SPARSE CODING AND AN APPLICATION TO TOPIC MODELS

By Ryan Wang*

Sparse coding represents data as sparse linear combinations of basis vectors from a learned dictionary. Algorithms for sparse coding generally alternate between a coding step, which involves a regularized least squares problem, and a dictionary learning step, which involves a matrix factorization problem. In this paper we review some recent theoretical work on the performance of procedures used for solving these problems, such as the LASSO. We then present an application to topic modeling, a set of techniques for analyzing collections of text documents, and discuss some possible extensions that incorporate structured sparsity. Finally, we discuss experimental results from applying a sparse coding topic model to a collection of papers from the Neural Information Processing Systems (NIPS) conference.

1. Introduction. Sparse coding provides a set of methods for representing data in terms of a set of basis vectors called a dictionary. In many cases it is useful to have sparse representations in terms of the dictionary, so that only a few elements from the dictionary make up any one representation. For example, sparse representations have been useful in applications to high-dimensional data and signal processing. In sparse coding the dictionary is simultaneously learned, in contrast to methods that use predefined dictionaries such as wavelet bases. In this paper we review some recent work on aspects of sparse coding and present an application to topic modeling, a widely used procedure for text analysis.

The general setup is as follows. A response vector $y_i \in \mathbb{R}^m$ is represented as a linear combination of basis vectors $b_1, ..., b_p \in \mathbb{R}^m$ with code vector $\theta_i = (\theta_1, ..., \theta_n)^T \in \mathbb{R}^p$, so that $\hat{y}_i = \sum_{j=1}^p \theta_i j b_j$. The problem is then to minimize reconstruction error with some kind of sparsity inducing penalty. For observations $y_1, ..., y_n$:

$$\min_{\{b_j\}_{j=1}^p, \{\theta_i\}_{i=1}^n} \frac{1}{2} \sum_{i=1}^n ||y_i - \sum_{j=1}^p \theta_{ij} b_j||_2^2 + \omega \sum_{i=1}^n \phi(\theta_i) \tag{1.1}$$

where the function $\phi(\cdot)$ induces sparsity in the coefficient vectors. The ℓ_0 pseudo-norm,

$$||\theta_i||_0 \triangleq \#\{j \text{ s.t. } \theta_{ij} \neq 0\}$$

*rywang@galton.uchicago.edu

and its convex relaxation the ℓ_1 norm,

$$||\theta_i||_1 \triangleq \sum_{j=1}^{p} |\theta_{ij}|$$

are most common. We will focus on the ℓ_1 norm since it admits widely used procedures for convex optimization.

The optimization problem in (1.1) also has a matrix representation, which can be useful for applying matrix factorization methods. Let $Y \in \mathbb{R}^{m \times n}$ be the matrix of stacked observation vectors, $B \in \mathbb{R}^{m \times p}$ the matrix of stacked basis vectors, and $\Theta \in \mathbb{R}^{p \times n}$ the matrix of stacked codes. Then the problem becomes:

$$\min_{B,\Theta} \frac{1}{2}||Y - B\Theta||_F^2 + \omega \sum_{j=1}^{n} \phi(\Theta_{\cdot j}) \tag{1.2}$$

where $||\cdot||_F$ denotes the Frobenius norm for matrices. It is typical to constrain the size of the columns of B in order to prevent the codes from becoming arbitrarily small. This constraint set often takes the form

$$\mathcal{C} = \{B \in \mathbb{R}^{m \times p} : b_j^T b_j \leq 1 \text{ for all } j = 1, ..., p\}.$$

In general (1.2) does not yield a jointly convex problem, even under an ℓ_1 penalty. However, it is convex with respect to B when Θ is fixed, and vice versa, under an ℓ_1 penalty. Thus, the classical approach is to alternately optimize with respect to each variable holding the other fixed. There has not been much theoretical study on the properties of this procedure, though we will briefly discuss some results for each component.

The rest of the paper is organized as follows. Section 2 reviews some relevant theoretical work on sparse estimation procedures. Section 3 describes an original application of sparse coding to topic modeling. Section 4 presents numerical experiments. Section 5 discusses some possible extensions.

2. Some Theoretical Results. Results on the performance of sparse estimation procedures often take the form of oracle inequalities. An oracle inequality provides guarantees on the statistical performance of an estimator in terms of its risk relative to an oracle estimator. The oracle represents the best model within a given family of models and can be viewed as the infinite data estimator. Thus if an estimator has risk close to the oracle risk, then the estimator performs well up to the error incurred from specifying a particular family of models. We first discuss oracle inequalities for an ℓ_1-penalized sparse density estimator. In the case of regression, an equivalent

estimator for solving the ℓ_1-penalized least squares problem is known as the least absolute shrinkage and selection operator (LASSO). This work is relevant because the coding step directly corresponds to solving such a problem.

2.1. *SPADES.* Consider the following density estimation setup from [4]. Let $X_1, ..., X_n \in \mathbb{R}^m$ be independent and identically distributed with density f. Let $\{f_1, ..., f_p\}$ be a given finite set of functions with $f_j \in L_2(\mathbb{R}^m)$ called a dictionary. The unknown density f is modeled by linear combinations of elements of the dictionary

$$f_\theta(x) = \sum_{j=1}^{p} \theta_j f_j(x) \tag{2.1}$$

with coefficient vector $\theta = (\theta_1, ..., \theta_p) \in \mathbb{R}^p$. To induce sparsity in the coefficients, an ℓ_1-penalized criterion is minimized.

The SPADES (SPArse Density EStimation) estimator minimizes an empirical counterpart of the squared loss. Noting that the inner product of $f, g \in L_2(\mathbb{R}^m)$ corresponds to an expectation under f, the squared loss becomes

$$||f - f_\theta||^2 = ||f||^2 + ||f_\theta||^2 - 2 < f, f_\theta >= ||f||^2 + ||f_\theta||^2 - 2E\left(f_\theta(X)\right). \tag{2.2}$$

Minimizing (2.2) with respect to θ corresponds to minimizing

$$\gamma(\theta) = -2E\left(f_\theta(X)\right) + ||f_\theta||^2. \tag{2.3}$$

Replacing the expectation by its empirical counterpart, $\frac{1}{n}\sum_{i=1}^{n} f_\theta(X_i)$, and adding an ℓ_1-penalty, $\phi(\theta) = \omega \sum_{j=1}^{p} |\theta_j|$, the SPADES estimator is then defined by

$$\hat{f}_{\text{SPADES}} = f_{\hat{\theta}}$$

where

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ -\frac{2}{n} \sum_{i=1}^{n} f_\theta(X_i) + ||f_\theta||^2 + \omega \sum_{j=1}^{p} |\theta_j| \right\}. \tag{2.4}$$

2.1.1. *General Oracle Inequality.* Define the noise level by

$$r(\delta) = r(p, n, \delta) = \sqrt{\frac{\log(p/\delta)}{n}}$$

where $0 < \delta < 1$ is specified. Finally, define the intra-dictionary correlations

$$\rho(i,j) = \frac{< f_i, f_j >}{||f_i||||f_j||}$$

for $i, j = 1, ..., p$. Under a condition on the "maximal local coherence"

$$\rho(\theta) = \max_{i \in I(\theta)} \max_{j \neq i} |\rho(i,j)|,$$

namely that $\theta \in \mathbb{R}^p$ satisfy $16C\rho(\theta) \leq 1$, Bunea et al (2010) show that for a simple choice of regularization parameter

$$\omega \propto r(\delta/2) \tag{2.5}$$

the following inequality holds with probability at least $1 - \delta$

$$||f_{\hat{\theta}} - f||^2 + C_2\omega \sum_{j=1}^{p} |\hat{\theta}_j - \theta_j| \leq C_3||f_\theta - f||^2 + C_4 r^2(\delta/2)g(\theta) \tag{2.6}$$

where the C's are constants. This result suggests that when the mixture representation is a good approximation of the function, that is the approximation error $||f_\theta - f||^2$ is small, then simultaneously the SPADES coefficients will be close to the oracle coefficients and the SPADES estimator will be close to the true density. A similar result holds under a condition on the "cumulative local coherence" defined by

$$\rho(\theta) = \sum_{i \in I(\theta)} \sum_{j > i} |\rho_M(i,j)|.$$

The proofs rely on an application of Bernstein's inequality, which describes the concentration of random variables around their mean.

For the purposes of sparse coding, the coherence conditions suggests that we should consider the correlations between dictionary elements if we are interested in predictive performance. In particular we do not want the elements to be highly correlated. This makes intuitive sense as there would be a problem of identification in such a case. The cumulative condition is weaker than the maximal condition in that it admits cases where the correlations may be relatively large for only a few pairs of dictionary elements. Note also that the oracle inequality is non-asymptotic and does not depend on M or n.

2.1.2. *Mixture Models.* In the mixture modeling context, we assume that $f = f_{\theta^*}$ has a true mixture representation that is sparse. Denote by $I^* \subseteq \{1, .., p\}$ the set of indices corresponding to the non-zero components of θ^*. The goal is then to find a good estimate $\hat{\theta}$ of θ^* and identify I^* with high probability, formally $p(I_{\hat{\theta}} = I^*) \to 1$ as $n \to \infty$. This is often referred to as model selection consistency, although the result from [4] is non-asymptotic, as before.

In this setting the coherence condition can be restated as

$$\rho^* \leq \frac{1}{16k^*} \tag{2.7}$$

where $\rho^* = \rho(\theta^*)$ and $k^* = |I^*|$. Intuitively, the condition implies that stricter conditions on the correlations are required as the number of true non-zero coefficients, k^*, increases. Put another way, the model selection problem becomes more difficult when the true coefficient vector is less sparse. A second condition for correct model selection is

$$\min_{j \in I^*} |\theta_j^*| > 4C\sqrt{\frac{\log(2p^2/\delta)}{n}} \tag{2.8}$$

where the p^2 term replaces the p term from above because the regularization required for model selection is larger than for good prediction. Intuitively, this condition implies that mixture weights must be above the noise level given by the square root term. Under these conditions, the result states

$$p(I_{\hat{\theta}} = I^*) \geq 1 - 2\delta(1 + \frac{1}{p}). \tag{2.9}$$

Note that the true coefficient vector and its cardinality are unknown so that these conditions are generally untestable in practice. The proof of this result also relies on the application of a concentration inequality.

2.2. *Generalization Error of Dictionary Learning.* Some theoretical work has studied the generalization error of dictionary learning. We briefly discuss some results from [10] who derive generalization bounds for both ℓ_1 and ℓ_0 penalization for arbitrary loss functions. Denote the approximation error of $y \in \mathbb{R}^m$ by

$$h_{A,B}(y) = \min_{\theta \in A} ||y - B\theta||.$$

We focus on the case where A defines an ℓ_1-constraint

$$A = \{\theta \in \mathbb{R}^p : ||\theta||_1 \leq \omega\}.$$

Note that this representation is equivalent to the one where the ℓ_1-penalty appears directly in the loss function to be minimized. In general the dictionary learning problem can be seen as minimizing

$$E(B) = E\left[h_{A,B}(y)\right] \tag{2.10}$$

where the expectation is taken with respect to the unknown density $y \sim f$. If a dictionary is learned from n observations $y_1, ..., y_n$, then the goal is to bound the generalization error, ϵ, that arises from learning a dictionary with a finite sample

$$E(B) \leq (1+\eta)E_n(B) + \epsilon \tag{2.11}$$

where $\eta \geq 0$. Using a covering number argument for $h_{A,B}$, a bound for ϵ on the order of $O(\sqrt{mp\log(n\omega)/n})$ is derived. The result implies that the generalization error shrinks with the sample size and has only a logarithmic dependence on the ℓ_1-norm of the coefficient vector.

3. Application to Topic Modeling. Topic modeling is a technique for text analysis that reduces a large collection of documents into groups of words with relatively distinct meanings. Such a procedure can be useful in text analysis for exploring the structure of a corpus or as a preliminary step to higher-level classification procedures. Latent Dirichlet allocation (LDA) [3] is perhaps the most widely studied model and the basis for many extensions. The basic LDA model can be described as a mixed-membership model. In particular the collection of documents is modeled by a shared collection of topics, represented by multinomial distributions over words, and each document exhibits topics with different proportions.

Sparsity is a key feature of topic models. It captures the intuitive notion that a given document should exhibit only a few of the corpus-wide topics that emerge. LDA is a Bayesian hierarchical model and achieves sparsity in topic proportions by introducing a sparse Dirichlet prior. An alternative way to achieve sparsity might utilize explicit ℓ_1-penalization using the sparse coding framework. As discussed above, this is useful because the theoretical literature has produced general results on the performance of sparse estimation procedures.

3.1. *Terminology.* The following terminology is widely used in the topic modeling literature and will be useful for our discussion. We will try to remain consistent with the notation used in the previous sections:

- A *vocabulary* is represented by a one-to-one mapping from the set of indices $\{1, ..., m\}$ to a set of words in the colloquial sense. Here

m represents the number of words in the vocabulary. Recall that m previously corresponded to the dimension of the observed data.

- A *document* is represented by a collection of word counts for each word, $w = (w_1, ..., w_m)$, where w_i indicates the number of occurrences of word i. This is referred to as a bag-of-words representation.
- A *corpus* is represented by a collection of documents, $D = (w_1, ..., w_n)$. Here n represents the number of documents in the corpus. Recall that n previously corresponded to the number of observations.
- A *topic* is represented by a multinomial distribution β_i over the vocabulary, alternatively a vector in the m-dimensional simplex. We will specify the number of topics, p, to be fit. The topics will be stored as an $m \times p$ matrix, $\beta = [\beta_1, ..., \beta_p]$.

3.2. *Latent Dirichlet Allocation.* The basic LDA model is a Bayesian hierarchical model. In this discussion let $w = (w_1, .., w_N)$ be a sequential representation so that each w_i corresponds to a word in the vocabulary, where N represents the number of words in the document. Consider the following generative process for a given document:

1. Choose the number of words $N \sim \text{Poisson}(\xi)$.
2. Choose topic proportions $\theta \sim \text{Dirichlet}(\alpha)$.
3. For each word w_i,
   (a) Choose a topic $z_i \sim \text{Multinomial}(\theta)$.
   (b) Choose a word $w_i \sim p(w_i|\beta_{z_i})$.

Here θ, z and the β's are latent variables to be estimated. Since N is independent of these variables, we ignore it for the purposes of this discussion. The parameter α for the Dirichlet prior will need to be specified. Note that there is an implicit assumption of exchangeability for words, called the "bag-of-words" assumption, since the specific sequence of words is ignored. Topics and documents are also assumed to be exchangeable, although these assumptions can be relaxed.

Fix the topics for now. To conduct inference for θ and z, we require their posterior distribution

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)}. \tag{3.1}$$

The generative process formulates the joint probability in the numerator as

$$p(\theta, z, w) = p(\theta|\alpha) \prod_{i=1}^{N} p(z_i|\theta) p(w_i|z_i, \beta). \tag{3.2}$$

Marginalizing over the latent variables gives

$$\begin{aligned} p(w|\alpha,\beta) &= \int_{\triangle} p(\theta|\alpha)\left(\prod_{i=1}^{N}\sum_{z_i=1}^{m} p(z_i|\theta)p(w_i|z_i,\beta)\right)d\theta \\ &= \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)}\int_{\triangle}\left(\prod_{j=1}^{p}\theta_j^{\alpha_j-1}\right)\left(\prod_{i=1}^{N}\sum_{j=1}^{p}\prod_{k=1}^{m}(\theta_j\beta_{jk})^{1(w_i=k)}\right)d\theta \end{aligned}$$

where the integral is taken over the $(p-1)$ simplex, $\triangle = \{\theta : \sum_{j=1}^{p}\theta_j = 1\}$. This is computationally intractable since there is a coupling between θ and β, so approximate methods such as Markov chain Monte Carlo or variational approximation must be used.

Variational approximation involves an iterative expectation-maximization (EM) algorithm. In the E-step, a lower bound for the log likelihood of a document is derived using Jensen's inequality to deal with the intractability of the likelihood. In particular, a computationally convenient variational distribution

$$q(\theta, z|\gamma,\phi) = q(\theta|\gamma)\prod_{i=1}^{N} q(z_n|\phi_n) \tag{3.3}$$

is used to approximate $p(\theta, z|w,\alpha,\beta)$. Optimizing over the variational parameters gives

$$(\gamma^*,\phi^*) = \underset{(\gamma,\phi)}{\operatorname{argmin}}\ \mathrm{D}(q(\theta,z|\gamma,\phi)\ ||\ p(\theta,z|w,\alpha,\beta)) \tag{3.4}$$

where $D(f\ ||\ g)$ denotes the Kullback-Leibler divergence. In the M-step, the resulting lower bound is maximized with respect to the parameters α and β.

3.3. *Sparse Coding Approach.* The sparse coding framework discussed in Sections 1 and 2.1 seems to provide a natural alternative to the Bayesian formulation of LDA. In particular the set of topics corresponds to a dictionary and document-level topic proportions correspond to codes. Thus, we can adapt a sparse coding formulation to specify an alternative sparse coding topic model (SCTM). Using the standard ℓ_1-penalty, we solve the following sparse coding problem:

$$\min_{\{\theta^{(i)}\}_{i=1}^{n},\{b_j\}_{j=1}^{p}} \sum_{i=1}^{n}\left(\frac{1}{2}||x^{(i)} - \sum_{j=1}^{p}\theta_j^{(i)}b_j||_2^2 + \omega||\theta^{(i)}||_1\right) \tag{3.5}$$

such that

$$\theta^{(i)} \succeq 0 \text{ for all } i \text{ and } b_j \succeq 0, \sum_k b_{jk} = 1 \text{ for all } j \tag{3.6}$$

where $x^{(i)}, i = 1, ..., n$ represents the observed word frequencies for the ith document. The θ's correspond to document-specific topic proportions and are constrained to be non-negative for interpretative purposes. We cannot constrain them to have unit sum due to the ℓ_1-penalty, so these topic proportions are not mixture coefficients as they were in LDA. Note that the theoretical work in [4], as discussed previously, requires no such constraints for predictive performance or model selection consistency in the coding step. The b's represent the dictionary elements to be learned and correspond to topics. To be interpreted as multinomial distributions over words, each topic is constrained to be nonnegative and have unit sum. As a result our model approximates the distribution over words for a document as a mixture. The probability for the ith word is given by

$$f(w_i) = \sum_{j=1}^{p} \theta_j f_j(w_i) \tag{3.7}$$

where $f_j(w_i) \equiv b_{ji}$ the ith entry in the vector b_j. We use an iterative procedure to fit the model, solving for document-level coefficients $\theta^{(i)}$ while holding the corpus-level topics $[b_1, ..., b_p]$ fixed, and vice versa.

3.3.1. *Solving for Coefficients.* The objective in (3.5) becomes separable in i when maximizing with respect to coefficients so we can consider a single document at a time. For a given document this requires the solution of a LASSO problem with nonnegative constraints on the coefficients when the topics are held fixed. The numerical solution of such problems has been well-studied and algorithms such as coordinate descent and least angle regression (LARS) work well for the standard LASSO.

We adopt a coordinate descent algorithm with soft thresholding. Denote the objective by $L(\theta) = g(\theta) + \omega||\theta||_1$. Let $g_k(\theta)$ denote the partial derivative of g with respect to θ_k and let $\theta^{(0)}_{-k} = \theta - \theta_k \boldsymbol{e_k}$ denote the vector θ with kth coordinate set to 0. For a given regularization parameter, θ, the algorithm takes the form:

1. Choose a direction $k \in \{1, .., p\}$.
2. If $|g_k(\theta^{(0)}_{-k})| < \omega$ then set $\theta = \theta^{(0)}_{-k}$. Otherwise obtain θ by a one-dimensional line search along the kth coordinate.
3. Iterate until convergence.

We handle the nonnegativity constraint by limiting the line search to nonnegative coordinates. LARS can also be adapted to handle the nonnegativity constraint as described in [6].

3.3.2. *Solving for Topics.* We now consider maximizing (3.5) with respect to topics. Here it will be helpful to adopt a matrix representation. Let $B = [b_1, ..., b_p] \in \mathbb{R}^{m \times p}$ be the dictionary formed by stacking the topics into a matrix. Form $X = [x_1, .., x_n] \in \mathbb{R}^{m \times n}$ and $\theta = [\theta_1, ..., \theta_n] \in \mathbb{R}^{p \times n}$. Holding θ's fixed, the problem becomes

$$\hat{B} = \underset{B}{\operatorname{argmin}} \frac{1}{2} ||X - B\Theta||_F^2 \tag{3.8}$$

such that

$$\sum_{i=1}^{d} B_{ij} = 1 \text{ for all } j = 1, ..., p \text{ and } B \succeq 0. \tag{3.9}$$

This is a quadratic program with the columns of B constrained to lie in the d-dimensional simplex and can be solved with a projected gradient descent algorithm. This approach is easily extendable to an online setting as in [8].

We iterate over columns since the simplex constraint is separable with respect to the columns. In particular we take a step along the gradient with respect to one column at a time and project to the simplex, which can be done using a linear-time algorithm proposed by [5]. This gives an update equation of the form

$$b_j = \Pi_{\triangle} \left(b_j - \eta \frac{\partial \mathcal{L}}{\partial b_j} \right) \tag{3.10}$$

where $\Pi_{\triangle}(\cdot)$ denotes a projection onto the simplex. The gradient is given by

$$\frac{\partial \mathcal{L}}{\partial b_j} = Bc_j - d_j \tag{3.11}$$

where $C = [c_1, ..., c_p] = \theta\theta^T$ and $D = [d_1, ..., d_p] = X\theta^T$. Finally, the step size can be tuned using a backtracking line search procedure or simply by taking the inverse of the Hessian given by

$$\frac{\partial \mathcal{L}^2}{\partial b_j \partial b_j} = C_{jj} I. \tag{3.12}$$

Note that taking steps one column at a time, as opposed to to the whole matrix at once, seems to help with convergence. To summarize, our projected gradient descent algorithm is as follows:

1. Calculate $C = \theta\theta^T$ and $D = X\theta^T$.
2. For each topic, perform the following update:
   (a) $\boldsymbol{u}_j \leftarrow b_j - \frac{1}{C_{jj}}(Bc_j - d_j)$
   (b) $b_j \leftarrow \Pi_{\triangle}(\boldsymbol{u}_j)$
3. Iterate until convergence.

3.3.3. *Other Considerations.* A number of other practical considerations must be addressed in fitting the model. First, the proportions and topics must be initialized. One difficulty is that we may approach a local minimimum when some of the topics become equal or nearly equal to each other. Indeed, we have found experimentally that a good initialization is important for learning useful topics. One reason for this may be that the simplex constraint on the dictionary can be very unforgiving, since the projection onto the simplex induces sparsity. Thus early steps can result in words that have zero probability in many topics. We follow [3] in selecting a small number of seed documents for each topic and smoothing across the entire vocabulary. The smoothing is done by drawing from a Dirichlet distribution with the seeded counts as parameters.

Second, we must choose an appropriate regularization parameter, which directly affects the sparsity of the topic proportions. This can be done via cross-validation and adapted to the specific task for which the topics will be used. Topic models are often evaluated based on held-out log likelihood or held-out perplexity, a likelihood-based measure of model fit. Note that we could choose a separate regularization parameter for each document, although we do not do so due to speed considerations. A simple heuristic is to run a coding update for a single document using the initialized topics and pick a regularization parameter which produces a suitable number of nonzero coefficients.

Third, we must choose the number of topics to be fit. Again, the most straightforward procedure is to use some kind of cross-validation procedure. Bayesian nonparametric extensions of LDA are able to deal with this issue more naturally using, for example, hierarchical Dirichlet processes. We do not explore analogous approaches in this paper.

3.4. *Comparison to LDA.* Qualitatively, one might view the sparse coding approach to topic modeling as a frequentist analogue to LDA. As a result we expect that both methods should output relatively similar topics, depending on the choice of parameters. Connections between LDA and non-probabilistic approaches to topic modeling involving matrix decompo-

sitions were noted in [3]. We also expect that sparse coding should allow for a potentially large number of topics.

Note that the SCTM does not contain a notion of topic assignment as in LDA. However, we can see the equivalence by marginalizing over topic assignments z to get

$$p(w_i|\theta, \beta) = \sum_j \theta_j \beta_{ij}$$

where β_{ij} denotes the probability of the ith word in the jth topic.

4. Numerical Results. We apply the SCTM described above to to a pre-processed collection of 1,500 full papers from the Neural Information Processing Systems (NIPS) Conference.[1] Papers come from fields such as computational neuroscience, machine learning and statistics. The preprocessing consisted of removal of stopwords, tokenization, and truncation of the vocabulary to words that occurred more than ten times. Nonetheless the original dataset contains some spurious words such as "a2i" or "aaa", so we further truncate the vocabulary to words that ocurred more than twenty-five times. The resulting dataset contains 6,727 unique words.

4.1. *Sample Output.* The output of the topic model includes topic proportions for each document and a set of topics for the entire corpus. With these, we can replicate some of the exploratory aspects of topic models illustrated for LDA [3, 1]. First, we can visualize topics by examining the most probable words. See Figure 4.1 for fifteen topics, visualized by their top five most probable words, from applying a SCTM with $p = 50$ topics and regularization parameter $\omega = 2^{-12}$ to the NIPS dataset. This parameter value was chosen using the simple heuristic previously discussed. Qualitatively, topics identify semantically related sets of words and the most probable words in a topic seem to correspond to keywords. For example the {*node, nodes, network, tree, graph*} topic seems to contain keywords used to describe network analysis. The {*bound, number, result, threshold, theorem*} topic seems to contain keywords that more generally capture theoretical concepts. This topic representation can be useful in itself as a dimensionality reduction tool for browsing a corpus.

We can use the topic proportions to examine the top documents that exhibit a given topic. Recall that topic proportions from the SCTM are not constrained in their sum due to the ℓ_1-penalty. To make direct comparisons we normalize them to have unit sum. For example, the top documents associated

[1] Newman, D. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/Bag+of+Words].

1	2	3	4	5
neuron	node	map	motion	distribution
synaptic	nodes	space	direction	gaussian
activity	network	feature	velocity	probability
synapses	tree	representation	moving	density
potential	graph	mapping	stage	mean
6	7	8	9	10
algorithm	signal	word	model	bound
gradient	noise	recognition	parameter	number
step	filter	speech	prediction	result
convergence	processing	character	modeling	threshold
update	component	speaker	structure	theorem
11	12	13	14	15
circuit	spike	training	neural	action
chip	information	error	network	policy
analog	rate	generalization	net	optimal
current	firing	test	recurrent	states
voltage	noise	performance	computation	step

FIG 1. *Fifteen topics from a sparse coding model applied to NIPS dataset with* $p = 50$ *total topics and regularization parameter* $\omega = 2^{-12}$*. Topics are visualized by their top five most probable words.*

with the {*distribution,gaussian,probability*} topic are "The Rectified Gaussian Distribution" and "Approximating Posterior Distributions in Belief Networks Using Mixtures". The top documents in the {*word,recognition,speech*} topic are "Comparison of Human and Machine Word Recognition" and "Performance Through Consistency: MS-TDNN's for Large Vocabulary Continuous Speech Recognition".

We can also look at similar documents based on the distance between their topic proportion vectors. This can be useful, for example, in document search or recommendation. Starting with the document "Polynomial Uniform Convergence of Relative Frequencies to Probabilities", the most similar documents based on Hellinger distance are "On the Distribution of Local Minima of a Random Function on a Graph" and "Mixture Density Estimation". All these papers contain a mix of theory and probability as captured by the {*bound, number, result*} and {*distribution, gaussian, probability*} topics.

4.2. *Generalization Performance.* We assess generalization performance relative to LDA by comparing held-out likelihood on a test set. In particular

we compute the perplexity score for a document, $w \in \mathbb{R}^m$ given by

$$\text{perplexity}(w) = \exp\left\{-\frac{\log\, p(w)}{N}\right\}$$

where N indicates the number of words in the document. Perplexity is commonly used in language modeling and can be interpreted as a per-word measure of likelihood. Lower perplexity indicates better generalization performance. For LDA, we use the *topicmodels* package in R which provides a wrapper for the C implementation of variational EM from [3]. We implement the SCTM in R. For SCTM the cross-validation involves learning topics from the training set, fitting topic proportions for test documents using LASSO, and finally computing the fitted probabilities for each word. The likelihood is calculated as

$$\log\, p(w) = \sum_{i=1}^{m} n_i \hat{p}(w_i)$$

where n_i indicates the number of times the ith word appeared in the test document and $\hat{p}(w_i) = \sum_{j=1}^{p} \hat{\theta}_j b_{ij}$.

Each model has input parameters other than the number of topics. To make a fair comparison, we run the models for multiple values of the parameters and select the best for each model. For LDA, the hyperparameter α specifies a symmetric Dirichlet prior for the topic proportions. We test for $\alpha \in \{1\times10^{-2}, 5\times10^{-2}, 1\times10^{-1}, 5\times10^{-1}\}$. Note that choosing $\alpha < 1$ encourages sparsity in the posterior topic proportions. For the SCTM, we choose the regularization parameter $\omega \in \{5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$.

Strangely, we find that for certain documents both models report very high perplexities (very low likelihoods). While LDA only reports very high perplexities and never infinite perplexities (zero likelihoods), SCTM occassionally does assign zero probability to words. This happens because we have induced sparsity in both the topic proportions and the topics themselves. In particular it is possible that an unlucky linear combination to lead to a zero fitted probability. In practice it may be necessary to smooth the topics so that all words in the vocabulary have positive probability. In LDA this can be implemented naturally by specifying a Dirichlet prior on the topics, introducing a second Dirichlet hyperparameter. In SCTM there is no such natural implementation, since the simplex constraint on the dictionary imposes sparsity, so the topics would have to be post-processed in some manner. For the purposes of this paper, we proceed by plotting the median and interquartile range.

See Figure 2 for results from a 10-fold cross-validation experiment. In this experiment we consider a randomly selected 500-document subset of

the NIPS corpus for speed purposes. The plot compares median held-out perplexities over the folds for LDA ($\alpha = 0.01$) and SCTM ($\omega = 5 \times 10^{-5}$). The results suggest that SCTM can be quite noisy for a small number of topics but that it performs better than LDA as the number of topics is increased. However, we should keep in mind that SCTM occassionally assigns zero probability to a word, leading to infinite perplexity. A simple fix is to lower the penalization when fitting the topic proportions for test documents in which words receive zero fitted probability. Future work should resolve this issue in a more natural way by exploring methods for smoothing the resulting topics or altering the model specification to eliminate this possibility.

It is interesting to note that while the median perplexity of LDA seems to level off as the number of topics increases, the perplexity of SCTM seems to continue falling. One possible reason is that sparse coding has been used to deal with overcomplete dictionaries, where $p > n$, and thus supports a much larger number of topics. Indeed the sparsity constraints may lead to very specialized topics which are more easily recombined for prediction purposes. Due to slow run times, we have not fit the model with more than 150 topics, though part of this is due to implementation and can easily be resolved in a performance-oriented language such as C. It would be interesting to see at what number of topics the held-out perplexity levels off. It would also be interesting to study the tradeoff, if there exists any, between interpretability of the topics and generalization performance.

Note that the choice of LDA hyperparameter does not affect the generalization performance much, perhaps because the posterior dominates the prior even with a training set of 450 documents. On the other hand the choice of SCTM regularization parameter can have a large effect for a small number of topics. In particular too much regularization can lead to too much sparseness, aggravating the zero likelihood issue. There is less difference for a large number of topics. See Figure 3 for an illustration.

5. Extensions. Various work has extended the basic LDA model to incorporate topical structure such as correlated topics [1] and dynamic topics [2]. Similar extensions can be made to the sparse coding approach by altering the penalization. Previous literature on structured sparsity has produced methods for inducing specific patterns of sparsity. We focus on extending the model to incorporate correlated topics.

The correlated topic model of [1] replaces the Dirichlet prior on the topic proportions with a logistic normal prior. In particular a multivariate normal

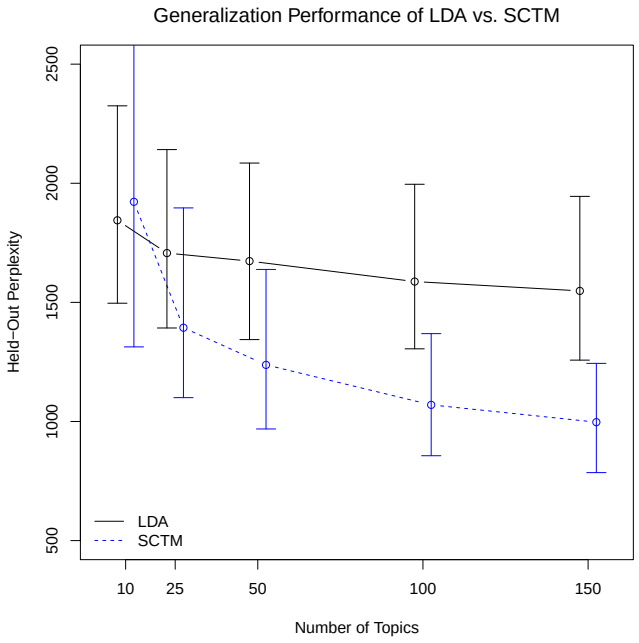

FIG 2. *Results from a 10-fold cross-validation experiment comparing held-out perplexities for LDA with* $\alpha = 0.01$ *and SCTM with* $\omega = 5 \times 10^{-5}$. *Note that points represent medians and error bars represent interquartile ranges.*

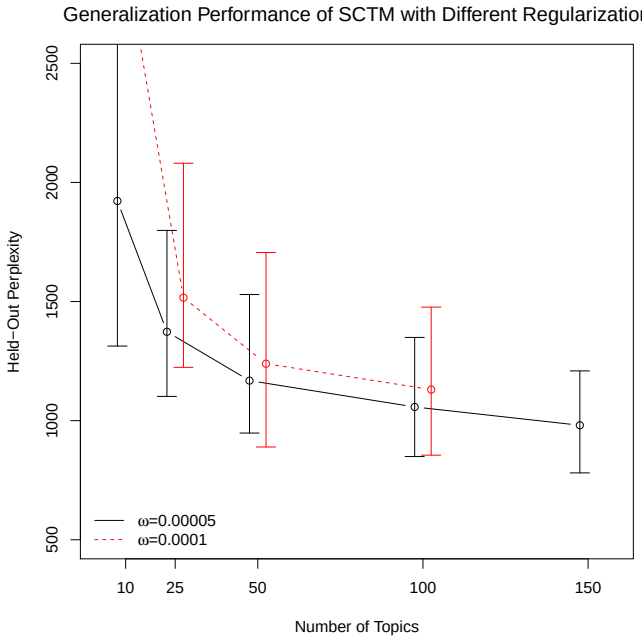

FIG 3. *A comparison of generalization for SCTM with different regularization parameters. For a small number of topics, too much penalization aggravates the zero likelihood issue.*

variable $\eta \sim N(\mu, \Sigma)$ is mapped to the simplex by

$$\theta = f(\eta) = \frac{\exp\{\eta\}}{\sum_j \exp\{\eta_j\}}. \tag{5.1}$$

Correlation between topics is captured by the covariance matrix Σ. The graph can be visualized using a variant of the graphical lasso procedure [7]. The graphical lasso produces a sparse estimate of $\Omega = \Sigma^{-1}$ and positive entries indicate the existence of an edge.

To implement a similar extension for sparse coding, we would like to analogously introduce a covariance matrix Σ governing the correlation between topics θ. To keep this covariance matrix sparse, we could also use graphical lasso to maximize a term of the form

$$\log \det(\Omega) - \mathrm{tr}(S\Omega) - \rho||\Omega||_1 \tag{5.2}$$

where S indicates the empircal covariance matrix given by $S = \frac{1}{n}\sum_{i=1}^n \theta_i\theta_i^T$. Note that this corresponds to a log-likelihood, partially maximized with respect to the mean, that arises from directly assuming θ is normal.

The objective becomes

$$\min_{\{\theta_i\}_{i=1}^n, B, \Omega} \sum_{i=1}^n \mathcal{L}(B, \theta_i) + \theta^T\Omega\theta - \log\det(\Omega) + \rho||\Omega||_1 \tag{5.3}$$

where

$$\mathcal{L}(B, \theta_i) = \frac{1}{2}||y_i - B\theta_i||_2^2 + \omega||\theta_i||_1.$$

This could be solved via coordinate descent, optimizing with respect to θ_i's, B and Ω separately. Optimizing with respect to B would remain the same as before. Optimizing with respect to Ω can be done using the graphical lasso procedure from [7]. Finally optimizing with respect to θ_i's can be done by rewriting the term involving Ω to get

$$\mathcal{L}(B, \theta_i) + \theta_i^T\Omega\theta_i = ||\begin{bmatrix} y_i \\ 0 \end{bmatrix} - \begin{bmatrix} B \\ L^T \end{bmatrix}\theta_i||_2^2 + \omega||\theta_i||_1 \tag{5.4}$$

where $\Omega = LL^T$ is a Cholesky decomposition of the inverse covariance matrix. This now becomes a standard LASSO problem that can be solved as before. Unfortunately, we have not been able to implement this procedure so that it converges. In particular there may be a difficulty because it implicitly assigns a normal prior to the θ_i's rather than an intermediate variable which is then normalized to the simplex.

6. Conclusion. In this paper, we have reviewed the technique of sparse coding and some theoretical issues surrounding both the coding step and the dictionary learning step. Theoretical work on the joint procedure is somewhat limited, though some recent results [9] have addressed the noiseless case. We have also reviewed the technique of topic modeling and presented an application of sparse coding to topic modeling. The sparse coding topic model (SCTM) can be seen as a frequentist analogue to the standard Bayesian model, Latent Dirichlet allocation (LDA). SCTM provides comparable performance to LDA and seems to support a larger number of topics. The algorithm also can be extended to an online setting as in [8]. Lastly, the model can be extended using norms for structured sparsity to, for example, replicate the correlated topic model of [1].

Acknowledgements. I thank Professor John Lafferty for his guidance throughout the writing of this paper. I also thank Andy Dahl for helpful discussions.

References.

[1] Blei, D. and Lafferty, J. (2007). A Correlated Topic Model of Science. *Annals of Applied Statistics* **1**(1): 17 – 35.

[2] Blei, D. and Lafferty, J. (2006). Dynamic Topic Models. In *Proceedings of the International Conference on Machine Learning*.

[3] Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research* **3**: 993–1022.

[4] Bunea, F., Tsybakov, A., Wegkamp, M., and Barbu, A. (2010). SPADES and Mixture Models. *Annals of Statistics* **38**(5): 2525–2558.

[5] Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). Efficient Projections Onto the ℓ_1-ball for Learning in High Dimensions. In *Proceedings of the International Conference on Machine Learning*.

[6] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least Angle Regression. *Annals of Statistics* **32**(2): 407–499.

[7] Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse Covariance Estimation with the Graphical Lasso. *Biostatistics* **9**: 432–441.

[8] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research* **11**: 19–60.

[9] Spielman, D., Wang, H., and Wright, J. (2012). Exact Recovery of Sparsely-Used Dictionaries. In *Proceedings of the 25th Annual Conference on Learning Theory*.

[10] Vainsencher, D., Mannor, S., and Bruckstein, A. (2011). The Sample Complexity of Dictionary Learning. *Journal of Machine Learning Research* **12**: 3259–3281.