(\sum_i W_{ij}x_i > \beta_j)\ M_j W_{ij} \quad (7)$$

where $\mathbb{I}(\cdot)$ is the indicator function. Clearly, this is an approximation because there is in fact a soft transition between the two regimes of the hinge function[1]. However, it clarifies the role of the weight matrix as a collection of topic vectors that form a new low dimensional basis for the latent representation. Count vectors get mapped into latent topic space by computing its coordinates in this basis as $\tilde{\mathbf{h}} = W\mathbf{x}$. The thresholds β then decide on the necessary magnitude of these coordinates before they will have an impact on the Poisson rates. We note that there are in fact 2^K (with K the total number of topics) different ways to modulate the log Poisson rates because there are 2^K subsets of $\{1,...,K\}$. Empirically we have found that the best performance in terms of retrieval and classification is obtained when the angle between latent coordinates is used as a measure of similarity: $K(\mathbf{x}_n,\mathbf{x}_m) = \cos(\tilde{\mathbf{h}}_n^T\tilde{\mathbf{h}}_m)$. This is not surprising as we expect that the length of a document roughly scales the count vector linearly assuming its topical content does not change.

2.2. RAP: Invariant Transformations

The marginal distribution $p(\mathbf{x})$ in equation (6) is given as a product of factors where each factor follows the general form $\log(1+e^z)$. It is not hard to check that the following identity holds, $\log(1+e^z) = z + \log(1+e^{-z})$ which has the consequence that we can change parameters without affecting the model. In other words, the parameters are not identifiable in the current parameterization. If we define an arbitrary subset $\mathcal{S}$ of the integers $\{1,...,K\}$, then the following transformations, when executed jointly, do

[1]This approximation is expected to be accurate when all parameter values $\{W,\beta\}$ are large.

not change the RAP model,

$$\log(\lambda_i) \rightarrow \log(\lambda_i) + \sum_{j \in \mathcal{S}} M_j W_{ij} \tag{8}$$

$$W_{ij} \rightarrow -W_{ij} \qquad \beta_j \rightarrow -\beta_j \qquad j \in \mathcal{S}. \tag{9}$$

Fortunately, it is easy to fix the spurious degrees of freedom by choosing for instance $\mathbf{w}_j^T \mathbf{x} > 0 \; \forall j$ or alternatively, fixing the sign of $\beta_j, \;\; \forall j$.

Going one step further, we can apply the transformation only to half the hinge nonlinearity and obtain, $\log(1+e^z) = \frac{1}{2}z + \frac{1}{2}\log(1+\cosh(z)) + \frac{1}{2}\log(2)$ where the constant term is absorbed in the normalization and the linear term is absorbed in the variable factors F_i. The new topic factors, $\tilde{F}_j = \sqrt{1+\cosh(z)}$, are now symmetric around $z = 0$ implying that large inner products $\tilde{\mathbf{w}}_j^T \mathbf{x}$ *with either sign* (i.e. aligned or anti-aligned) result in a positive contribution of that factor to the probability of input $\mathbf{x}$. We will call this type of basis vectors $\{\mathbf{w}_j\}$ "prototypes" in contrast to "constraints" which contribute positive probability for an input $\mathbf{x}$ when it is approximately orthogonal to it: $\mathbf{w}_j^T \mathbf{x} \approx 0$ (Welling et al., 2002).

2.3. RAP: Learning

Parameter learning for the RAP model is performed by stochastic gradient ascent on the log-likelihood of the data. For large redundant datasets it is more efficient to estimate the required gradients on small batches rather than on the entire dataset. This is true in particular in the initial phase of learning where there is consensus among the data on how to change the parameters. Towards convergence it is useful to either increase the batch-size or decrease the stepsize in order to reduce the variance of the stochastic optimization. We have also included a momentum term to help speed up convergence.

The derivatives of the log-likelihood of the RAP model are easy to write down (but hard to calculate in practice due to the intractable normalization constant),

$$\delta \log \lambda_i \propto \langle x_i \rangle_{\tilde{p}} - \langle x_i \rangle_{p_T} \tag{10}$$

$$\delta \beta_j \propto -M_j [\langle \sigma(\mathbf{w}_j^T \mathbf{x} - \beta_j) \rangle_{\tilde{p}} - \langle \sigma(\mathbf{w}_j^T \mathbf{x} - \beta_j) \rangle_{p_T}]$$

$$\delta W_{ij} \propto M_j [\langle x_i \sigma(\mathbf{w}_j^T \mathbf{x} - \beta_j) \rangle_{\tilde{p}} - \langle x_i \sigma(\mathbf{w}_j^T \mathbf{x} - \beta_j) \rangle_{p_T}]$$

where $\tilde{p}$ denotes the empirical distribution[2] and p_T the model distribution at the current values of the parameters. Note that our estimate of the gradients of β and W involves Rao-Blackwellisation over the latent variables. Here we replace a sample average $\frac{1}{N}\sum_{n=1}^{N} f(\mathbf{h}_n)$ with $\frac{1}{N}\sum_{n=1}^{N} f(\mathbf{h}) p(\mathbf{h}|\mathbf{x}_n)$. This is *guaranteed* to reduce the variance of our estimates (Casella & Robert, 1996).

[2]The average over the empirical distribution is simply given by the sample average over the data-cases.

It is in particular the negative terms in these equations that are hard to estimate. One approach is to run the Gibbs sampler defined by equations (1) and (2). Note however that at every iteration of learning we have to run this sampler to equilibrium. Instead, we will follow the contrastive divergence (CD) paradigm where for every data-case in the batch we initialize a separate Gibbs sampler at that data-case and run it for only a few steps. With p_1 (i.e. $T = 1$ in equation (10)) we will denote the Gibbs chain[3] that samples: $\mathbf{h}_n^0 \sim p(\mathbf{h}|\text{data-case}_n) \rightarrow \mathbf{x}_n^1 \sim p(\mathbf{x}|\mathbf{h}_n^0)$. CD-learning simply boils down to obtaining samples from p_1 through the above one-step Gibbs chain and computing noisy estimates of the gradient through equation 10. The averages in the first terms are again computed as sample estimates of data-cases in the batch while the averages in the second terms are computed as sample averages over the samples from p_1.

Although truncating the Markov chain will introduce a bias in the estimates of the gradients, the final bias of the parameter estimates has been shown to be small empirically for a number of applications (Carreira-Perpinan & Hinton, 2005). Moreover, the variance of the gradient estimates and hence the variance of the final parameter estimates is greatly reduced (albeit at the expense of introducing a bias).

Below a summary of the CD-learning algorithm as described in the preceding text. We have also implemented

Algorithm 1 Contrastive Divergence Learning for RAP

Repeat until convergence:

1 For each data-case x_n do:

1a Sample the hidden units given the data-case clamped to the visible units from $\mathbf{h}_n^0 \sim \prod_j p(h_{jn}|x_n)$ using Eqn.(2).

1b Resample the data-case given the sampled values of the hidden units from $\mathbf{x}_n^1 \sim \prod_i p(x_{in}|\mathbf{h}_n^0)$ given in Eqn.(1).

2 Compute the data averages and sample averages in Eqn.(10) with T=1.

3 Perform gradient updates according to Eqn.(10) with T=1.

a mean field learning algorithm where Gibbs sampling updates are replaced by mean field updates (Welling & Hinton, 2001), but we found the results to be significantly inferior to the sampling based algorithm.

3. Experiments: Document Retrieval

In this and the next section we describe how the latent structure of the RAP model can be used for two different tasks, namely document retrieval and object recognition[4]. We compare its performance against two other latent vari-

[3]Note that sampling from the equilibrium distribution should be denoted as p_∞, i.e. $T = \infty$.

[4]Matlab code for training the RAP model and the prerpocessed text data can be obtained from http://www.kyb.mpg.de/∼pgehler

able models: PLSI and LSI. Performance of LDA has never significantly surpassed PLSI (in fact we often found inferior results) which is the reason we left them out.

In document retrieval the goal is to match a given query, represented by a word count vector, with a subset of a text corpus where the retrieved subset should resemble the query as closely as possible. A latent variable model can be turned into a document retrieval algorithm through the following three steps: 1) estimate the parameters of the model on a training corpus, 2) map all training and query documents into the dimensionally reduced latent space, 3) compute similarities between queries and training documents based on the latent representation, 4) retrieve the k most similar training documents form the corpus for every query. We have used the cosine similarity measure in our experiments. One can also compute similarity in (tf-idf reweighted) word space directly which we use in our experiments as a baseline.

3.1. Text Corpora

In these experiments we used three well known datasets: Reuters-21578, Ohsumed, and 20-Newsgroups[5]. We use the BOW package and its front-end RAINBOW to preprocess the data (McCallum, 1996). All documents were stemmed with the Porter stemmer, a list of stop-words and all words with less than three characters were removed. Additionally, for the Reuters and Ohsumed datasets all words were removed which occur only once in the training data or in only a single training document. For the 20-Newsgroups dataset the 10000 words with highest average mutual information with the class variable were extracted. This preprocessing left the Newsgroup dataset with 10000 words, 18798 documents and 20 classes, and the Reuters dataset with 12317 words, 15437 documents and 91 classes (we also used another split of the data with 115 classes but found very similar results). The Ohsumed dataset consists of 30689 words, 34389 documents and 23 classes where each data-point might belong to more than one class. The corpus was split into a training set and a test set whose items are used as query documents during the performance evaluation. For the Reuters dataset the predefined ModApte split of the data into 11413 train and 4024 test documents was used. Ohsumed is split in 33% test and 67% training data while in the newsgroup corpus we held out 10% for testing purposes.

[5]These corpora are available from http://ai-nlp.info.uniroma2.it/moschitti/corpora.htm. The original sources and specifics concerning these sets can be found on this site and are omitted here for brevity.

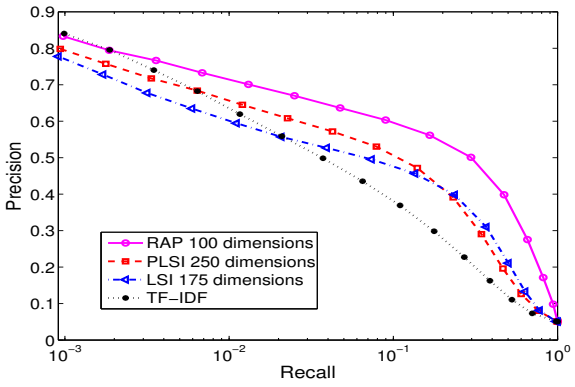

Figure 4. RPC plot on a log-scale of the 20 Newsgroups dataset for various models. As a baseline the retrieval results with tf-idf reweighed word-counts are shown. Number of topics for each model was chosen by optimizing 1-NN classification performance on the test set corresponding to the average precision for retrieving a single document (left most marker).

3.2. Results

Learning of the RAP model was done with a small learning rate and a momentum term in 200k iterations using mini-batches of 100 training samples per iteration. The latent representation of any document $\mathbf{x}$ is then computed by a matrix multiplication $W\mathbf{x}$. LSI is computed by performing a SVD decomposition on the tf-idf [6] reweighed word counts. PLSI models are trained using the tempered version of the EM algorithm (Hofmann, 1999). 10% of the training data was held out for validation purposes and the temperature parameter β is initialized at 1 and whenever the log-likelihood on the validation data decreases β is decreased about .025 until no more improvement was observed. The latent representation is defined by the posterior distribution over the topics $\mathbf{z}$: $P(\mathbf{z}|\mathbf{d})$. For a query document $\mathbf{q}$, $P(\mathbf{z}|\mathbf{q})$ was computed using 25 iterations of the folding-in heuristic (Hofmann, 1999). For comparison we also show the baseline results which are obtained by computing the similarity of tf-idf reweighed documents in word space. As performance measure we use the recall precision curve (RPC) where

$$\text{Recall} = \frac{\#(\text{correctly retrieved documents})}{\#(\text{relevant documents in the corpus})} \qquad (11)$$

$$\text{Precision} = \frac{\#(\text{correctly retrieved documents})}{\#(\text{retrieved documents})}. \qquad (12)$$

For a given test document, all training documents were ranked in terms of their cosine similarity. Then recall and precision values were computed for $1, 2, 4, 8, 16 \ldots$ re-

[6]$\text{tf-idf}(d,w) = \frac{n(d,w)}{\sum_{w'} n(d,w')} \log_2 \left[\frac{\#\text{docs in the corpus}}{\#\text{docs with word w}}\right]$, where $n(d,w)$ are the occurrences of word w in document d

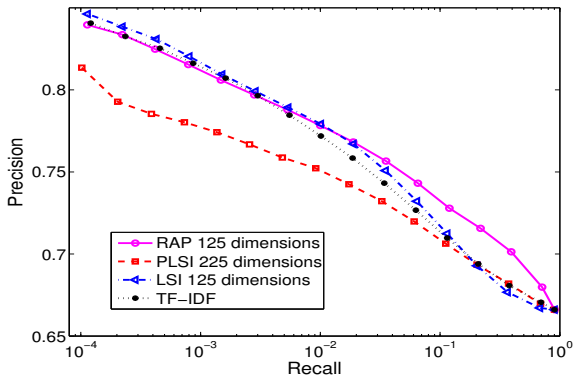

Figure 5. Same as figure 4 for Ohsumed dataset.

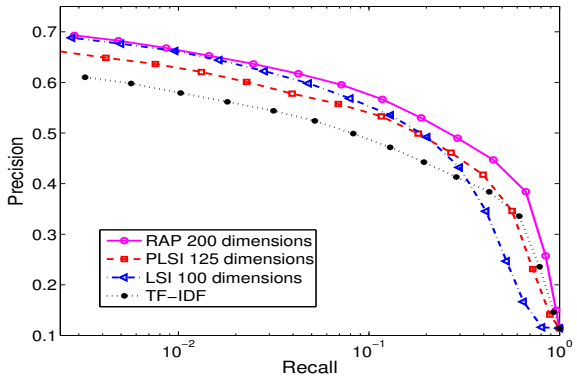

Figure 6. Same as Figure 4 for the Reuters dataset.

trieved documents. The RPC curves of all models are plotted in Figures 4, 5 and 6, where the recall and precision values are averaged over the entire test-set. In figure 7 we show the "area under the RPC" (AUC) as a function of the number of topics.

The leftmost point on an RPC, i.e. the average precision for retrieving a single document, corresponds to the 1-NN classification performance using the cosine distance. The latent dimensionality of the models shown in the plots were selected to be the best according to this measure, where we scanned the number of topics from 25 to 250 at increments of 25. The RAP model yields the best retrieval performance on all datasets in terms of AUC, and scores only slightly worse than LSI on Ohsumed in terms of 1-NN classification performance. According to figure 7 the RAP model also seems to suffer less from overfitting as the number of topics increases.

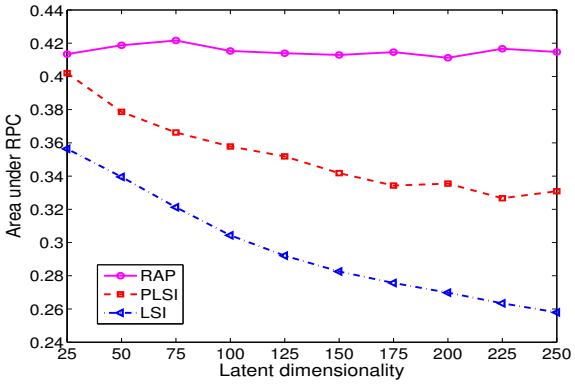

Figure 7. Area under the RPC as a function of the latent dimensionality on the Reuters dataset.

4. Experiments: Object Recognition

Latent models have recently been applied to both object (Fergus et al., 2005) and scene (Li & Perona, 2005) recognition. In this section we compare the performance of the RAP model in the visual object recognition domain. We followed these steps in our visual experiments: (1) interest point detection and extraction, (2) vocabulary generation, (3) latent analysis, (4) kernel classification on latent representations. We briefly describe these steps below.

Images were initially normalized to be the same size. Interesting regions of images (interest points) were detected using three different feature detectors: multi-scale Harris, multi-scale Hessian, and entropy-based (Kadir & Brady, 2001). Grey-scale patches were extracted from images based on both the scale and location indicated by the different interest point detectors. All patches from all detectors were intensity normalized and resized to 13×13 and subsequently converted into vectors. We performed K-means clustering on the patches in order to discretize feature space and create a visual vocabulary of words. The number of clusters was left as a free parameter of the system and typically varied from $100 - 300$. Each image contains a set of interest point detections. An interest point was assigned to the visual cluster (word) closest in a Euclidean sense to that feature. The cumulative counts over all clusters were used as feature vectors to represent each image, such that each image was represented by a vector of dimensionality equal to the size of the visual vocabulary. Similar to the document experiments described above, we are not utilizing any spatial information between the extracted patches. We compared the performance of three different latent algorithms described above: LSI, PLSI and RAP. The latent representations for each image were used to train SVM classifiers using the LIBSVM[7] package with a linear kernel. Ten-

[7] Available at: www.csie.ntu.edu.tw/∼cjlin/libsvm/.

Figure 8. Example images used for the object recognition experiments. (Top Row) Example images from the Caltech4. Classes: Airplanes, Motorcycles, Faces, Leopards. (Bottom Two Rows) Example images of four random classes from the Caltech101. Two images of each class shown to give an indication of the within class variance. Classes: Budha, Chair, Watch, Brain. Note that the Caltech101 includes the Caltech4 classes.

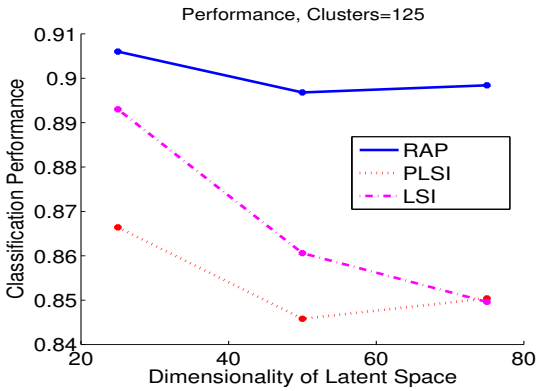

Figure 9. Caltech4 performance comparison. All experiments averaged 35 times. Baseline (chance) performance is 25%. Plotted is the test performance as a function of the number of latent dimensions with 125 clusters and using a linear kernel. Performance differences between RAP and PLSI/LSI were significant for all numbers of latent dimensions ($p < 0.05$).

fold cross-validation was used to find the optimal values of the SVM hyper-parameters. We used a one-vs-one training paradigm for the multi-class datasets. Feature dimensions were normalized to zero variance and unit standard deviation. We conducted experiments on both the Caltech4 and the challenging Caltech101 datasets (figure 8 illustrates representative examples of some categories). These datasets can be found at: www.vision.caltech.edu/html-files/archive.html. The Caltech4 contains a total of 4 object categories and is regarded to be relatively easy to classify due to stereotyped poses and drastic visual dissimilarity between classes. The Caltech101 contains a total of 101 object categories and is more challenging due to the sheer number of object categories. 15 training images and a maximum of 50 testing images were used for all experiments. For the Caltech101, the class "Faces-Easy" was removed. Performance results reported correspond to the average classification performance across all categories. Figures 9, 10 and 11 show comparisons between RAP and LSI/PLSI. Error bars are not shown because the variation from one split of the data to another was larger than the variation between models. Instead we used the two-sided paired sign test to determine whether the median difference in performance is significantly different from zero at a level of $\alpha = 0.05$. We conclude that almost always RAP significantly outperforms LSI and PLSI.

5. Discussion

The experiments provide clear evidence for the claim that harmonium models, and in particular RAP, can be efficiently and successfully trained on relatively large datasets. Relative to popular existing methods such as LSI and PLSI the latent representations generated by RAP are superior in two application domains: document retrieval and object classification. Moreover, mapping test-data into latent space is orders of magnitude faster for RAP (through a simple matrix multiplication) than for PLSI (through the iterative "folding-in" heuristic).

A natural next step is to train hierarchical models. Dependencies between topics are then modelled with a new layer of "meta-topics". Initial experiments in this direction have not shown improved retrieval or classification performance. However, recent work by (Hinton et al., 2006) indicates that deep hierarchies can be a promising direction for improvement.

The choice of a conditional Poisson distribution may not be optimal due to the effect that words that have been used already become more likely to be used than others, i.e. their frequency grows with document length. This calls for distributions with longer tails such as the negative-Binomial distribution (Airoldi et al., 2005). The Poisson distribution in RAP can be easily interchanged with a negative-Binomial incorporating this effect.

A Bayesian approach for harmonium models seems an important topic for future investigation.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0447903.

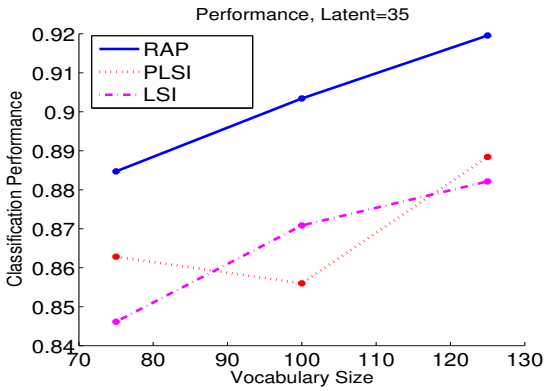

Figure 10. Same experiment as in figure 9 but plotting performance as a function of the size of the vocabulary using 35 latent dimensions. Performance differences between RAP and PLSI/LSI were significant for vocabulary sizes ($p < 0.05$).

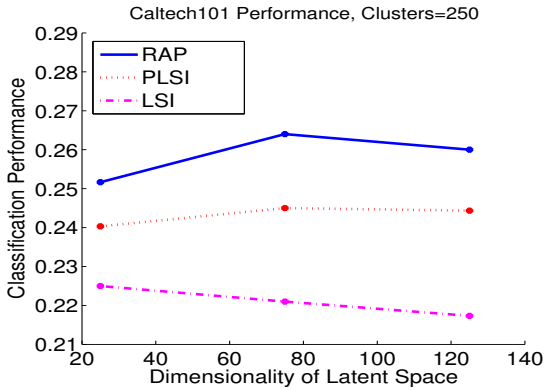

Figure 11. Caltech101 performance comparisons using 250 clusters. All experiments averaged 7 times. Baseline (chance) performance is 1% for this task. Same plot as in figure 9. Performance differences between RAP and PLSI were significant for 75 and 125 latent dimensions ($p < 0.05$).

References

Airoldi, E., Cohen, W., & Fienberg, S. (2005). Bayesian methods for frequent terms in text. *Proc. of the CSNA & INTERFACE Annual Meetings*.

Blei, D. M., & Jordan, M. I. (2004). Variational inference for dirichlet process mixtures. *Bayesian Analysis*, *1*, 121–144.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Buntine, W. (Ed.). (2002). *Variational extensions to em and multinomial pca*, vol. 2430 of *Lecture Notes in Computer Science*. Helsinki, Finland: Springer.

Buntine, W., & Jakulin, A. (2004). Applying discrete pca in data analysis. *Proceedings of the 20th conference on Uncertainty in artificial intelligence* (pp. 59–66). Banff, Canada.

Carreira-Perpinan, M., & Hinton, G. (2005). On contrastive divergence learning. *Tenth International Workshop on Artificial Intelligence and Statistics*. Barbados.

Casella, G., & Robert, C. (1996). Rao-blackwellisation of sampling schemes. *Biometrika*, *83(1)*, 81–94.

Deerwester, S., Dumais, S., Landauer, T., Furnas, G., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*, 391–407.

Fergus, R., Fei-Fei, L., Perona, P., & Zisserman, A. (2005). Learning object categories from google's image search. *Proceedings of the International Conference on Computer Vision*.

Girolami, M., & Kaban, A. (2003). On an equivalence between PLSI and LDA. *Proceedings of SIGIR 2003*.

Griffiths, T., & Steyvers, M. (2002). A probabilistic approach to semantic representation. *Proceedings of the 24th Annual Conference of the Cognitive Science Society*.

Hinton, G. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*, 1771–1800.

Hinton, G., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief networks. *Neural Computation*. to appear.

Hofmann, T. (1999). Probabilistic latent semantic analysis. *Proc. of Uncertainty in Artificial Intelligence, UAI'99*. Stockholm.

Kadir, T., & Brady, M. (2001). Saliency, scale and image description. *Int. J. Comput. Vision*, *45*, 83–105.

Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

Li, F., & Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

McCallum, A. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/∼mccallum/bow.

Minka, T., & Lafferty, J. (2002). Expectation-propogation for the generative aspect model. *Proc. of the 18th Annual Conference on Uncertainty in Artificial Intelligence* (pp. 352–359).

Olshausen, A., & Field, D. (1997). Sparse coding with overcomplete basis set: A strategy employed by v1? *Vision Research*, *37*, 3311–3325.

Roweis, S. (1997). Em algorithms for pca and spca. *Neural Information Processing Systems* (pp. 626–632).

Welling, M., & Hinton, G. (2001). A new learning algorithm for mean field Boltzmann machines. *Proc. of the Int'l Conf. on Artificial Neural Networks*. Madrid, Spain.

Welling, M., Hinton, G., & Osindero, S. (2002). Learning sparse topographic representations with products of student-t distributions. *Neural Information Processing Systems*.

Welling, M., Rosen-Zvi, M., & Hinton, G. (2004). Exponential family harmoniums with an application to information retrieval. *Neural Information Processing Systems*.

Xing, E., Yan, R., & Hauptman, A. (2005). Mining associated text and images with dual-wing harmoniums. *Proc. of the Conf. on Uncertainty in Artificial Intelligence*.