SPARSE MACHINE LEARNING METHODS FOR UNDERSTANDING LARGE TEXT CORPORA

LAURENT EL GHAOUI*, GUAN-CHENG LI*, VIET-AN DUONG**, VU PHAM***,
ASHOK SRIVASTAVA****, AND KANISHKA BHADURI****

Abstract. Sparse machine learning has recently emerged as powerful tool to obtain models of high-dimensional data with high degree of interpretability, at low computational cost. This paper posits that these methods can be extremely useful for understanding large collections of text documents, without requiring user expertise in machine learning. Our approach relies on three main ingredients: (a) multi-document text summarization and (b) comparative summarization of two corpora, both using sparse regression or classification; (c) sparse principal components and sparse graphical models for unsupervised analysis and visualization of large text corpora. We validate our approach using a corpus of Aviation Safety Reporting System (ASRS) reports and demonstrate that the methods can reveal causal and contributing factors in runway incursions. Furthermore, we show that the methods automatically discover four main tasks that pilots perform during flight, which can aid in further understanding the causal and contributing factors to runway incursions and other drivers for aviation safety incidents.

1. Introduction

Sparse machine learning refers to a collection of methods to learning that seek a trade-off between some goodness-of-fit measure and sparsity of the result, the latter property allowing better interpretability. In a sparse learning classification task for example, the prediction accuracy or some other classical measure of performance is not the sole concern: we also wish to be able to explain what the classifier means to a non-expert. Thus, if the classification task involves say gene data, one wishes to provide not only a high-performance classifier, but one that only involves a few genes, allowing biologists to focus their research efforts on those specific genes.

There is an extensive literature on the topic of sparse machine learning, with terms such as compressed sensing [12, 5], l_1-norm penalties and convex optimization [42], often associated with the topic. Successful applications of sparse methods have been reported, mostly in image and signal processing, see for example [15, 28, 31]. Due to the intensity of research in this area, and despite an initial agreement that sparse learning problems are more computationally difficult than their non-sparse counterparts, many very efficient algorithms have been developed for sparse machine learning in the recent past. A new consensus might soon emerge that sparsity constraints or penalties actually *help* reduce the computational burden involved in learning.

Our paper makes the claim that sparse learning methods can be very useful to the *understanding* large text databases. Of course, machine learning methods in general have already been successfully applied to text classification and clustering, as evidenced for example by [21]. We will show that sparsity is an important added property that is a crucial component in any tool aiming at providing interpretable statistical analysis, allowing in particular efficient multi-document summarization, comparison, and visualization of huge-scale text corpora.

*EECS Dept., UC Berkeley, (guanchengli,elghaoui)@eecs.berkeley.edu;
**Ecole des Mines d'Alès, School of Production & Systems Engineering, viet-an.duong@mines-ales.org;
***University of Science, VNU-HCMC, Ho-Chi-Minh City, Vietnam, ptvu@acm.org;
****System-Wide Safety and Assurance Technologies Project, NASA,
(ashok.n.srivastava,kanishka.bhaduri-1)@nasa.gov.

To illustrate our approach we focus here on Aviation Safety Reporting System (ASRS) text reports, which is a crucial component of the continuing effort to maintain and improve aviation safety. The text reports are written by members of the flight crew, air traffic controllers, and others on a voluntary basis. The reports are de-identified so that the author and other specific information regarding the flight is not revealed. Each report is a small paragraph describing any incident that the author wishes to discuss and is assigned a category among a set of pre-defined ones by a team of ASRS experts. The ASRS database consists of about 100,000 reports spanning approximately 30 years. Although the report intake fluctuates on a monthly basis, the ASRS report intake for March 2011 was 6148 reports. ASRS data are used by experts to identify deficiencies in the National Aviation System so that they can be corrected. The data are also used to further deepen our understanding of human factors issues in aviation which is a critical component of aviation safety. It is widely thought that over two-thirds of all aviation accidents and incidents have their roots in human performance errors [1].

The ASRS data contains several of the crucial challenges involved under the general banner of "large-scale text data understanding". First, its scale is huge, and growing rapidly, making the need for automated analyses of the processed reports more crucial than ever. Another issue is that the reports themselves are far from being syntactically correct, with lots of abbreviations, orthographic and grammatical errors, and other shortcuts. Thus we are not facing a corpora with well-structured language having clearly defined rules, as we would if we were to consider a corpus of laws or bills or any other well-redacted data set. Finally, in many cases we do not know in advance what to look for because the goal is to discover precursors to aviation safety incidents and accidents. In other words, the task is not about search, and finding a needle in a haystack: in many cases, we cannot simply to monitor the emergence or disappearance of a few keywords that would be known in advance. Instead the task resembles more one of trying to visualize the haystack itself, compare various parts of it, or summarize some areas.

In examining the ASRS data, we would like to be able to pinpoint some emerging issues, highlight some trends, broken down by time, type of flight, incident, or airport. For example, the class of incidents known as "runway incursion" might occur more frequently at some airports; runway incursions might be due to different causes, necessitating differentiated responses (such as improved ground lighting, or changes in taxiway configurations). How can we quickly figure out the *type* of runway incursions involved at each airport, and respond accordingly?

Our paper is organized as follows. Section 2 is devoted to a review of some of the main models and algorithms in sparse machine learning. We explain how these methods can be used in text understanding in section 3. Section 4 illustrates the approach in the context of ASRS data, and also reviews some prior work on this specific data set. Although our focus here is on ASRS data, most of the approaches depicted here have been developed in the context of news data analysis, see [18, 30, 6].

2. Sparse Learning Methods

In this section we review some of the main algorithms of sparse machine learning.

2.1. **Sparse classification and regression.**

2.1.1. *The LASSO.* Perhaps the most well known example of sparse learning is the variant of least-squares known as the LASSO [41], which takes the form

$$\min_{\beta} \|X^T\beta - y\|_2^2 + \lambda\|\beta\|_1, \tag{1}$$

where X is a $n \times m$ data matrix (with each row a specific feature, each column a specific data point), y is a m-dimensional response vector, and $\lambda > 0$ is a parameter. The l_1-norm penalty encourages

[1] See `http://asrs.arc.nasa.gov` for more information on the ASRS system. The text reports are available on this website along with analyses performed by the ASRS analysts.

the regression coefficient vector β to be sparse, bringing interpretability to the result. Indeed, if each row is a feature, then a zero element in β at the optimum of (1) implies that that particular feature is absent from the optimal model. If λ is large, then the optimal β is very sparse, and the LASSO model then allows to select the few features that are the best predictors of the response vector.

2.1.2. *Solving the LASSO.* The LASSO problem looks more complicated than its classical least-squares counterpart. However, there is mounting evidence that, contrary to intuition, the LASSO is substantially *easier* to solve than least-squares, at least for high values of λ. As shown later, in typical applications to text classification, a high value of λ is desired, which is precisely the regime where the LASSO is computationally very easy to solve.

Many algorithms have been proposed for LASSO; at present it appears that, in text applications with sparse input matrix X, a simple method based on minimizing the objective function of (1) one coordinate of β at a time is extremely competitive [16, 33]. The so-called safe feature elimination procedure [14], which allow to cheaply detect that some of the components of β will be zero at optimum, enables to treat data sets having millions of terms and documents, at least for high values of λ.

2.1.3. *Other loss functions.* Similar models arise in the context of support vector machines (SVM) for binary classification, where the sparse version takes the form

$$\min_{\beta,b} \frac{1}{m}\sum_{i=1}^{m} h(y_i(x_i^T\beta + b)) + \lambda\|\beta\|_1, \tag{2}$$

where now y is the vector of ± 1's indicating appartenance to one of the classes, and h is the so-called hinge loss function, with values $h(t) = \max(0, 1-t)$. At optimum of problem (2), the above model parameters (β, b) yield a classification rule, *i.e.* predict a label $\hat{y}$ for a new data point x, as follows: $\hat{y} = \mathbf{sign}(x^T\beta + b)$. A smooth version of the above is sparse logistic regression, which obtains upon replacing the hinge loss with a smooth version $l(t) = \log(1+e^{-t})$. Both of these models are useful but somewhat less popular than the LASSO, as state-of-the-art algorithms are have not yet completely caught up. For our text applications, we have found that LASSO regression, although less adapted to the binary nature of the problem, is still very efficient [30].

2.2. **Sparse principal component analysis.**

2.2.1. *The model.* Sparse principal component analysis (Sparse PCA, see [48, 47] and references therein) is a variant of PCA that allows to find sparse directions of high variance. The sparse PCA problem can be formulated in many different ways, one of them (see [39, 27]) involves a low-rank approximation problem where the sparsity of the low-rank approximation is penalized:

$$\min_{p,q} \|M - pq^T\|_F^2 + \lambda\|p\|_1 + \mu\|q\|_1, \tag{3}$$

where M is the data matrix, $\|\cdot\|_F$ is the Frobenius norm, and $\mu \geq 0, \lambda \geq 0$ are parameters.

The model above results in a rank-one approximation to M (the matrix pq^T at optimum), and vectors p, q are encouraged to be sparse due to the presence of the l_1 norms, with high value of the parameters λ, μ yielding sparser results. Once sparse solutions are found, then the rows (resp. columns) in M corresponding to zero elements in p (resp. in q) are removed, and problem (3) is solved with the reduced matrix as input. If M is a term-by-document matrix, the above model provides sparsity in the feature space (via p) and the document space (via a "topic model" q), allowing to pinpoint a few features and a few documents that jointly "explain" data variance.

2.2.2. *Algorithms.* Several algorithms have been proposed for the above problem, for example [23, 39, 8]. In practice, one algorithm that is very efficient (although it is only guaranteed to converge to a local minimum) consists in solving the above problem alternatively over p, q many times [39]. This leads to a modified power iteration method

$$p \to P(S_\lambda(Mq)), \quad q \to P(S_\mu(M^T q)),$$

where P is the projection on the unit circle (assigning to a non-zero vector v its scaled version $v/\|v\|_2$), and for $t \geq 0$, S_t is the "soft thresholding" operator (for a given vector v, $S_t(v) = \mathbf{sign}(v)\max(0, |v| - t)$, with operations acting component-wise). We can replace the soft thresholding by hard thresholding, for example zeroing out all but a fixed number of the largest elements in the vector involved.

With $\lambda = \mu = 0$ the original power iteration method for the computation of the largest singular value of M is recovered, with optimal p, q the right- and left- singular vectors of M. The presence of λ, μ modifies these singular vectors to make them sparser, while maintaining the closeness of M to its rank-one approximation. The hard-thresholding version of power iteration scales extremely well with problem size, with greatest speed increases over standard power iteration for PCA when a high degree of sparsity is asked for. This is because the vectors p, q are maintained to be extremely sparse during the iterations.

2.2.3. *Thresholded PCA.* An alternative to solving the above that was proposed earlier for sparse PCA is based on solving a classical PCA problem, then thresholding the resulting singular vectors so that they have the desired level of sparsity. For large-scale data, PCA is typically solved with power iteration, so the "thresholded PCA" algorithm is very similar to the above thresholded power iteration for sparse PCA. The only difference is in how many times thresholding takes place. Note that in practice, the thresholded power iteration for sparse PCA is much faster than its plain counterpart, since we are dealing with much sparser vectors as we perform the power iterations.

2.3. **Sparse graphical models.**

2.3.1. *Covariance selection.* Sparse graphical modeling seeks to uncover a graphical probabilistic model for multivariate data that exhibits some sparsity characteristics. One of the main examples of this approach is the so-called sparse covariance selection problem, with a Gaussian assumption on the data (see [34], and related works such as [17, 29, 45, 40, 26, 24]). Here we start with a $n \times n$ sample covariance matrix S, and assuming the data is Gaussian, formulate a variant to the corresponding maximum likelihood problem:

$$\max_X \log\det X - \mathbf{Tr}SX - \lambda\|X\|_1, \tag{4}$$

where $\lambda > 0$ is a parameter, and $\|X\|_1$ denotes the sum of the absolute values of all the entries in the $n \times n$ matrix variable X. Here, $\mathbf{Tr}SX$ is the scalar product between the two symmetric matrices S and X, that is, the sum of the diagonal entries in the matrix product SX. When $\lambda = 0$, and assuming S is positive-definite, the solution is $X = S^{-1}$. When $\lambda > 0$, the solution X is always invertible (even if S is not), and tends to have many zero elements in it as λ grows. A zero element in the (i, j) entry of X corresponds to the conditional independence property between nodes i and j; hence sparsity of X is directly related to that of the conditional independence graph, where the absence of an edge denotes conditional independence.

2.3.2. *Solving the covariance selection problem.* The covariance selection problem is much more challenging than its classical counterpart (where $\lambda = 0$), which simply entails inverting the sample covariance matrix. At this point it appears that one of the most competitive algorithms involves solving the above problem one column (and row) of X at a time. Each sub-problem can be interpreted as a LASSO regression problem between one particular random variable and all the others [34, 17]. Successful applications of this approach include Senate voting [34] and gene data analysis [34, 11]

Just as in the PCA case, there is a conceptually simple alorithm, which relies on thresholding. If the covariance matrix is invertible, we simply invert it and threshold the elements of the inverse. Some limited evidence points to the statistical superiority of the sparse approach (based on solving problem (4)) over its thresholded counterpart. On the computational front however, and contrarily to the models discussed in the previous two sections, the thresholding approach remains computationally competitive, although still very challenging in the high-dimensional case.

2.4. **Thresholded models.** The algorithms in sparse learning are built around the philosophy that sparsity should be part of the model's formulation, and not produced as an afterthought. Sparse modeling is based on some kind of direct formulation of the original optimization problem, involving, typically, an l_1 penalty. As a result of the added penalty, sparse models have been originally thought to be substantially more computationally challenging than their non-penalized counterparts.

In practice, sparse results can be obtained via the use of *any* learning algorithm, even one that is not necessarily sparsity-inducing. Sparsity is then simply obtained via thresholding the result. This is the case for example with naïve Bayes classification, or Latent Dirichlet Allocation (LDA). In the case of LDA, the result is a probability distribution on all the terms in the dictionary. Only the terms with the highest weights are retained, which amounts in effect to threshold the probability distribution. The notion of *thresholded models* refers to the approach of applying a learning algorithm and obtaining sparsity with a final step of thresholding.

The question about which approach, "direct" sparse modeling or sparse modeling via thresholding, works better in practice, is a natural one. Since direct sparse modeling appears to be more computationally challenging, why bother? Extensive research in the least-squares case shows that thresholding is actually often sub-optimal [30]. Similar evidence has been reported on the PCA case [47]. Our own experiments in section 4 support this viewpoint.

There is an added benefit to direct sparse modeling—a computational one. Originally thresholding was considered as a computational shortcut. As we argued above for least-squares, SVM and logistic regression, and PCA, sparse models can be actually surprisingly easier to solve than classical models; at least in those cases, there is no fundamental reason for insisting on thresholded models, although they can produce good results. For the case of covariance selection, the situation is still unclear, since direct sparse modeling via problem (4) is still computationally challenging.

The above motivates many researchers to "sparsify" existing statistical modeling methodologies, such as Latent Dirichlet Allocation [4]. Note that LDA also encodes a notion of sparsity, not in the feature space, but on the document (data) space: it assumes that each document is a mixture of a small number of topics, where the topic distribution is assumed to have a Dirichlet prior. Thus, depending on the concentration parameter of this prior, a document comprised of a given set of words may be effectively restricted to having a small number of topics.

This notion of sparsity (document-space sparsity) does not constrain the number of features active in the model, and does not limit overall model complexity. As a result, in LDA, the inclusion of terms that have little discrimination power between topics (such as 'and', 'the', etc.) may fall into multiple topics unless they are eliminated by hand. Once a set of topics is identified the most descriptive words are depicted as a list in order of highest posterior probability given the topic. As with any learning method, thresholding can be applied to this list to reveal the top most descriptive words given a topic. It may be possible to eliminate this thresholding step using a modified objective function with an appropriate sparsity constraint. This is an area of very active research, as evidenced by [13].

3. Application to Text Data

3.1. **Topic summarization.** Topic summarization is an extensive area of research in natural language processing and text understanding. For a recent survey on the topic, see [7]. There are many instances of this problem, depending on the precise task that is addressed. For example the focus could be to summarize a single unit of text, or summarize multiple documents, or summarize two classes of documents in order to produce the summaries that offer the best contrast. Some further references to summarization include [19, 20, 32].

The approach introduced in [18] and [30] relies on LASSO regression to produce a summary of a particular topic as treated in multiple documents. This is part of the *extraction* task within a summarization process, where relevant terms are produced and given verbatim [7]. Using predictive models for topic summarization has a long history, see for example [37]; the innovation is the systematic reliance on *sparse* regression models.

The basic idea is to divide the corpora in two classes, one that corresponds to the topic, and the other to the rest of the text corpora. For example, to provide the summary of the topic "China" in a corpora of news articles from *The New York Times* over a specific period, we may separate all the paragraphs that mention the term "china" (or related terms such as "chinese", "china's", etc) from the rest of the paragraphs. We then form a numerical, matrix representation X (via, say, TF-IDF scores) of the data, and form a "response" vector (with 1's if the document mentions China and -1 otherwise). Solving the LASSO problem (1) leads to a vector β of regressor coefficients, one for each term of the dictionary. Since LASSO encourages sparsity, many elements of β are zero. The non-zero elements point to terms in the dictionary that are highly predictive of the appearance of "china" in any paragraph in the corpus.

The approach can be used to contrast to set of documents. For example, we can use it to highlight the terms that allow to best distinguish between two authors, or two news sources on the same topic.

Topic summarization is closely related to *topic modeling* via Latent Dirichlet Allocation (LDA) [4], which finds on a latent probabilistic model to produce a probability distribution of all the words. Once the probability distribution is obtained, the few terms that have the highest probability are retained, to produce some kind of summary in an unsupervised fashion. As discussed in section 2.4, the overall approach can be seen as a form of indirect, thresholding method for sparse modeling.

3.2. **Discrimination between several corpora.** Here the basic task is to find out what terms best describe the differences between two or more corpora. In a sparse classification setting, we may simply classify one of the corpora against all the others. The resulting classifier weight vector, which is sparse, then points to a short list of terms that are most representative of the salient differences between the corpora and all the others. Of course, related methods such as multi-class sparse logistic regression can be used.

3.3. **Visualization and clustering.** Sparse PCA and sparse graphical models can provide insights to large text databases. PCA itself is a widely used tool for data visualization, but as noted by many researchers, the lack of interpretability of the principal components is a challenge. A famous example of this difficulty involves the analysis of Senate voting patterns. It is well-known in political science that, in that type of data, the first two principal components explain the total variance very accurately [34]. The first component simply represents party affiliation, and accounts for a high proportion of the total variance (typically, 80%). The second component is much less interpretable.

Using sparse PCA, we can provide axes that are sparse. Concretely this means that they involve only a few features in the data. Sparse PCA thus brings an interpretation, which is given in terms of which few features explain most of the variance. Likewise, sparse graphical modeling can be very revealing for text data. Because it produces sparse graphs, it can bring an understanding as to which variables (say, terms, or sources, or authors) are related to each other and how.

4. Application to ASRS Data

4.1. **ASRS data sets.** In this section our focus is on reports from the Aviation Safety Reporting System (ASRS). The ASRS is a voluntary program in which pilots, co-pilots, other members of the flight crew, flight controllers, and others file a text report to describe any incident that they may have observed that has a bearing on aviation safety. Because the program is completely voluntary and the data are de-identified, meaning that the author, his or her position, the carrier, and other identifying information is not available in the report. After reports are submitted, analysts from ASRS may contact the author to obtain clarifications. However, the information provided by the reporter is not investigated further. This motivates the use of (semi-) automated methods for the real-time analysis of the ASRS data.

A first data set is the one used as part of the SIAM 2007 Text Mining Competition. The data consists in about 20,000 flight reports submitted by pilots after their flight. Each report is a small paragraph describing any incident that was recorded during flight, and is assigned a category (totaling 22), or type of incident. We refer to this data set as the "category" data set. In the

category data set, the airport names, the time stamps and other information has been removed. The documents in this corpora were processed through a language normalization program that performs stemming, acronym expansion, and other basic pre-processing. The system also removes non-informative terms such as place names.

We have also worked with an ASRS data set of raw reports that include airport names and contain the term "runway incursion". Our goal with this data set is to focus on understanding the causal factors in runway incursions, which is an event in which one aircraft moves into the path of another aircraft during landing or takeoff. A key question that arises in the study of runway incursions is to understand whether there are significant distinguishing features of runway incursions for different airports. Although runway incursions are common, the causes may differ with each airport. These are the causal factors that enable the design of the intervention appropriate for that airport, whether it may be runway design, runway lighting, procedures, etc. Unlike the category data set, these data were not processed through a language normalization program.

4.2. **Related work on ASRS data.** In this section we list some previous work in applying data mining/machine learning methods for analyzing ASRS data, along with pointers for further research.

Text Cube [25] and Topic Cube [46] are multi-dimensional data cube structures which provide a solid foundation for effective and flexible analysis of the multidimensional ASRS text database. The text cube structure is constructed based on the TF/IDF (i.e., vector space) model while the topic cube is based on a probabilistic topic model. Techniques have also been developed for mining repetitive gapped subsequences [9], multi-concept document classification [43][44], and weakly supervised cause analysis [1]. The work in [25] has been further extended in [10] where the authors have proposed a keyword search technique. Given a keyword query, the algorithm ranks the aggregations of reports, instead of individual reports. For example, given a query "forced landing" an analyst may be interested in finding the external conditions (e.g. weather) that causes this kind of query and also find other anomalies that might co-occur with this one. This kind of analysis can be supported through keyword search, providing an analyst a ranked list of such aggregations for efficient browsing of relevant reports. In order to enrich the semantic information in a multidimensional text database for anomaly detection and causal analysis, Persing and Ng have developed new techniques for text mining and causal analysis from ASRS reports using semi-supervised learning [36] and subspace clustering [3].

Some work has also been done on categorizing ASRS reports into anomalous categories. It poses some specific challenges such as high and sparse dimensionality as well as multiple labels per document. Oza et al. [35] presents an algorithm called Mariana which learns a one-vs-all SVM classifier per anomaly category on the bag-of-words matrix. This provides good accuracy on most of the ASRS anomaly categories.

Topic detection from ASRS datasets have also received some recent attention. Shan et al. have developed the Discriminant Latent Dirichlet Allocation (DLDA) model [38], which is a supervised version of LDA. It incorporates label information into the generative model using logistic regression. Compared to Mariana, it not only has a better accuracy, but it also provides the topics along with the classification.

Gaussian Process Topic Models (GPTMs) by Agovic and Banerjee [2] is a novel family of topic models which define a Gaussian Process Mapping from the document space into the topic space. The advantage of GPTMs is that it can incorporate semi-supervised information in terms of a Kernel over the documents. It also captures correlations among topics, which leads to a more accurate topic model compared to LDA. Experiments on ASRS dataset show better topic detection compared to LDA. The experiments also illustrate that the topic space can be manipulated by changing the Kernel over documents.

4.3. **Recovering categories.** In our first experiment, we sought to understand if the sparse learning methods could perform well in a blind test. The category data did not contain category *names*, only referring to them with letter capitals. We sought to understand what these categories were about.

Category	term 1	term 2	term 3	term 4	term 5	term 6	term 7
A	MEL	install	maintain	mechanic	defer	logbook	part
B	CATA	CATN	airspace	install	MEL	AN	
C	abort	reject	ATO	takeoff	advance	TOW	pilot
D	grass	CATJ	brake	mud	veer	damage	touchdown
E	runway	taxi	taxiway	hold	tower	CATR	ground control
F	CATH	clearance	cross	hold	feet	runway	taxiway
G	altitude	descend	feet	CATF	flightlevel	autopilot	cross
H	turn	head	course	CATF	radial	direct	airway
I	knotindicator	speed	knot	slow	airspeed	overspeed	speedlimit
J	CATO	CATD	wind	brake	encounter	touchdown	pitch
K	terrain	GPWS	GP	MD	glideslope	lowaltitude	approach
L	traffic	TACAS	RA	AN	climb	turn	separate
M	weather	turbulent	cloud	thunderstorm	ice	encounter	wind
N	airspace	TFR	area	adiz	classb	classdairspace	contact
O	CATJ	glideslope	approach	high	goaraound	fast	stabilize
P	goaround	around	execute	final	approach	tower	miss
Q	gearup	land	towerfrequency	tower	contacttower	gear	GWS
R	struck	damage	bird	wingtip	truck	vehicle	CATE
S	maintain	engine	emergency	CATA	MEL	gear	install
T	smoke	smell	odor	fire	fume	flame	evacuate
U	doctor	paramedic	nurse	ME	breath	medic	physician
V	police	passenger	behave	drink	alcohol	seat	firstclass

Table 1: LASSO images of the categories: each list of terms correspond to the most predictive list of features in the classification of one category against all the others. The meaning of abbreviations is listed in Table 2.

Abbreviation	Meaning	Abbreviation	Meaning
aborted take-off	ATO	minimumdescent	MD
aircraftnumber	AN	minimumequipmentlist	MEL
airtrafficcontrol	ATC	noticestoairspace	NTA
gearwarningsystem	GWS	resolutionadvisory	RA
groundproximity	GP	trafficalertandcollisionavoidancesystem	TACAS
groundproximitywarningsystem	GPWS	takeoffclear	TOC
groundproximitywarningsystemterrain	GPWS-T	takeoffwarning	TOW
knotsindicatedairspeed	KIAS	temporaryflightrestriction	TFR
medicalemergency	ME		

Table 2: Some abbreviations used in the ASRS data.

To this end, we have solved one LASSO problem for each category, corresponding to classifying that category against all the others. As shown in Table 1, we did recover a very accurate and differentiated image of the categories. For example, the categories M, T, U correspond to the ASRS categories *Weather/Turbulence*, *Smoke/Fire/Fumes/Odor,* and *Illness*. These categories names are part of the ASRS Events Categories as defined in `http://asrs.arc.nasa.gov/docs/dbol/ASRS_Database_Fields.pdf`. This blind test indicates that the method reveals the correct underlying categories using the words in the corpus alone.

The analysis reveals that there is a singular category, labelled B. This category makes up about 50% of the total number of reports. Its LASSO images points to two terms, which happen to be two categories, A (mechanical issues) and N (airspace issues). The other terms in the list are common to either A or N. The analysis points to the fact that category is a "catch-all" one, and that many reports in it could be re-classified as A or N.

4.4. **Sparse PCA for understanding.** A first exploratory data analysis step might be to plot the data set on a pair of axes that contain a lot of the variance, at the same time maintaining some level of interpretability to each of the four directions.

We have proceeded with this analysis on the category data set. To this end we have applied a sparse PCA algorithm (power iteration with hard thresholding) to the category data matrix M (with each column an ASRS report), and obtained Fig. 1. We have not thresholded the direction q, only the direction p, which is the vector along which we project the points, so that it has at most 10 positive and 10 negative components. The sparse PCA plot shows that the data involves four

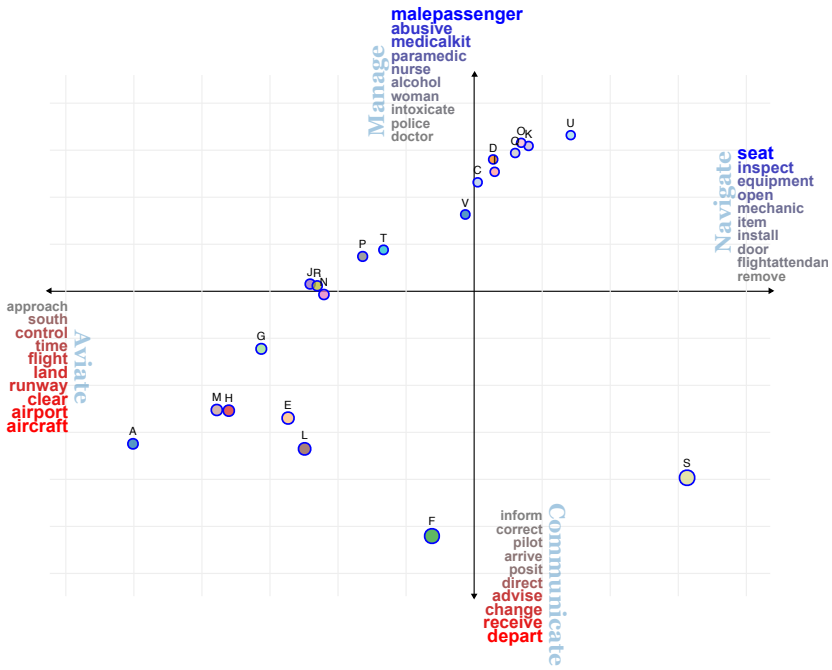

Figure 1: A sparse PCA plot of the category ASRS data. Here, each data point is a category, with size of the circles consistent with the number of reports in each category. We have focussed the axes and visually removed category B which appears to be a catch-all category. Each direction of the axes is associated with only a few terms, allowing an easy understanding of what each means. Each direction matches with one of the missions assigned to pilots in FAA documents (in light blue).

different themes, each corresponding to the positive and negative directions of the first two sparse principal components.

Without any supervision, the sparse PCA algorithm found themes that are consistent with the four missions of pilots, as is widely cited in aviation documents [22]: *Aviate*, *Navigate*, *Communicate*, and *Manage Systems*. These four actions form the basis of flight training for pilots in priority order. The first and foremost activity for a pilot is to aviate, *i.e.*, ensure that the airplane stays aloft and in control. The second priority is to ensure that the airplane is moving in the desired direction with appropriate speed, altitude, and heading. The third priority is to communicate with other members of the flight crew and air traffic control as appropriate. The final priority is to manage the systems (and humans involved) on the airplane to ensure safe flight. These high-level tasks are critical for pilots to follow because of their direct connection with overall flight safety. The algorithm discovers these four high-level tasks as the key factors in the category data set.

We validated our discovery by applying the Latent Dirichlet Allocation algorithm to the ASRS data and set the desired number of topics equal to 4. Because there is currently no method to discover the 'correct' number of topics, we use this high-level task breakdown as for an estimate of the number of topics described in the documents. While the results did not reveal the same words as sparse PCA, it revealed a similar task breakdown structure.

A a second illustration we have analyzed the runway data set. Fig 2 shows that two directions remain associated with the themes found in the category data set, namely "aviate" (negative horizontal direction) and "communicate". The airports near those directions, in the bottom left quadrant of the plot (CLE, DFW, ORD, LAX, MIA, BOS) are high-traffic ones with relatively bigger number of reports, as is indicated by the size of the circles. This is to be expected from airports where large amounts of communication is necessary (due to high traffic volume and complicated layouts).

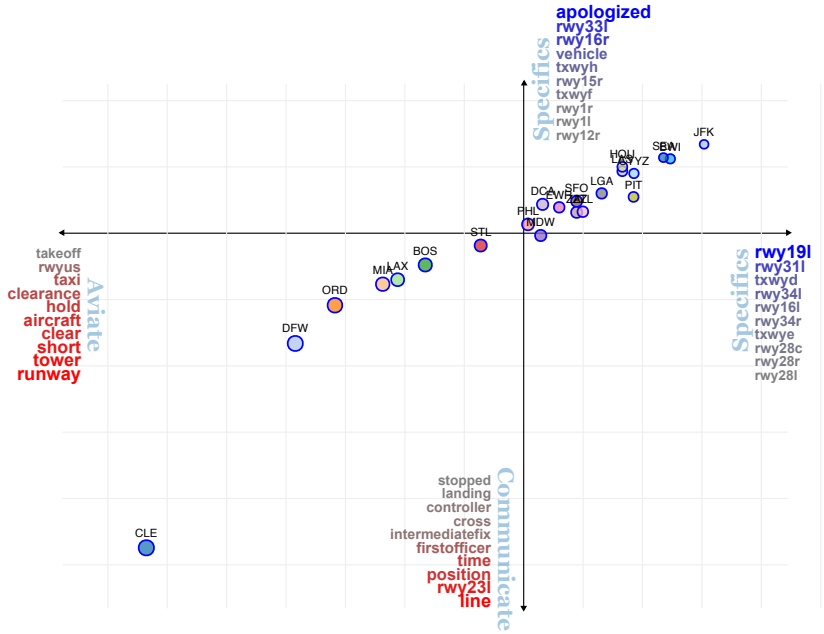

Figure 2: A sparse PCA plot of the runway ASRS data. Here, each data point is an airport, with size of the circles consistent with the number of reports for each airport.

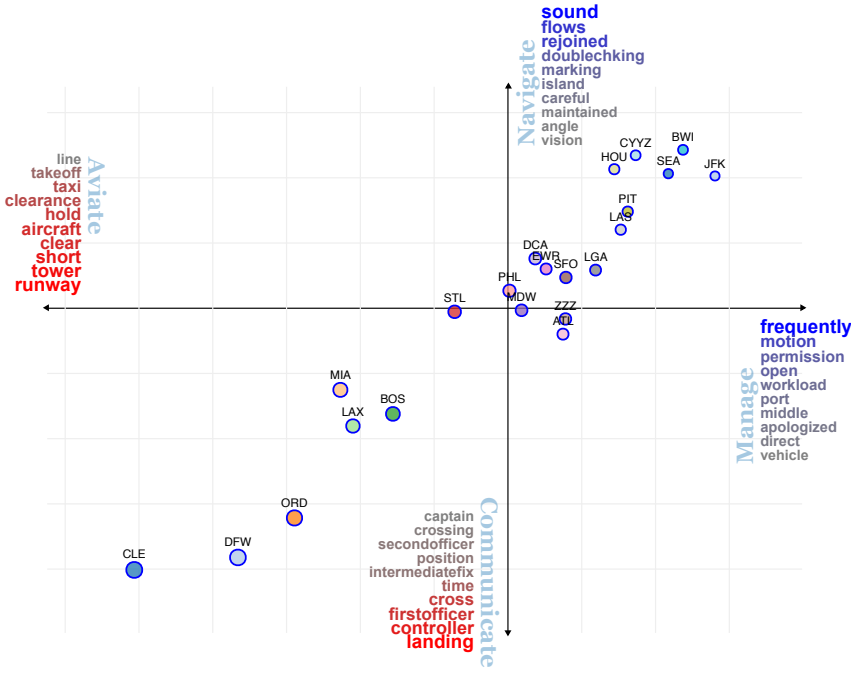

Figure 3: A sparse PCA plot of the runway ASRS data, with runway features removed.

Another cluster (on the NE quadrant) corresponds to the two remaining directions, which we labelled "specifics" as they related to specific runways and taxiways in airports. This other cluster of airports seem to be affected by issues related to specific runway configuration that are local to each airport.

In a second plot (Fig. 3) we redid the analysis after removal of all the features related to runways and taxiways, in order to discover what is "beyond" runway and taxiway issues. We recover the four themes of *Aviate*, *Navigate*, *Communicate* and *Manage*. As before, high-traffic airports remain affected mostly by aviate and communicate issues. Note that the disappearance of passenger-related issues within the *Manage* theme, which was defining the positive-vertical direction in Fig 1. This is to be expected, since the data is now restricted to runway issues: what involved passenger issues in the category data set, now becomes mainly related to the other humans in the loop, pilots ("permission"), drivers ("vehicle") and other actors, and their actions or challenges ("workload, open, apologized").

A look at the sparse PCA plots (Figs. 3 and 1) reveals a commonality: the themes of *Aviate* and *Communicate* seem to go together in the data, and are opposed to the other sub-group of *Navigate* and *Manage Systems*.

How about thresholded PCA? Fig. 4 shows the total explained variance by the two methods (sparse and thresholded PCA) as a function of the number of words allowed for the axes, for the category data set. We observe that thresholded PCA does not explain as much variance (in fact, only half as much) as sparse PCA, with the same budget of words allowed for each axis. This ranking is reversed only after 80 words are allowed in the budget. The two methods do reach the maximal variance explained by PCA as we relax our word-budget constraint. Similar observations can be made for the runway data set.

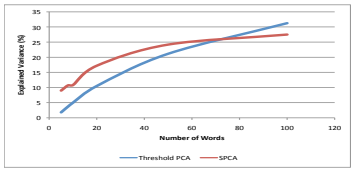

Figure 4: Explained variance.

4.5. **LASSO images of airports.** Our goal here is to use the runway data to help understand what specific runway-related issues affect each airport. To do this, we consider a specific airport and separate the runway incursion data in two sets: one set corresponds to the ASRS reports that contain the name of the airport under analysis; the other contains all the remaining ASRS documents in our corpus.

Using LASSO we can classify these two data sets, and discover the few features (terms in the dictionary) that are strong predictors of the differences. Hence we are able to single out a short list of terms that are strongly associated with the specific airport under consideration. Repeating this process for every airport provides a global, differentiated view of the runway incursion problem, as reported in the corpus analyzed. We have selected for illustration purposes the top twenty airports, as ordered by the number of reports that mention their name. The resulting short lists for a few of the airports are shown in Table 3. As expected, some airports' images point to the runways of that airport, and more importantly, to a few specific taxiways. The image of other airports, such as YYXZ (Toronto), points to other problems (lines, in the case of YYZ), and taxiways issues are less prevalent.

The LASSO analysis mostly points to specific runways for each airport. In order to go beyond this analysis, we focus on a single airport (say DFW). In the left panel of Fig 5, we propose a two-stage LASSO analysis allowing to discover a tree structure of terms. The inner circle corresponds to the LASSO image of DFW. Then, for each term in that image, we re-ran a LASSO analysis, comparing all the documents in the DFW-related corpus containing the term against all the other documents in the DFW-related corpus.

The tree analysis highlights which issues are pertaining to specific runways, and where *attention* could be focussed. In the airport diagram 6, we have highlighted some locations discussed next.

airport	term 1	term 2	term 3	term 4	term 5	term 6	term 7	term 8
CLE	Rwy23L	Rwy24L	Rwy24C	Rwy23R	Rwy5R	Line	Rwy6R	Rwy5L
DFW	Rwy35C	Rwy35L	Rwy18L	Rwy17R	Rwy18R	Rwy17C	cross	Tower
ORD	Rwy22R	Rwy27R	Rwy32R	Rwy27L	Rwy32L	Rwy22L	Rwy9L	Rwy4L
MIA	Rwy9L	TxwyQ	Rwy8R	Line	Rwy9R	PilotInCommand	TxwyM	Takeoff
BOS	Rwy4L	Rwy33L	Rwy22R	Rwy4R	Rwy22L	TxwyK	Frequency	Captain
LAX	Rwy25R	Rwy25L	Rwy24L	Rwy24R	Speed	cross	Line	Tower
STL	Rwy12L	Rwy12R	Rwy30L	Rwy30R	Line	cross	short	TxwyP
PHL	Rwy27R	Rwy9L	Rwy27L	TxwyE	amass	TxwyK	AirCarrier	TxwyY
MDW	Rwy31C	Rwy31R	Rwy22L	TxwyP	Rwy4R	midway	Rwy22R	TxwyY
DCA	TxwyJ	Airplane	turn	Captain	Line	Traffic	Landing	short
SFO	Rwy28L	Rwy28R	Rwy1L	Rwy1R	Rwy10R	Rwy10L	b747	Captain
ZZZ	hangar	radio	Rwy36R	gate	Aircraft	Line	Ground	Tower
ERW	Rwy22R	Rwy4L	Rwy22L	TxwyP	TxwyZ	Rwy4R	papa	TxwyPB
ATL	Rwy26L	Rwy26R	Rwy27R	Rwy9L	Rwy8R	atlanta	dixie	cross
LGA	TxwyB4	ILS	Line	notes	TxwyP	hold	vehicle	Taxiway
LAS	Rwy25R	Rwy7L	Rwy19L	Rwy1R	Rwy1L	Rwy25L	TxwyA7	Rwy19R
PIT	Rwy28C	Rwy10C	Rwy28L	TxwyN1	TxwyE	TxwyW	Rwy28R	TxwyV
HOU	Rwy12R	Rwy12L	citation	Takeoff	Heading	Rwy30L	Line	Tower
BWI	TxwyP	Rwy15R	Rwy33L	turn	TxwyP1	Intersection	TxwyE	Taxiway
CYYZ	TxwyQ	TxwyH	Rwy33R	Line	YYZ	Rwy24R	short	toronto
SEA	Rwy34R	Rwy16L	Rwy34L	Rwy16R	AirCarrier	FirstOfficer	TxwyJ	SMA
JFK	Rwy31L	Rwy13R	Rwy22R	Rwy13L	vehicle	Rwy4L	amass	Rwy31R

Table 3: The terms recovered with LASSO image analysis of a few airports in the "runway" ASRS data set.

Figure 5: A tree LASSO analysis of the DFW (left panel) and CYYZ (right panel) airports, showing the LASSO image (inner circle) and for each term in that image, a further image.

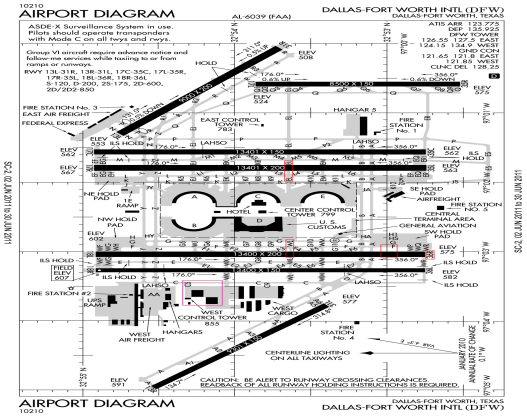

Figure 6: Diagram of DFW.

For example, as highlighted in red in the airport diagram 6, the major runway 35L crosses the taxiway EL, and the term in the tree image "simultaneously" evokes a risk of collision; similar comments can be made for the runway 36R and its siblings taxiway WL and F. At those particular intersections, the issues seem to be about obtaining "clearance" to "turn" from the tower, which might be due to the absence of line of sight from the tower (here we are guessing that the presence of the west cargo area could be a line-of-sight hindrance). The tree image is consistent with the location of DFW in the sparse PCA plot (Fig. 3), close to the themes of *Aviate* and *Communicate*.

Similar comments can be made about the tree image of the CYYZ airport, as shown in the right panel of Fig. 5. Note here that there is no mention of "ice" or other weather-related issues, which indicates that the measures taken to address them seem to work properly there.

5. Conclusions and future work

We have discussed several methods that explicitly encode sparsity in the model design. This encoding leads to a higher degree of interpretability of the model without penalizing, or even improving, the computational complexity of the algorithm. We demonstrated these techniques on real-world data from the Aviation Safety Reporting System and showed that they can reveal contributing factors to aviation safety incidents such as runway incursions. We also show that the sparse PCA and LASSO algorithms can discover the underlying task hierarchy that pilots perform.

Sparse learning problems are formulated as optimization problem with explicit encoding of sparsity requirements, either in the form of constraint or penalty. As such, the results have an explicit tradeoff between accuracy and sparsity based on the value of the sparsity-controlling parameter that is chosen. In comparison to thresholded PCA or similar methods, which provide "after-the-fact" sparsity, sparse learning methods offer a principled way to explicitly encode the tradeoff in the optimization problem. Thus, the enhanced interpretability of the results is a direct result of the optimization process.

In the safety monitoring of most critical, large-scale complex systems, from flight safety to nuclear plants, experts have relied heavily on physical sensors and indicators (temperature, pressure, etc). In the future we expect that human-generated text reporting, assisted by automated text understanding tools, will play an ever increasing role in the management of critical business, industry or government operations. Sparse modeling, by offering a great trade-off between user interpretability and computational scalability, appears to be well equipped to address some of the corresponding challenges.

6. Acknowledgments

A.N.S. thanks the NASA Aviation Safety Program, System Wide Safety and Assurance Technologies project for supporting this work, and Dr. Irving Statler, Linda Connell, and Charles Drew for their valuable insights regarding the Aviation Safety Reporting System and related discussions. L.E.G.'s work is partially supported by the National Science Foundation under Grants No. CMMI-0969923 and SES-0835531.

References

[1] M. A. U. Abedin, V. Ng, and L. Khan. Cause identification from aviation safety incident reports via weakly supervised semantic lexicon construction. *J. Artif. Int. Res.*, 38:569–631, May 2010.

[2] A. Agovic and A. Banerjee. Gaussian process topic models. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 10–19, Corvallis, Oregon, 2010.

[3] M. S. Ahmed and L. Khan. SISC: A Text Classification Approach Using Semi Supervised Subspace Clustering. In *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, ICDMW '09, pages 1–6, 2009.

[4] D. Blei and J. McAuliffe. Supervised topic models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. MIT Press, Cambridge, MA, 2008.

[5] E. Candès and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *Annals of Statistics*, 37:2145–2177, 2009.

[6] X. Dai, J. Jia, L. El Ghaoui, and B. Yu. SBA-term: Sparse bilingual association for terms. In *Fifth IEEE International Conference on Semantic Computing*, Palo Alto, CA, USA, 2011.

[7] D. Das and A. F. T. Martins. A survey on automatic text summarization, 2007.

[8] A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

[9] B. Ding, D. Lo, J. Han, and S.-C. Khoo. Efficient Mining of Closed Repetitive Gapped Subsequences from a Sequence Database. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, pages 1024–1035, 2009.

[10] B. Ding, B. Zhao, C. X. Lin, J. Han, and C. Zhai. Topcells: Keyword-based search of top-k aggregated documents in text cube. *Data Engineering, International Conference on*, pages 381–384, 2010.

[11] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196 – 212, 2004. Special Issue on Multivariate Methods in Genomic Data Analysis.

[12] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[13] J. Eisenstein, A. Ahmed, and E. P. Xing. parse additive generative models of text. In *International Conference on Machine Learning (ICML)*, 2011.

[14] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination for the LASSO. *Journal of Machine Learning Research*, 2011. Submitted, April 2011.

[15] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[16] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.

[17] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432, 2008.

[18] B. Gawalt, J. Jia, L. Miratrix, L. El Ghaoui, B. Yu, and S. Clavier. Discovering word associations in news media via feature selection and sparse classification. In *Proc. 11th ACM SIGMM International Conference on Multimedia Information Retrieval*, 2010.

[19] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 40–48, 2000.

[20] L. Hennig. Topic-based multi-document summarization with probabilistic latent semantic analysis. In *Recent Advances in Natural Language Processing (RANLP)*, 2009.

[21] T. Joachims. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer, 2002.

[22] J. E. Jonsson and W. R. Ricks. Cognitive models of pilot categorization and prioritization of flight-deck information. Technical report, 1995.

[23] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *arXiv:0811.4724*, 2008.

[24] M. Kolar, A. Parikh, and E. Xing. On Sparse Nonparametric Conditional Covariance Selection. *International Conference on Machine Learning*, 2010.

[25] C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao. Text Cube: Computing IR Measures for Multidimensional Text Database Analysis. *IEEE International Conference on Data Mining*, pages 905–910, 2008.

[26] Z. Lu, R. Monteiro, and M. Yuan. Convex optimization methods for dimension reduction and coefficient estimation in multivariate linear regression. *Mathematical Programming*, 9(1):1–32, 2010.

[27] L. Mackey. Deflation methods for sparse PCA. *Advances in Neural Information Processing Systems*, 21:1017–1024, 2009.

[28] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 7(1):214–241, 2008.

[29] N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, 37(1):246–270, 2008.

[30] L. Miratrix, J. Jia, B. Gawalt, B. Yu, and L. El Ghaoui. Summarizing large-scale, multiple-document news data: sparse methods and human validation. submitted to JASA.

[31] B. Moghaddam, Y. Weiss, and S. Avidan. Fast Pixel/Part Selection with Sparse Eigenvectors. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.

[32] B. L. Monroe, M. P. Colaresi, and K. M. Quinn. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403, 2008.

[33] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *CORE Discussion Papers*, 2010.

[34] O.Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, March 2008.

[35] N. C. Oza, J. P. Castle, and J. Stutz. Classification of Aeronautics System Health and Safety Documents. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 39(6):670–680, 2009.

[36] I. Persing and V. Ng. Semi-supervised cause identification from aviation safety reports. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, pages 843–851, 2009.

[37] F. Schilder and R. Kondadadi. Fastsum: fast and accurate query-based multi-document summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 205–208, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[38] H. Shan, A. Banerjee, and N. C. Oza. Discriminative Mixed-Membership Models. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 466–475, Washington, DC, USA, 2009.

[39] H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *J. Multivar. Anal.*, 99:1015–1034, July 2008.

[40] J. Songsiri and L. Vandenberghe. Topology selection in graphical models of autoregressive processes. *Journal of Machine Learning Research*, 2010.

[41] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal statistical society, series B*, 58(1):267–288, 1996.

[42] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Trans. Inform. Theory*, 51(3):1030–1051, Mar. 2006.

[43] C. Woolam and L. Khan. Multi-concept Document Classification Using a Perceptron-Like Algorithm. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 570–574, 2008.

[44] C. Woolam and L. Khan. Multi-label large margin hierarchical perceptron. *IJDMMM*, 1(1):5–22, 2008.

[45] M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19, 2007.

[46] D. Zhang, C. Zhai, J. Han, A. Srivastava, and N. Oza. Topic modeling for OLAP on multidimensional text databases: topic cube and its applications. *Stat. Anal. Data Min.*, 2:378–395, December 2009.

[47] Y. Zhang, A. d'Aspremont, and L. El Ghaoui. Sparse PCA: Convex relaxations, algorithms and applications. In M. Anjos and J. Lasserre, editors, *Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*. Springer, 2011. To appear.

[48] H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational & Graphical Statistics*, 15(2):265–286, 2006.