Anomaly Detection for Astronomical Data

Liang Xiong LXIONG@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University

Barnabas Poczos BAPOCZOS@CS.CMU.EDU
Robotics Institute
Carnegie Mellon University

Andrew Connolly AJC@ASTRO.WASHINGTON.EDU
Department of Astronomy
University of Washington

Jeff Schneider SCHNEIDE@CS.CMU.EDU
Robotics Institute
Carnegie Mellon University

Abstract

Modern astronomical observatories can produce massive amount of data that are beyond the capability of the researchers to even take a glance. These scientific observations present both great opportunities and challenges for astronomers and machine learning researchers. In this project we address the problem of detecting anomalies/novelties in these large-scale astronomical data sets.

Two types of anomalies, the *point anomalies* and the *group anomalies*, are considered. The point anomalies include individual anomalous objects, such as single stars or galaxies that present unique characteristics. The group anomalies include anomalous groups of objects, such as unusual clusters of the galaxies that are close together. They both have great values for astronomical studies, and our goal is to detect them automatically in un-supervised ways.

For point anomalies, we adopt the subspace-based detection strategy and proposed a robust low-rank matrix decomposition algorithm for more reliable results. For group anomalies, we use hierarchical probabilistic models to capture the generative mechanism of the data, and then score the data groups using various probability measures.

Experimental evaluation on both synthetic and real world data sets shows the effectiveness of the proposed methods. On a real astronomical data sets, we obtained several interesting anecdotal results. Initial inspections by the astronomers confirm the usefulness of these machine learning methods in astronomical research.

Keywords: Astronomical data; Anomaly detection; Low-Rank decomposition; Robust methods; Group/Collective Anomaly; Hierarchal Probabilistic Models.

1. Introduction

Contemporary astronomical observation systems produce massive amount of data out of their automatic pipelines. Take the *Sloan Digital Sky Survey*[1] (SDSS) for example, over eight years of effort produced images and spectroscopic measurements covering more than

1. `http://www.sdss.org`

a quarter of the sky (over 1.1×10^5 square degrees). The resulting data set contains about 230 millions of celestial objects, for 1.6×10^6 of which we have the spectra. Nowadays, researchers are planning to build the *Large Synoptic Survey Telescope*[2] (LSST), which will be able to scan half of the sky twice in one week to provide update-to-date and detailed information about what is happening in the universe. These survey data can open a new window to the observable universe for us.

However, these rich sets of data also presents challenges besides opportunities. Examining them by human experts is conventional but clearly not feasible given the large volume of the data. The solution is to use computational methods to automate some processes so as to assist subsequent studies. Since the knowledge about the whole data set is generally missing, our first step is to do unsupervised anomaly/novelty detection on these data. The hope is to let machine learning algorithms run through the data and pick out the most "interesting" things on which the experts can do further detailed examinations.

Anomaly/novelty detection is about finding unusual things that do not conform to our established knowledge about the data. Two goals were implied by the two different names of this technique: 1) to eliminate the influence of outliers and find a reliable model for the data; 2) to find outliers themselves that could bring us new insights. In the context of astronomical data, we are trying to pick out from a vast pool the unusual objects that may bare interesting scientific values. These objects can then be presented to human expert for further study. This step is very important because based on it we start to build our knowledge base for the data.

Two types of anomalies detection problems are addressed in this project. The first type is the *point anomaly*. In this project, anomalies of this kind are individual celestial objects that present unusual characteristics. For example, supernovas, planetary nebulae, and black holes (although they are not observed in SDSS) themselves are very unique and possess great scientific values. In the machine learning terms, point anomalies are individual samples that do not conform to the majority's behavior in the whole data set. This type problems have been extensively studied in the anomaly detection literatures, and the main idea is usually to find points in the low-density region of the data distribution. For a comprehensive overview we refer readers to the survey by Chandola et al. (2009).

The second problem is the detection of *group anomalies*. A group anomaly is an unusual collection of points. This type of anomaly occurs naturally in practical problem such as time series and spatial data analysis, in which points are grouped according to their temporal or spatial affinity. A group of points can be considered abnormal either because it is a collection of anomalous points, or because that the way its member points aggregate is unusual, even if the points themselves are perfectly normal. While the former case is primary addressed by scan methods such as Neill and Cooper (2010), here we focus on the latter case. To see an example, imagine that on a grassland where you see many sheep and wolves. Clearly there will be no surprise when you see a group of sheep or a group of wolves. But if sheep and wolves are mixed in a group, something unusual is going on (*e.g.* a hunt is taking place) even if sheep and wolves on their own are very common on the grassland. In astronomy, our target of detection is the spatial clusters of stars and galaxies. Finding special instances of these groups can help us understand issues such as galaxy evolution and dark matter.

2. `http://www.lsst.org/lsst`

For the point anomaly detection problem, since the data set is high-dimensional and has a large volume, we adopt the subspace-based anomaly detection method. The basic assumption is that the variability of normal data is limited *i.e.* the feature of a normal sample point can be reconstructed by the linear combination of a few basis features. Then, samples that cannot be well-reconstructed by these bases are considered anomalies. We solve this problem using low-rank decomposition methods like *principal component analysis* (PCA) or *singular value decomposition* (SVD). To improve the reliability of the bases and hence the detection results, we propose a simple but effective robust low-rank matrix factorization algorithm called *Mixed-Error Matrix Factorization* (MEMF). MEMF can effectively alleviate the influence of outliers while maintaining the efficiency of matrix factorization methods, and its flexibility makes it a suitable framework to support various robust low-rank analysis of matrices and subspace learning.

For the group anomaly detection problem, we develop a hierarchical probabilistic model to capture the generative process of the data. By treating each group as a bag of exchangeable points, we can use a hierarchical mixture model to characterize the data. For anomaly detection, different probabilities are used as our criterion for scoring the groups according to their anomalousness. The effectiveness of this model is verified by empirical results. Another advantage of this generative method is that in addition to find anomalous data, we get a compact summary of the data set that can help us interpret the results.

This report consists of two relatively separate parts. Section 3 addresses the point anomaly detection problem and describe the MEMF algorithm. Section 4 addresses the group anomaly detection problem. Besides these two parts, we describe the data set in section 2 and present the experimental results in section 5. Discussion and Conclusions are made in section 6.

2. Data Description

In this project, we deal with the data from the *Sloan Digital Sky Survey*[3] (SDSS) data release 7, which can be downloaded from site `http://cas.sdss.org/dr7/en/tools/search/sql.asp`. This data set contains 230 millions of celestial objects through eight years of effort. Specifically, SDSS provides us with the spectra of 3.8×10^5 stars, 9.3×10^5 galaxies, and 1.1×10^5 quasars. The subset of data we are interested in now includes 85564 stars and 807118 galaxies, which are basically the portion with adequate quality.

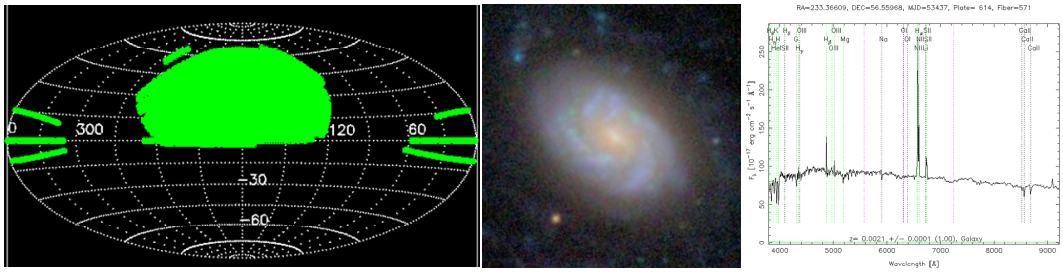
For each of the object, SDSS provides a rich set of features, as described below. Some sample data from SDSS are shown in Figure 1.

- **Astrometric** feature: The location of the object in the sky. Note that the objects' depths/distances are hard to measure. For stars, we can use the *parallax* method to accurately measure their distances. But this process is expensive and only feasible for close-by stars. For galaxies, the distances are estimated from the their *redshifts*. While this is simple, the precision is low.

- **Photometric** feature: This is the actual photo taken by the telescope. Currently it is not used in the project.

3. `http://www.sdss.org`

- **Spectroscopic** feature: The spectra of the observed objects. This description gives us a object's flux in each frequency bin/pixel, as well as the measured emission lines and the redshift. This feature is our primary source of information about he physical characteristics of the objects.



(a) (b) (c)

Figure 1: Summary of the SDSS data set. (a) The coverage map of SDSS. (b) A sample imaging data. (c) A sample spectroscopic data *i.e.* the spectrum.

We pre-process the data before apply learning algorithms on them. Basically we throw out bad observations that are marked bad, too noisy, too bright, or contains too many bad pixels. Then the bad pixels are repaired by smoothing based on PCA. After the pre-processing, we have two data sets. The first one contains 49529 stars. And the second one contains 651473 galaxies. Each object is described by the following two features:

- **Spectrum**: This is the raw spectroscopic observation from the telescope. We re-bin the spectrum to form a 500-dimensional vectors. We call the spectrum the $\mathcal{S}$ feature.
- **Continuum**: This is the smoothed spectrum with emission lines removed. We call the continuum the $\mathcal{C}$ feature.

Having these two feature, we can normalize them so that the total flux of each object is 1. This normalization takes out the effect of the brightness/distance of the an object, and we call the resulting features $\mathcal{S}_{s1}$ and $\mathcal{C}_{s1}$ respectively.

3. Point Anomaly Detection

In this section we address the problem of detecting point anomalies.

Our astronomical data set is characterized by large sample size and high dimensionality, which impose specific constraints on the anomaly detection algorithms we use. Because of the high dimensionality, most of the density based methods (*e.g. Local Outlier Factor* by Breunig et al. (2000)) might not work properly. Also, the scale of the data set urges us to develop highly efficient algorithm. Therefore, we resort to *subspace-based* methods for anomaly detection. The basic idea is that, we assume the normal data lie in a low-dimensional subspace, and the anomalies lie outside of that subspace. The implication of this assumption is that a normal star's feature (*e.g.* the spectrum) can be represented as a

linear combination of a few bases. This idea is very similar to the case where we approximate signals using limited-bandwidth *Fourier transform* (FT). Having the bases, we can identify anomalies if they cannot be well reconstructed *i.e.* is far away from the normal subspace.

To do this, we can use various subspace modeling methods such as *principal component analysis* (PCA) and *non-negative matrix factorization* (NMF) by Lee and Seung (1999); Ding et al. (2010). Most of these methods are essentially finding *low-rank decomposition* of the data matrix, which is formed by stacking the feature vectors together. These algorithms are usually quite efficient since they are reducing the dimensionality of the data. Besides the anomaly scores, their outputs are also very useful for further analysis such as learning and visualization.

However, before we can confidently use a model, which in our case consists of the few linear bases, to define what data are anomalous, we need to make sure that our model is itself reliable. This could be a problem when we are doing un-supervised detection without knowing which points are anomalous, and the anomalies are used to train model together with the normal data. To alleviate the influence of these outliers, we need robust model estimation techniques.

We usually estimate models by minimizing the errors between the model and the data. For many tasks, the The L_2 or the *sum of squared errors* measurement is used for errors, however it is also unfortunately well-known for its sensitivity to outliers. A common way to deal with outliers is to use "robust" measurements of errors. In machine learning and statistics, the L_1 of the *sum of absolute errors* is used in robust algorithms such as *least absolute deviations* regression (Bloomfield and Steiger (1983)) and *robust principal component analysis* (Wright et al. (2009)). *Hinge loss* has been proved very effective in promoting the performance of classifiers such as the *support vector machine* (SVM). Other measures like the *Huber loss* (Huber and J. (1964)) and the *Geman-McClure* function have also be employed in robust procedures in la Torre and Black (2003); Nguyen and la Torre (2009).

Following this direction, here we propose a method that does robust modeling and anomaly detection in the context of low-rank matrix factorization. Here the notion of normality, as mentioned before, is that the rows and columns of the target matrix $\mathbf{X}$ can be well approximated by the linear combinations of a few bases, which in matrix terms means that $\mathbf{X}$ is of low-rank. The goal of the proposed model, named *mixed-error matrix factorization* (MEMF), is to identify outlier entries in $\mathbf{X}$ and fit a robust low-rank model simultaneously.

The intuition behind the MEMF model is as follows. We assume that the approximation error of the low-rank factorization is an additive *mixture* of both the regular Gaussian noise and the outliers. Then we design two separate parts in the MEMF model that account for these two types of errors respectively. In the estimation process, MEMF tries to fit the whole matrix using the low-rank model under the Gaussian noise assumption. But it is also allowed to throw out entries that is unacceptable under the Gaussian model and put them into the outlier part, so that they will not interfere with the low-rank structure. Moreover, due the properties of the outlier measures, sparsity is often induced in the model so we can easily identify the outliers.

We developed an efficient algorithm to estimate the MEMF model based on block coordinate descent. This algorithm is very flexible and allows for the user to plug-in their own favorite factorization modules and outlier detection components. For factorization, various

off-the-shelf methods such as SVD and NMF can be used in MEMF without adaptation. MEMF is able to fully enjoy the advantages such as efficiency and interpretability of the factorization components.

For the outlier part, we show that in addition to finding outlier entries in a matrix, we can also design structured outlier measures similar to *group lasso* by Yuan and Lin (2006); Wang and Leng (2008) to detect anomalous patterns such as rows, columns, or any groups of entries in the matrix. These structured measurements are able to aggregates partial evidences into a whole to get a better indication of anomalies. Empirically we show that if a proper structure was designed for the outliers not only can they be better detected but also the fitting of the normal data can be more accurate.

Having these flexibilities, we consider the MEMF model as a general framework to convert (constrained) L_2 error based matrix factorization methods into their robust versions. Several concrete realizations of MEMF using different components are demonstrated in various applications.

We test the performance of MEMF on both synthetic and real-world data sets. In simulated experiments, we illustrate the key differences and advantages of MEMF over its state-of-the-art peers in both efficiency and accuracy. We then test the performance of MEMF on real-world problems including video modeling, text clustering. Results show that MEMF is a simple, versatile, and powerful tool in handling these tasks. On our astronomical data set, MEMF is able to produce very interesting and promising results.

The rest of this section is structured as follows. We give some background and notation in section 3.1. In section 3.2 we formally describe the proposed mixed-error matrix factorization algorithm. Related work are discussed in section 3.3. And we summarize this the MEMF method in 3.4. Empirical results are presented in section 5.1.

3.1 Background and Notation

Matrices are extremely useful in representing data in various problems. For example, in regression and classification analysis, the samples are often organized into a *design matrix* in which each row corresponds to a sample and each column corresponds to a feature/attribute. A similar representation called document-term matrix is used for text data. *Connectivity matrices* are widely used to express network and graph data. Here we denote the data matrix of size $m \times n$ as $\mathbf{X} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$. Having $\mathbf{X}$, one of the first analysis we could do is matrix factorization. For design matrices, PCA/SVD can be applied to $\mathbf{X}$ so that we see the linear structure and intrinsic dimensionality of the data. Given text data, PLSI can be applied. For network data, the notion of low-rank matrices is extremely useful in *matrix completion* (Candès and Tao (2009); Mazumder et al. (2009)) and *collaborative filtering* (Rennie and Srebro (2005); Salakhutdinov and Minh (2007)).

3.1.1 Matrix Factorization

In this paper, we assume that the data matrix has a low rank and can be factorized as

$$\mathbf{X} \approx \mathbf{U}^T\mathbf{V}, \mathbf{U} \in \mathbb{R}^{k\times m}, \mathbf{V} \in \mathbb{R}^{k\times n}, \tag{1}$$

where k is the rank of the factorization and usually $k \ll \min(m, n)$. The intuition behind this factorization models is that the rows/columns of $\mathbf{X}$ can be approximated by the

Norm	Value	Comment
$\|\mathbf{E}\|_F$	$\sqrt{\sum_{i,j} E_{ij}^2} = \sqrt{trace\left(\mathbf{E}^T\mathbf{E}\right)}$	The L_2-norm or the Frobenius-norm. Sum of squares.
$\|\mathbf{E}\|_0$	$\sum_{i,j} I\left(E_{ij} \neq 0\right)$	The L_0-norm (not strictly a "norm"). Number of non-zero entries.
$\|\mathbf{E}\|_1$	$\sum_{i,j} \|E_{ij}\|$	The L_1-norm. Sum of absolute values.
$\|\mathbf{E}\|_{0\text{-}1}$	$\sum_{i=1}^{m} I\left(\mathbf{E_{i:}} \neq \mathbf{0}\right)$	$L_{0\text{-}1}$-norm. Number of non-zero rows.
$\|\mathbf{E}\|_{2\text{-}1}$	$\sum_{i=1}^{m} \|\mathbf{E_{i:}}\|_2$	$L_{2\text{-}1}$-norm. Sum of the "length" of rows.

Table 1: Norms for error measurement.

combination of a few bases (rows of $\mathbf{V}/\mathbf{U}$). Note that sometimes the tri-factorization form

$$\mathbf{X} \approx \mathbf{USV}^T, \mathbf{U} \in \mathbb{R}^{m\times k}, \mathbf{S} \in \mathbb{R}^{k\times k}, \mathbf{V} \in \mathbb{R}^{n\times k} \tag{2}$$

is used, as in SVD. Our MEMF model admits both forms, but for simplicity we will only use (1) in this paper.

In many cases constraints are imposed on the factor matrices $\mathbf{U}, \mathbf{V}$ for purposes like interpretability. For instance, in SVD we require the columns to be *orthonormal*. In the NMF family (Ding et al. (2010)) various settings are applied to accommodate specific data and applications. In PLSI the factor matrices have probabilistic interpretations. In the MEMF model, we generalize and denote them in the form

$$\mathbf{X} \approx \mathbf{U}^T\mathbf{V}, \mathbf{U} \in \mathcal{D}_{\mathbf{U}}, \mathbf{V} \in \mathcal{D}_{\mathbf{V}}, \tag{3}$$

where $\mathcal{D}_{\mathbf{U}}/\mathcal{D}_{\mathbf{V}}$ are the feasible domains of $\mathbf{U}/\mathbf{V}$. This extension is very important as it supports better interpretability and domain-specific applications.

3.1.2 Robust Error Measurement

Another important part of MEMF is robust error measurement. As mentioned in the introduction, there exist many choices for the purpose of robust modeling. Here we mainly consider measures that can be formalized as the norm of the error matrix. Suppose we have an error matrix $\mathbf{E}$, the norms we used to measure $\mathbf{E}$ are listed in Table 1. Note that many of these norms are generalized vector norms and not strictly the matrix norms that we usually see. We use the *Matlab* notation to denote sub-matrices. For example $\mathbf{X_{i:}}$ means the i-th row of $\mathbf{X}$ and $\mathbf{X_{:j}}$ is the j-th column.

A very attractive property of these robust measures is that they often induce *sparsity*. That is, when we minimize a error measurement defined by these norms, many components of the error matrix will be exactly zero and let the outliers stand out. Recently in machine learning, statistics, and signal processing, these norms have been used as regularizations to get compact model representations. Particularly, measurements proposed for *structured*

sparsity (Jenatton et al. (2009)) can enable us to incorporate prior knowledge on the structures of outliers. Using these structures, we can collect evidences of anomalousness from each entry and aggregate them to get better performance.

3.2 Mixed-Error Matrix Factorization

In this section we describe the *mixed-error matrix factorization* (MEMF) algorithm. As in other matrix factorization models, we are seeking a decomposition $\mathbf{X} \approx \mathbf{U}^T\mathbf{V}$. However, the factorization errors are treated differently. We consider the additive error decomposition $\mathbf{X} = \mathbf{U^T V} + \mathbf{E} + \mathbf{O}$, where $\mathbf{E}$ is the "small" Gaussian noise and $\mathbf{O}$ is the outlier matrix whose distribution is unknown and magnitude can be very large.

The factorization model is fitted by minimizing the errors. If the errors $\mathbf{E}$ and $\mathbf{O}$ are together measured using the *Frobenius* norm, then the factorization is a least squares problem and solved by SVD. However it has been shown that this method is not robust according to Hampel et al. (1986). So instead, we choose to measure the errors and outliers differently. Specifically, the Gaussian noise $\mathbf{E}$ is measured by the Frobenius norm and the outlier $\mathbf{O}$ is measured by a robust L_r-norm, so that we can accommodate outliers in the model without much impact on the true low-rank model.

We adopt the common assumption that the amount of outliers in the whole data matrix is very small. Therefore to optimize the model parameters, we try to minimize the Gaussian error while allowing the model to exclude a small part of the matrix as outliers. Meanwhile, the constraints on factor matrices $\mathbf{U}, \mathbf{V}$ are still retained. The above motivations can be summarized by the following optimization problem called *mixed-error matrix factorization* (MEMF):

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{O}} \quad & \|\mathbf{E}\|_F^2 \\ & \|\mathbf{O}\|_r \leq \eta \\ s.t. \quad & \mathbf{X} = \mathbf{U}^T\mathbf{V} + \mathbf{E} + \mathbf{O} \\ & \mathbf{U} \in \mathcal{D}_{\mathbf{U}}, \mathbf{V} \in \mathcal{D}_{\mathbf{V}} \end{aligned} \tag{4}$$

where $\|\cdot\|_r$ is the robust L_r-norm of the user's choice, and η is the maximal amount of outliers that can be excluded. The intuition of the above problem is obvious: the model tries to fit the best factorization model given that it is allowed to throw out some outliers. For example, if we choose the $r = 0$ and use the L_0-norm, then we are allowing the model to pick out η outliers, and fit the best factorization model for the rest of data in the matrix. In the end small errors will be in $\mathbf{E}$, and the large sparse outliers will be in $\mathbf{O}$ so that they are separated from the low-rank model and easy to identify.

Note that this way of mixing the errors are quite different from what is commonly done in probabilistic methods. While we are assume *additive mixture* of noise, in probabilistic modeling it is often assumed that the errors are from a *probabilistic mixture* of different noise distributions (*e.g.* Kuss et al. (2005)). In this case we need latent variables in the model which are later integrated out. The resulting algorithm is usually an *Expectation-Maximization* procedure. Unfortunately, the M-step involves optimizing a objective containing weighted sum of $\|\cdot\|_r$, which is often hard and slow. On the other hand, as we will show later, our additive mixture leads to much simpler and more efficient algorithms.

In this report we deal with the *Lagrangian* version of problem (4) below

$$\begin{array}{cc} \min_{\mathbf{U},\mathbf{V},\mathbf{O}} & \frac{1}{2}\left\|(\mathbf{X}-\mathbf{O})-\mathbf{U}^T\mathbf{V}\right\|_F^2+\lambda\|\mathbf{O}\|_r \\ s.t. & \mathbf{U}\in\mathcal{D}_{\mathbf{U}},\mathbf{V}\in\mathcal{D}_{\mathbf{V}}, \end{array} \tag{5}$$

which is easier to handle. To optimize this problem, we adopt the *block coordinate descent* scheme. We first fix $\mathbf{O}$ to its current value and solve for $\mathbf{U},\mathbf{V}$, and then we fix $\mathbf{U},\mathbf{V}$ and solve for $\mathbf{O}$. This procedure is described in algorithm 1.

Algorithm 1 Mixed-Error Matrix Factorization (MEMF)

1. Specify the robust norm $\|\cdot\|_r$ for outliers, and the parameter λ.
2. While not converged:
   (a) Solve the decomposition problem

$$\begin{aligned} &\mathbf{U},\mathbf{V}=\arg\min_{\mathbf{U},\mathbf{V}}\frac{\mathbf{1}}{\mathbf{2}}\left\|\mathbf{A}-\mathbf{U^T V}\right\|_{\mathbf{F}}^{\mathbf{2}},\mathbf{A}=\mathbf{X}-\mathbf{O}, \\ &s.t.\ \mathbf{U}\in\mathcal{D}_{\mathbf{U}},\mathbf{V}\in\mathcal{D}_{\mathbf{V}} \end{aligned} \tag{6}$$

   (b) Solve the outlier problem

$$\mathbf{O}=\arg\min_{\mathbf{O}}\frac{\mathbf{1}}{\mathbf{2}}\|\mathbf{B}-\mathbf{O}\|_{\mathbf{F}}^{\mathbf{2}}+\lambda\|\mathbf{O}\|_{\mathbf{r}},\mathbf{B}=\mathbf{X}-\mathbf{U^T V}, \tag{7}$$

The advantage of the MEMF algorithm is that the decomposition problem and outlier problem are now optimized separately. This allows us to plug-in existing factorization procedures based on the L_2-norm to solve (6), and then efficiently solve (7) for $\mathbf{O}$ with $\|\cdot\|_r$. Although theoretically more sound and efficient algorithms such as *proximal gradient* (Nesterov (2007)) can easily be developed for the MEMF problem (5), we observe that algorithm 1 works very efficiently, and we prefer its simplicity and flexibility.

In the rest of the section, we describe several realizations of the MEMF model with different choices of the outlier measure and the decomposition model. The convergence property and scalability will also be discussed.

3.2.1 The Outlier Problem

In this section we show how to optimize (7) w.r.t. robust measures in table 1 and discuss their properties. Most of the usual robust measurements are included here, showing the flexibility of MEMF.

L_0-Norm and L_1-Norm The L_0-norm counts the number of outliers in the data set. In some sense it is the ideal measurement we should use for detecting outliers and learning sparse models since it does not assume anything about the distribution or characteristics of the outliers. However, the L_0-norm is usually not used in statistics and machine learning because it cannot be used as a pure measurement on its own. For a noisy data set where the model is unable to match the data perfectly, counting the number of errors is senseless:

you will almost always get the number of data points. However, but decomposing the errors and allowing for ubiquitous errors like the Gaussian noise, we are able to use the L_0-norm. In experiments, we observe that the L_0-norm indeed can be advantageous over others in some problems.

Although L_0-norm is in general difficult to optimize due to its non-convexity, here we are able get the exact solution thanks to its separability. It is easy to see that the solution to (7) with the L_0-norm is

$$\begin{aligned} o_{ij} &= \arg\min_{o_{ij}} \frac{1}{2}(b_{ij} - o_{ij})^2 + \lambda I\left(o_{ij} \neq 0\right) \\ &= \begin{cases} b_{ij} & \text{if } b_{ij}^2 > 2\lambda \\ 0 & \text{otherwise} \end{cases}, \forall i, j \end{aligned} \tag{8}$$

where o_{ij} is the (i,j)-th entry of $\mathbf{O}$ and b_{ij} is the (i,j)-th entry of $\mathbf{B}$.

The L_1-norm is a classic choice for robust measurement and regularization for sparsity. It is the tightest convex relaxation of the L_0-norm. The optimization of the L_1-norm has been extensively studied in the *lasso* (Tibshirani (1996)) family algorithms. Here the problem (7) with L_1-norm can also be decomposed into the sub-problems

$$\begin{aligned} o_{ij} &= \arg\min_{o_{ij}} \frac{1}{2}(b_{ij} - o_{ij})^2 + \lambda \left|o_{ij}\right| \\ &= \begin{cases} b_{ij} - \lambda & \text{if } b_{ij} > \lambda \\ b_{ij} + \lambda & \text{if } b_{ij} < -\lambda \\ 0 & \text{otherwise} \end{cases} \forall i, j \end{aligned} \tag{9}$$

in which the solution is obtained by setting the *sub-differential* of the objective to be zero. The operator leading to this solution is often called *soft-thresholding* in literature.

We can see that with both the L_0 and L_1 norms, the solution to $\mathbf{O}$ is obtained by *thresholding* the residual of the factorization. This operation can be done very efficiently. It is also very intuitive: if a residual is too large, we put it in the outlier part. In the L_0-norm case, we just put large residuals into $\mathbf{O}$, while in the L_1-norm case we are "conservative" and only put the "shrinked" residuals into $\mathbf{O}$. These solutions justify the common practice of truncating large residuals.

By adjusting the value of λ, we can control the threshold for the residuals and how many outliers to put into $\mathbf{O}$. Furthermore, since the thresholding procedure will leave us a sparse $\mathbf{O}$, it is very easy to identify the outliers.

$L_{0\text{-}1}$-Norm and $L_{2\text{-}1}$-Norm The L_0 and L_1 norms are additive combinations of measures on individual entries. Now we consider norms that are only separable w.r.t. groups of entries. We call them the *structured* or *composite* norms. Here, both the $L_{0\text{-}1}$-norm and the $L_{2\text{-}1}$-norm group elements by rows. The $L_{0\text{-}1}$-norm measures the number of nonzero rows in a matrix and $L_{2\text{-}1}$ measures the total "length" of the row vectors. They can be considered the generalizations of the L_0 and L_1 norms, since they are equivalent on matrices of size $m \times 1$. Like the L_0 and L_1 norms, these *composite* norms are also robust and can induce sparsity. The latter has been extensively used in tasks that desires *group sparsity* when handling categorical variables (Yuan and Lin (2006)) and multi-task learning problems (Liu et al. (2009)).

Similar to the case of L_0-norm, we can get the solution to the $L_{0\text{-}1}$-norm as

$$\begin{aligned}\mathbf{O}_{i:} &= \arg\min_{\mathbf{O}_{i:}} \frac{1}{2}\|\mathbf{B}_{i:} - \mathbf{O}_{i:}\|_2^2 + \lambda I\left(\mathbf{O}_{i:} \neq 0\right) \\ &= \begin{cases} \mathbf{B}_{i:} & \text{if } \|\mathbf{B}_{i:}\|_2^2 > 2\lambda \\ 0 & \text{otherwise} \end{cases}, \forall i.\end{aligned} \tag{10}$$

And the solution to the $L_{2\text{-}1}$-norm problem is

$$\begin{aligned}\mathbf{O}_{i:} &= \arg\min_{\mathbf{O}_{i:}} \frac{1}{2}\|\mathbf{B}_{i:} - \mathbf{O}_{i:}\|_2^2 + \lambda\|\mathbf{O}_{i:}\|_2 \\ &= \begin{cases} \left(1 - \lambda/\|\mathbf{B}_{i:}\|_2\right)\mathbf{B}_{i:} & \text{if } \|\mathbf{B}_{i:}\|_2 > \lambda \\ 0 & \text{otherwise} \end{cases}, \forall i,\end{aligned} \tag{11}$$

which can be obtained by analyzing the KKT conditions. Again these solutions can still obtained via efficiently thresholding the rows of the residual matrix $\mathbf{B}$.

The composite norms allow us to specify the structures of outliers so that we can aggregate the residuals by groups to better indicate anomalies. These structures appear a lot in practical problems. For example, in design matrices each row corresponds to a sample point. While MEMF with the L_0-norm is able to detect entries *i.e.* one feature value for one sample point, MEMF with the $L_{0\text{-}1}$-norm can detect the entire sample point as whole. For graph data, picking out an entry means detecting an anomalous link, while picking out a row means detecting an anomalous node/entity. These structured detection results are usually much more intuitive to perceive and interpret. In the experiments we also show that they can generate better robust models compared to un-structured models when the underlying problem admits.

It should be noted that although we choose the $L_{0\text{-}1}$ and $L_{2\text{-}1}$ norms to demonstrate structured outlier detection for their clarity, the structure of outliers that can be specified for MEMF is not restricted to rows or columns. In fact, it is very easy to design arbitrarily shaped groups in the outlier measure $\|\cdot\|_r$ to accommodate specific applications, and then optimize them using procedures similar to (10)(11) as long as the groups do not overlap.

3.2.2 The Factorization Problem

In this section we describe some of the factorization methods that can solve problem (6) in the MEMF algorithm.

A common choice of constraint on the factor matrices is orthonormality, as in SVD. In MEMF, if orthonormality is required, then $\mathbf{U}, \mathbf{V}$ are the left and right singular vectors of the matrix, and we can use SVD to solve (6) directly, and the solution will be globally optimal. In practice, since only the leading singular vectors are need, we can use fast partial SVD software such as PROPACK by Larsen to accelerate the computation. If we relax and impose no constraints on $\mathbf{U}, \mathbf{V}$, then the problem can be solved by *alternating least squares*, which optimize $\mathbf{U}$ and $\mathbf{V}$ in turn. Each sub-problem in this case is convex and done by solving linear systems $\left(\mathbf{V}\mathbf{V}^T\right)\mathbf{U} = \mathbf{V}\mathbf{A}^T$ and $\left(\mathbf{U}\mathbf{U}^T\right)\mathbf{V} = \mathbf{U}\mathbf{A}^T$.

We can introduce domain knowledge into the MEMF factorization component. For example, the non-negativity constraints in NMF can be directly applied in MEMF. NMF

results usually have strong interpretability and connection to other methods such as clustering. Recently, NMF gained a lot of interest in the data mining community and many algorithms have been devised (Ding et al. (2010)). It is easy to incorporate various non-negativity constraints into MEMF and derive the corresponding robust NMF algorithms.

If the speed of the algorithm is crucial, or we want some pass-efficient algorithms for disk-based data, then accelerated versions of approximate SVD can be used in MEMF. For example, we can integrate the algorithm by Nguyen et al. (2009) into MEMF. The result would be a large-scale robust matrix factorization algorithm, which could be very useful in large-scale data analysis.

3.2.3 Convergence and Scalability

It is easy to see that each step of Algorithm 1 is guaranteed to improve the objective within the feasible region, so the algorithm is going to converge. However, it is important to note that the matrix factorization problem is not convex. Although sometimes the special structure of the problem allows algorithms like SVD to achieve a global minimum (Srebro and Jaakkola (2003)), we usually only get local optimums. Recently, the convex relaxations of the low-rank factorization problems such as in Candès and Tao (2009); Wright et al. (2009); Mazumder et al. (2009) became popular and can also fit into the MEMF framework if we convert the explicit factorization form into a low-rank constraint. We might be concerned if the algorithm is converging to a stationary point of the objective function or will it get stuck somewhere else. Empirically we observe that it is unlikely that the MEMF algorithm gets stuck at a non-stationary point, possibly due to the very large block size.

We have shown that the solution to (7) is exact and efficient. Then the complexity of MEMF is determined by the specific sub-routine use to solve (6). For many choices the complexity is usually $O(kmn)$, and can be further reduced using basically any improvements for the base factorization component. This cost can be further reduced if the input is sparse or has other special structures. Therefore the MEMF algorithm can be scaled up to large matrices if k the rank of factorization is small. Further, accelerated factorizations like (Nguyen et al. (2009)) can be used in MEMF easily. The number of iteration required depends on the specific problem. Empirically we found that if the outliers distinguish themselves from normal data and the value of λ is proper, the convergence is very fast.

An important issue in MEMF is how to choose the value of parameter λ. λ is important because it dictates the threshold to determine a residual as an outlier, and controls the number of outliers. Specifying λ a priori is often difficult, so we compute a *path* of solutions for different values of λ. The basic strategy is to use the *warm-start* technique, in the hope that the solution for some λ_0 is close to the solution for a slightly smaller λ_1. In practice we found that this strategy works well.

3.3 Related Work and Discussion

PCA and in a more general sense subspace learning is one of the most widely used methods in data mining and machine learning. The robust subspace learning methods are also of great value in practical situations. One direction of learning robust subspaces is to directly replace the L_2 error measure with some robust alternatives. For example, Hawkins et al. (2001); Ke and Kanade (2005); Wright et al. (2009) use the L_1-norm; la Torre and

Black (2003); Nguyen and la Torre (2009) use the *Geman-McClure* loss and exponentially decaying loss. These algorithms usually need specifically designed optimization algorithms, which can be quite complicated. MEMF on the other hand can use plug-in components and takes advantage of the mature optimizers. Besides, by decomposing the errors, MEMF is able to easily use various outlier measurements, including the L_0-norm and structured norms, as well as the constraints on the factorization.

In recent years, the convex relaxation of low-rank factorizations has become popular in the fields of machine learning and compressive sensing. It is shown that the constraint on the rank of a matrix can be relaxed to the constraint on the *nuclear norm* (sum of singular values) $\|\cdot\|_*$ (Fazel (2002)) of the matrix. The resulting problem becomes convex and certain optimality were proved in Candès and Tao (2009); Mazumder et al. (2009); Wright et al. (2009).

Wright et al. (2009) firstly proposed *Robust PCA* (RPCA) as a way of robust factorization using the convex relaxation. A parallel work called *stable principal component analysis* (SPCA) was recently proposed by Zhou et al. (2010). In SPCA, the authors measure Gaussian errors and outliers separately. They solve a convex problem by simultaneously minimizing the regular Gaussian error, the outliers measured by L_1-norm , and the nuclear norm of reconstruction matrix. To make a clear comparison of the algorithms, we list the core problem they are solving as below:

- Direct optimization:
$$\min_{\mathbf{N}} \|\mathbf{X}-\mathbf{N}\|_1, s.t. \quad rank\,(\mathbf{N}) = K$$
- RPCA:
$$\min_{\mathbf{N}} \|\mathbf{X}-\mathbf{N}\|_1, s.t. \quad \|\mathbf{N}\|_* \leqslant K$$
- SPCA:
$$\min_{\mathbf{N},\mathbf{O}} \|\mathbf{X}-\mathbf{N}-\mathbf{O}\|_F^2 + \lambda\|\mathbf{O}\|_1, s.t. \quad \|\mathbf{N}\|_* \leqslant K$$
- MEMF:
$$\min_{\mathbf{N},\mathbf{O}} \|\mathbf{X}-\mathbf{N}-\mathbf{O}\|_F^2 + \lambda\|\mathbf{O}\|_r, s.t. \quad rank\,(\mathbf{N}) = K, \mathbf{N} \in \mathcal{D}_{\mathbf{N}}$$

where $\mathbf{N}$ is the low-rank approximation to the data, and $\mathcal{D}_{\mathbf{N}}$ represents the constraints on this approximation.

By comparison, we can see that MEMF extends SPCA in two ways. First, instead of just the L_1-norm, we propose to use a general class of outlier measurements the L_r-norm, including the L_0-norm and structured norms. Secondly, we adopt the strict rank constraint instead of the convex relaxation. By doing this, we obtain the freedom to use various factorization components, which may give us better efficiency and interpretability. These two extensions enable us to tailor the algorithm to incorporate prior knowledge for better performance, with the cost of losing convexity. It should be noted that the problems solved by SPCA and MEMF are very similar, such that we can use the convex relaxation in MEMF, or use the more general L_r-norm in SPCA. Therefore, we consider MEMF to be a generalization of SPCA.

One limitation of MEMF is that the factorization rank k has to be specified by the user. In fact similar issues virtually exist for many factorization problems. In practice, the value of k is often determined by knowledge, heuristics, and available computational resources.

3.4 Summary

The proposed *mixed-error matrix factorization* (MEMF) algorithm provides a flexible and efficient framework for robust low-rank factorizations and outlier detection. The basic motivation of MEMF is to consider the low-rank approximation error to be an additive mixture of the regular Gaussian noise and the outliers, and then measure them differently in the model. This kind of error mixing is different from what we do in probabilistic models, but it is much simpler and also effective.

The MEMF model is very flexible and can serve as a general framework for robust subspace learning. The users of MEMF can plug-in constraints on the low-rank factors, as well as their own implementations of L_2-Norm based factorization to get the corresponding robust version. This algorithm also justifies that the intuitive action of truncating large residuals in model fitting is actually optimizing certain robust measures. Further, we can design structures in the outlier measures to incorporate prior knowledge and capture the outlier patterns instead of merely individual points.

For our anomaly detection problem, MEMF can be used to enhance the low-rank decomposition methods we adopted in the subspace-based detection algorithms. MEMF can be further enhanced in several ways. For example, to inject more domain knowledge and detect more sophisticatedly structured anomalies, we can use the *fused lasso signal approximation* technique by Friedman et al. (2007) to specify local smoothness and detect outlier patterns whose shape are not pre-defined.

The experimental results will be presented in section 5.1.

4. Group Anomaly Detection

In the previous section, we focused on finding unusual data points. Nonetheless, in many applications we are more interested in finding *group anomalies*. One concept for group anomalies is just a group of individually anomalous points. A more interesting, and often more difficult to discover, case is where the individual data points are relatively normal but their behavior as a group is unusual. *The main contribution* of this section is to propose methods for discovering both kinds of group anomalies.

For astronomical data, unusual groups of objects are valuable targets for scientific research besides the individual objects. For example, spatial clusters of objects have played a role in each other's evolution and the distribution of their features gives insight into how they developed. Another example is the analysis of large particle simulations. In these systems each particle is normal, and the motion of a single particle is seldom interesting, but detect interesting phenomena (*e.g.* whirlwinds) is important. Similar problems exist in many other domains, such as text and image processing, where aggregated behaviors are of interest. Note that in all these examples, a cluster may have an unusual and interesting behavior even if each of its member points is normal.

To solve the group anomaly detection problem, we start from a standard statistical anomaly detection approach of creating a generative model for the data and then flagging

data points that are relatively unlikely to have been generated by that model. We propose a hierarchical probabilistic model for this purpose. We treat each group of points as a 'bag-of-objects', and assume that the points in each group are *exchangeable*. According to the *De Finetti*'s theorem in de Finetti (1931), the joint distribution of every infinitely exchangeable sequence of random variables can be represented with mixture models, thus we will apply a hierarchical mixture model to represent the data. Having estimated this model, posterior probabilities of the data can be used to identify anomalies.

We use a hierarchical mixture model to characterize the generative processes of data, similar to the work done in *Latent Dirichlet Allocation* (LDA) by Blei et al. (2003). In this model, we generate random variables in a top-down fashion, from global parameters, to group variables, and finally the observations. The basic assumption is that each individual data point falls into one of the several *topics*, and each group is a mixture of different topics. Further, at the group level we introduce the concept *genre* to capture different types of groups, where each genre is a distribution of topics. In the astronomical context, each topic can be interpreted as a certain type of galaxy, and each group consists of several types of galaxies. And, each genre is characterized by a typical distribution of different types of galaxies. Having these characterizations of data, we can capture the typical statistics of the data at multiple levels, and thus can detect anomalies at both the point level and the group level. In fact, our method is able to identify groups in which some member points are anomalous, *and* those in which the members points are normal on their own, but the topic distribution is unusual.

Efficient learning algorithms base on the variational EM technique are derived for this model. We demonstrate the performance of the proposed methods on synthetic data sets, and show that they are able to identify anomalies that cannot be found by other generative model based anomaly detectors. Empirical results are also shown for the our astronomical data.

We also note that in addition to detecting group anomalies, the proposed models can be useful in exploring data, thanks to their pure generative nature. By interpreting the estimated parameters, we can find out what the topics (*i.e.* the types of galaxies) are, and what genres (*i.e.* distribution of the galaxies types) we have in the data set. This information can also be useful for scientific study.

The section is structured as follows. In Section 4.1 we summarize some related work. We formally define the problem set-up in Section 4.2. The proposed models and how we can learn them are described in Section 4.3. Summary of this problem and discussions are in Section 4.4. Experimental results both on artificially generated toy problems and on real astronomical data are shown in Section 5.3.

4.1 Related Work

As mentioned before, group anomalies can be quite different from point anomalies. Group anomaly detection is not a new problem, but only a few results have been published on it. One idea is to transform each group into a point, and then apply point anomaly detectors for these groups. To do this, we need to define a set of features for the groups Chan and Mahoney (2005); Keogh et al. (2005). A problem with this approach is that it relies heavily on feature engineering, which can be domain specific and difficult. As shown later,

estimating an integral model for the generation process is in some sense learning how to transform the groups, and thus performs better. Besides, we believe that directly modeling the generative process of the data can give us more insight when exploring the data sets.

Another approach is to first identify the individual anomaly points in the data set, and then try to find aggregations of these points. Scan and segmentation methods are often used for this purpose. On image data, Hazel (2000) applied point anomaly detectors to find anomalous pixels, and then used Markov Random Fields to segment the image and the anomalous group of pixels together. Das et al. (2008) first used an anomaly detector to find interesting points, and then found subsets of the data with a high ratio of anomalous points. Das et al. (2009) proposed a scan statistic-based method to find anomalous subsets of points. In these approaches the anomalousness of a group is determined by the anomalousness of its individual member points, and thus they are not able to find the groups that are anomalous only at the group level.

From the methodology perspective, the proposed model is a natural realization of hierarchical probabilistic models. A well-know example of kind of models is the *Latent Dirichlet Allocation* (LDA) by Blei et al. (2003). In LDA, we also assume that points are organized by topics, and that the topic distribution of each group is generated from a global Dirichlet distribution, which is often set to be symmetric and thus not very informative. Interpreting it in terms of topics and genres, LDA is trying using one global non-informative genre to govern the topics distribution of all groups in the data set. On the other hand, our model allows for multiple genres, and put the emphasis on learning these genres as well as learning the topics. In this sense, the proposed model extends LDA so that the topic distributions are generated from a mixture of Dirichlet distributions.

Recently, the *Pachinko allocation model* (PAM) by Li and McCallum (2006) was proposed to model more complex distribution of topics. In PAM, the topic for each word is select from a multi-level mixture of multinomial distributions instead of a single multinomial as in LDA. Using this extra sophistication PAM is able to capture complex correlations among topics. The PAM model and our proposal are similar in that they allow the model to use multiple topic distributions. The difference is that in PAM the selection of topic distribution happens at the word level, while in our model it happens at the documents level. Intuitively, this means that PAM focuses on the correlation of topics and we care more about the aggregation behavior at the group level.

4.2 Formal Problem Definition

In this section we define formally our problem. For simplicity we will explain the set-up by borrowing terms from astronomy, but our solution to this problem can be used anywhere where the observations can be naturally clustered into groups.

Assume that we have M groups of objects, they are denoted by $\mathbf{G}_1, \ldots, \mathbf{G}_M$. Each group $\mathbf{G}_m$ consists of N_m objects, denoted by $x_{mn} \in \mathbb{R}^d$, $n = 1, \ldots, N_m$. These are our observations, *e.g.* x_{mn} is the $d = 500$ dimensional spectra of the n-th galaxy in the m-th galaxy group, where these galaxy groups were created based on the spatial locations of the galaxies. Assume further that each object (galaxy) x_{mn} belongs to one of the K different *topics* (galaxy types), and if we know its source topic $z_{mn} \in \{1, \ldots, K\}$, then $x_{mn} \sim P(\cdot|\beta_{z_{mn}})$, where $\beta = \{\beta_k\}_{k=1}^K$ is a dictionary of the parameters for different topics'

observation models. For example when $K = 3$, then we might think of these topics as 'red', 'blue', and 'spiky' galaxies. Each group $\mathbf{G}_m$ is a set of N_m objects which are from the K topics. Introduce the $\mathbb{S}^K = \{s \in \mathbb{R}^K | s_k \geq 0, \sum_{k=1}^K s_k = 1\}$ notation for the K-dimensional probability simplex. Then we use $\theta_m \in \mathbb{S}^K$ to denote the distribution of topics in G_m.

Now we ask the question for group $\mathbf{G}_m$, do the galaxies look normal (from the 3 topics)? And does the distribution of these 3 topics (red, blue, and spiky galaxies) looks normal? In the following sections we propose a generative probabilistic model to help answer this question and detect anomalous groups.

Before proposing the model, we first describe the LDA model as the background. In the original LDA model the data set is a text corpus, that is a collection of documents $\{\mathbf{G}_m\}_{m=1,\cdots,M}$, each of which is a set of N_m words. The number of distinct words is d. LDA assumes that each document is a random mixture over the topics sampled from a Dirichlet distribution, and each topic is characterized by a multinomial distribution over words. To be more formal, let $\mathcal{D}ir(\alpha)$ denote the Dirichlet distribution with parameter α, and let $\mathcal{M}(\theta)$ be the multinomial distribution with parameters $\theta \in \mathbb{S}^K$. In LDA, given the global hyper-parameters $\alpha \in \mathbb{R}_+^K$, for each of the M group we first generate $\theta_m \in \mathbb{S}^K$ from $\mathcal{D}ir(\alpha)$. Having θ_m, for each of the words in group m we generate $z_{mn} \sim \mathcal{M}(\theta_m)$ indicating which topic is active. Finally, words are generated from $x_{mn} \sim \mathcal{M}(\beta_{z_{mn}})$, where $\beta = \{\beta_1, \ldots, \beta_K\} (\beta_k \in \mathbb{S}^d)$ is a dictionary of parameters for K d-dimensional multinomial distributions. Having the above set up, we can estimate the model parameters α, β and latent variables θ, Z given the data.

4.3 The Dirichlet Genre Model

In this section we describe how we extend the LDA model to get a generative model that describes the normal behavior of the group data, and then we show how we can detect anomalous groups using this new model.

The first step is to specify the observation model. While LDA has been shown to be very successful for modeling discrete data like text corpora, it is not suitable for real, vector valued observations, such as the spectra in our astronomical data. Thus instead of using the multinomial distribution $\mathcal{M}(\beta_{z_{mn}})$, we assume that $\beta_k = \{\mu_k, \Sigma_k\}$ is the parameters for one Gaussian, and the observations are generated from $x_{mn} \sim \mathcal{N}(\beta_{z_{mn}}) = \mathcal{N}(\mu_{z_{mn}}, \Sigma_{z_{mn}})$. In other words, each topic is a Gaussian distribution of spectra in our model. But we should note that the proposed model in general works for any observation model.

Another problem with the LDA model is that it has only one global Dirichlet distribution to generate the topic distributions θ_m. In other words, in LDA model there is only one topic distribution (considering the exchangeability of topics) that is most probable under the model. This is a serious limitation when we are interested in learning the distribution of topics, especially when the topic distributions in all the groups are diverse and complex (*e.g.* multi-modal).

In order to profile the topic distributions in the data set, we need to further introduce more variables and distributions for the concept *genres*. Similar to topics which are typical distributions of points, the genres are typical distributions of the topics. We assume that there are T genres in the data set, and each genre is represented by a Dirichlet distribution. In the generative process, we will first sample one genre *i.e.* Dirichlet distribution for each

group, and then sample the topic distributions according to the genres. We call this model the *Dirichlet Genre Model* (DGM). We will show that the DGM is suitable for modeling our group data and can answer the questions we asked in section 4.2.

4.3.1 Model Specification

First we summarize the Dirichlet Genre Model (DGM) using the graphical model in Figure 2 and the generative process in Algorithm 2.

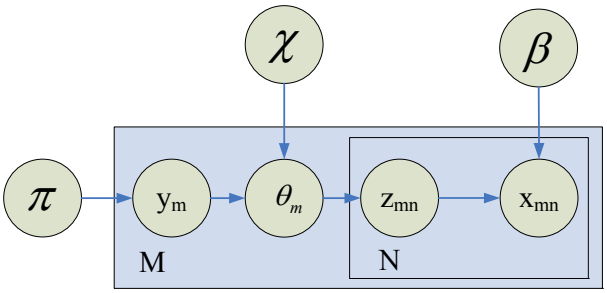


Figure 2: The graphical model for the Dirichlet Genre model.

Algorithm 2 Generative process for the Dirichlet Genre model

for group $m = 1$ to M **do**
- Choose a genre $\{1, \ldots, T\} \ni y_m \sim \mathcal{M}(\pi)$, and get the genre parameter χ_{y_m}.
- Choose a topic distribution $\mathbb{S}^K \ni \theta_m \sim \mathcal{D}ir(\chi_{y_m})$.

for points $n = 1$ to N_m **do**
- Choose a topic $\{1, \ldots, K\} \ni z_{mn} \sim \mathcal{M}(\theta_m)$, and get the topic parameter $\beta_{z_{mn}}$
- Generate a point observation $x_{mn} \sim P(\cdot|\beta_{z_{mn}})$.

end for

end for

In DGM, we have three parameters π, χ, β and the $\{y_m\}, \{\theta_m\}, \{z_{mn}\}$ are the latent variables. $\pi \in \mathbb{S}^T$ is the multinomial parameter for the distribution of genres. $\chi = \{\chi_1, \cdots, \chi_T\}$ is a dictionary of Dirichlet parameters for the genres, and $\chi_t \in \mathbb{R}_+^K$ is the parameter for the t-th genre. β as before the dictionary of parameters for the topics. As for the latent variables, y_m is the genre of group m, θ_m is the topic distribution, and z_{mn} is the topic of the n-th point.

Our strategy for group anomaly detection is as follows. Using the training set $\{\mathbf{G}_m\}$ we first learn the parameters $\Theta = \{\pi, \chi, \beta\}$ of the model. Assuming that anomalies are just a small part of the data, this model will capture the normal behavior of the data. If a group $\mathbf{G}$ is not compatible with our model, then it will lead to a small likelihood under this model. Then we can detect it as an anomalous group.

Unfortunately, learning the parameters of DGM and calculating the likelihood function, as in many hierarchical models, is intractable, thus we resort to *variational EM* methods (Jordan (1999)) for inference and learning.

4.3.2 Inference and Learning

For the sake of brevity, introduce the shorthands $\mathbf{G}_m = \{x_{mn}\}_{n=1}^{N_m}$, and $\mathbf{z}_m = \{z_{mn}\}_{n=1}^{N_m}$. Given the observations and latent variables, the complete likelihood of a group $\mathbf{G}_m$ under DGM is as follows:

$$\begin{aligned} &P\left(y_m, \theta_m, \mathbf{z}_m, \mathbf{G}_m | \pi, \chi, \beta\right) = P\left(y_m|\pi\right) P\left(\theta_m|\chi, y_m\right) \prod_{n=1}^{N_m} P\left(z_{mn}|\theta_m\right) P\left(\mathbf{x}_{mn}|z_{mn}, \beta\right) \\ &= \mathcal{M}\left(y_m|\pi\right) \mathcal{D}ir\left(\theta_m|\chi, y_m\right) \prod_{n=1}^{N_m} \mathcal{M}\left(z_{mn}|\theta_m\right) P\left(\mathbf{x}_{mn}|z_{mn}, \beta\right) \\ &= \pi_{y_m} \frac{1}{Z(\chi_{y_m})} \prod_{k=1}^{K} \theta_{mk}^{\chi_{y_m,k}-1} \prod_{n=1}^{N_m} \theta_{m,z_{mn}} P\left(\mathbf{x}_{mn}|z_{mn}, \beta\right), \end{aligned} \tag{12}$$

in which $Z(\cdot)$ is the partition function for the Dirichlet distribution.

The marginal likelihood of the observations $\mathbf{G}_m$ can be calculated as

$$\begin{aligned} P\left(\mathbf{G}_m|\pi, \chi, \beta\right) &= \sum_{y_m} \int_{\theta_m} \sum_{z_m} P\left(y_m, \theta_m, z_m, G_m\right) d\theta_m \\ &= \sum_t \pi_t \frac{1}{Z\left(\chi_t\right)} \int_{\theta_m} \prod_{k=1}^{K} \theta_{mk}^{\chi_{tk}-1} \prod_{n=1}^{N_m} \sum_{k=1}^{K} \theta_{mk} P\left(\mathbf{x}_{mn}|\beta_k\right) d\theta_m \end{aligned} \tag{13}$$

To learn the parameters $\{\pi, \chi, \beta\}$ by using maximum likelihood estimation, our task would be to calculate

$$\pi, \chi, \beta = \arg\max_{\pi,\chi,\beta} \prod_{m=1}^{M} P(\mathbf{G}_m|\pi, \chi, \beta) \tag{14}$$

The conventional way of doing this is to use the *EM* algorithm in Mclachlan and Krishnan (1996), in which we first estimate the expectations of the latent variables, and then maximize expected complete likelihood. However, the interdependence between variables in DGM makes direct EM methods intractable. Thus we make use of the variational approach. That is, instead of maximizing the exact likelihood, we will only maximize a lower bound of it.

Denote the parameters by $\Theta = \{\pi, \chi, \beta\}$. According to the Jensen inequality, for any distribution $q(y, \theta, z) = \prod_{m=1}^{M} q_m(y_m, \theta_m, z_m)$ we have that

$$\begin{aligned} &\sum_{m=1}^{M} \log P(\mathbf{G}_m|\Theta) \\ &\quad \geq \sum_{m=1}^{M} \int \mathrm{d}(y, \theta, z) q_m(y, \theta, z) \log \frac{P(y, \theta, z, \mathbf{G}_m|\Theta)}{q_m(y, \theta, z)} \\ &\quad = \sum_{m=1}^{M} \mathbb{E}_{q_m}[\log P(y, \theta, z, \mathbf{G}_m|\Theta)] - \mathbb{E}_{q_m}[\log q_m(y, \theta, z)], \end{aligned}$$

[4]with equality iff $q(y,\theta,z) = P(y,\theta,z|\{G_m\},\Theta)$. Since the marginal distribution (13) has a difficult, intractable form, instead of the direct maximization of $\sum_{m=1}^{M}\log P(\mathbf{G}_m|\Theta)$ we will only seek to maximize this lower bound of the data marginal and solve the problem

$$\Theta, q = \arg\max_{\Theta,q} \sum_{m=1}^{M} \mathbb{E}_{q_m}[\log P(Y,Z,\mathbf{G}_m|\Theta)] - \mathbb{E}_{q_m}[\log q_m], \tag{15}$$

where we use the surrogate distribution q in a special decomposed parametric form as

$$\begin{aligned} &q\left(y_m,\theta_m,z_m|\tau_m,\gamma_m,\phi_m\right) = q\left(y_m|\tau_m\right) q\left(\theta_m|\gamma_m\right) \prod_{n=1}^{N_m} q\left(z_{mn}|\phi_{mn}\right) \\ &= \mathcal{M}\left(y_m;\tau_m\right) \mathcal{D}ir\left(\theta_m;\gamma_m\right) \prod_{n=1}^{N_m} \mathcal{M}\left(z_{mn};\phi_{mn}\right). \end{aligned} \tag{16}$$

Here $\tau_m \in \mathbb{S}^T$, $\gamma \in \mathbb{R}_+^K$ and $\phi_{mn} \in \mathbb{S}^K$ are the variational parameters. Using Eq. 12 and Eq. 15, we have that the variational EM problem we need to solve is

$$\arg\max_{\{\tau_m\},\{\gamma_m\},\{\phi_m\},\Theta} \sum_{m=1}^{M} L_m\left(\tau_m,\gamma_m,\phi_m,\Theta\right), \tag{17}$$

where L_m has the following form:

$$\begin{aligned} &L_m\left(\tau_m,\gamma_m,\phi_m;\pi,\chi,\beta\right) = E_q \log\left(y,\theta,z,G|\pi,\chi,\beta\right) - E_q \log q\left(y,\theta,z\right) \\ &= E_q \log P\left(y_m|\pi\right) + E_q \log P\left(\theta_m|\chi,y_m\right) + E_q \sum_{n=1}^{N_m} \log P\left(z_{mn}|\theta_m\right) + E_q \sum_{n=1}^{N_m} \log P\left(x_{mn}|z_{mn},\beta\right) \\ &- E_q \log q\left(y_m|\tau_m\right) - E_q \log q\left(\theta_m|\gamma_m\right) - E_q \sum_{n=1}^{N_m} \log q\left(z_{mn}|\phi_{mn}\right) \\ &= \sum_{t=1}^{T} \tau_{mt} \log \pi_t - \sum_{t=1}^{T} \tau_{mt} \log \tau_{mt} \\ &+ \sum_{t=1}^{T} \tau_{mt} \left(\log\Gamma\left(\sum_{k=1}^{K} \chi_{tk}\right) - \sum_{k=1}^{K} \log\Gamma\left(\chi_{tk}\right) + \sum_{k=1}^{K} \left(\chi_{tk}-1\right) \left(\Psi\left(\gamma_{mk}\right) - \Psi\left(\sum_{i=1}^{K} \gamma_{mi}\right)\right)\right) \\ &+ \sum_{n=1}^{N} \sum_{k=1}^{K} \phi_{mnk} \left(\Psi\left(\gamma_{mk}\right) - \Psi\left(\sum_{i=1}^{K} \gamma_{mi}\right)\right) \\ &+ \sum_{n=1}^{N} \sum_{k=1}^{K} \phi_{mnk} \log P\left(x_{mn}|\beta_k\right) - \sum_{n=1}^{N_m} \sum_{k=1}^{K} \phi_{mnk} \log \phi_{mnk} \\ &- \log\Gamma\left(\sum_{k=1}^{K} \gamma_{mk}\right) + \sum_{k=1}^{K} \log\Gamma\left(\gamma_{mk}\right) - \sum_{k=1}^{K} \left(\gamma_{mk}-1\right) \left(\Psi\left(\gamma_{mk}\right) - \Psi\left(\sum_{i=1}^{K} \gamma_{mi}\right)\right) \end{aligned} \tag{18}$$

4. $\mathbb{E}_q$ denotes the expected value w.r.t. distribution q.

where $\Gamma(\cdot)$ is the gamma function, and $\Psi(\cdot)$ is the digamma function (first derivative of $\log\Gamma$).

By solving the problem 17 we can get the model and variational parameters. The way in which variational EM works is to update one of $\{\tau_m\}, \{\gamma_m\}, \{\phi_m\}, \pi, \chi, \beta$ at a time while keeping others fixed at their current value. This procedure is in fact using block coordinate descent to optimize the low-bound we constructed for the true likelihood of data. Here we just show the end results, the details of the calculations are omitted.

The variational parameters can be updated as below (given the observations and all other parameters)

$$\begin{aligned}
\phi_{mnk} &\propto \exp\left(\Psi\left(\gamma_{mi}\right) - \Psi\left(\sum_{k=1}^{K}\gamma_{mi}\right) + \log P\left(x_n|\beta_k\right)\right) \\
\gamma_{mk} &= \sum_{t=1}^{T}\tau_{mt}\chi_{tk} + \sum_{n=1}^{N_m}\phi_{mnk} \\
\tau_{mt} &\propto \exp\left(\log\pi_t + \log\Gamma\left(\sum_{k=1}^{K}\chi_{tk}\right) - \sum_{k=1}^{K}\log\Gamma\left(\chi_{tk}\right) + \sum_{k=1}^{K}\left(\chi_{tk}-1\right)\left(\Psi\left(\gamma_{mk}\right) - \Psi\left(\sum_{i=1}^{K}\gamma_{mi}\right)\right)\right)
\end{aligned} \tag{19}$$

Note that the parameters ϕ_{mn} and τ_m need to be normalized so that they are on simplexes $\mathbb{S}^K$ and $\mathbb{S}^T$ respectively.

The model parameters can be similarly updated as below. The value of π can be directly calculated as

$$\pi = \frac{1}{M}\sum_{m=1}^{M}\tau_m. \tag{20}$$

To calculate β, we need to solve

$$\arg\max_{\beta_k}\sum_{m=1}^{M}\sum_{n=1}^{N_m}\sum_{k=1}^{K}\phi_{mnk}\log P\left(x_{mn}|\beta_k\right) \tag{21}$$

which is just a maximum likelihood estimation problem given weighted data. This can be easily done as in EM algorithms such the *Gaussian mixture model*.

To update the value of χ we can use the constrained *Newton-Raphson* method. Concretely, we update one χ_t at time, using the following objective, gradient, and Hessian:

$$\begin{aligned}
f_{\chi t} &= \left(\log\Gamma\left(\sum_{k=1}^{K}\chi_{tk}\right) - \sum_{k=1}^{K}\log\Gamma\left(\chi_{tk}\right)\right)\sum_{m=1}^{M}\tau_{mt} \\
&\quad + \sum_{m=1}^{M}\tau_{mt}\sum_{k=1}^{K}\left(\chi_{tk}-1\right)\left(\Psi\left(\gamma_{mk}\right) - \Psi\left(\sum_{i=1}^{K}\gamma_{mi}\right)\right) \\
\frac{\partial f_{\chi t}}{\partial\chi_{tk}} &= \left(\Psi\left(\sum_{i=1}^{K}\chi_{ti}\right) - \Psi\left(\chi_{tk}\right)\right)\sum_{m=1}^{M}\tau_{mt} + \sum_{m=1}^{M}\tau_{mt}\left(\Psi\left(\gamma_{mk}\right) - \Psi\left(\sum_{i=1}^{K}\gamma_{mi}\right)\right) \\
\frac{\partial^2 f_{\chi t}}{\partial\chi_{ti}\partial\chi_{tj}} &= \left(\delta\left(i,j\right)\Psi'\left(\chi_{ti}\right) - \Psi'\left(\sum_{k=1}^{K}\chi_{tk}\right)\right)\sum_{m=1}^{M}\tau_{mt}
\end{aligned} \tag{22}$$

By applying the update equations described above, we can estimate both the model parameters and the variational parameters. It is clear that these updates is guaranteed to improve our lower bound for the marginal probability of the data, so this algorithm is going to converge. However, since the problem is not convex, we need to try multiple times using random initial estimates to get a good result.

Due to the limitation of variational approximations, we cannot know how good is the solution given by this variational EM procedure. From our current empirical results we observe the approximation is quite good when the number of topics and genres is not large.

4.3.3 Anomaly Scoring

Having estimated the mode and variational parameters for DGM, we score each group G_m to rank their anomalousness. Here we propose to use two probabilities as the anomaly scores.

One choice of anomaly score is

$$\log P(G_m|\Theta) = \log \sum_t \pi_t \frac{1}{Z(\chi_t)} \int_{\theta_m} \prod_{k=1}^{K} \theta_{mk}^{\chi_{tk}-1} \prod_{n=1}^{N_m} \sum_{k=1}^{K} \theta_{mk} P(\mathbf{x}_{mn}|\beta_k) d\theta_m,$$

which is just the likelihood that G_m is generated from the model. This probability accounts for the anomalousness of both the points and the topic distribution for this group. Its value will be low if either this group contains a point that does not belong to any of the topics, or the topic distribution in this group is not compatible with the normal behavior specified by π and χ. Unfortunately, probability $P(G_m|\Theta)$ is hard to compute analytically. So we use *Monte Carlo* method to compute the integral over θ_m.

While the score $\log P(G_m|\Theta)$ is intuitive and is able to detect both group and point anomalies, it has the disadvantage that the effects of these two anomalies are not separated. Therefore, it is not able to answer query "which group has the most unusual composition of topics?". In fact, we can see that a group whose N_m is large or a group that contains bad point anomalies, the anomaly score of G_m will be dominated by the point scores $P(x_{mn}|\beta_k)$, and the group-level scores, which is of our primary interest, becomes invisible. One way to alleviate the size effect is to use *perplexity*, which is obtained by normalizing the above score by the group size. But it is still unable to suppress the effect of point anomalies.

To solve this problem, we use the probability $\log P(\theta_m|\Theta)$ as the anomaly score. Compared to $\log P(G_m|\Theta)$, this score only focus on the distribution of topics in the group. The difficulty left is that θ_m is a latent variable that we do not observe. The solution is to first estimate the distribution of $P(\theta_m|G_m, \Theta)$, and then use the expectation

$$\mathbb{E}_{\theta_m \sim P(\theta_m|G_m,\Theta)} (\log P(\theta_m|\Theta)) = \int_{\theta_m} P(\theta_m|G_m, \Theta) \log P(\theta_m|\Theta) d\theta. \tag{23}$$

To further simplify the computation, we can use the variational distribution $q(\theta_m|\gamma_m)$ as an approximation to $P(\theta_m|G_m, \Theta)$, and use the *Monte Carlo* method to compute the integral. We call this quantity the *genre score*, since it indicates if a group belongs to a normal genre.

Although we have two scoring function targeting at point anomalies and group anomalies respectively, currently we are not able to specify the balance between these two. In practice, we recommend detect the point anomalies first, remove them, and then apply the group anomaly detectors.

4.3.4 Model Selection

One major limitation of the DGM is that T the number of genres and K the number of topics need to be assigned by the user. To automatically determine their values, a simple way is to score different models using methods such as BIC Schwarz (1974) or AIC Akaike (1974). The definition of BIC score is given by $BIC(X, \Theta) = \ln L(X, \Theta) - \frac{1}{2} \ln(|X|) |\Theta|$, where $|\cdot|$ stands for the number of free parameters. We can directly use these two scoring functions to perform a two dimensional search for the best values for T and K. Since we usually do not know about the correlations among the parameters, the weight of the second term in BIC is adjusted so that it achieves best performance on the validation set.

Another way to automatically control the complexity of the model is to use non-parametric Bayesian methods. Specifically, we can use a *Dirichlet Process* (DP) by Ferguson (1973) to replace the role of π, χ so that the model can potentially use infinitely many number of genres. This model is still under development. However, we noted that handling both T and K using DP is difficult because it will need a 'DP' whose samples are DPs.

4.4 Summary

In this section we investigated how to use hierarchical probabilistic models for the group anomaly detection problem. Following the paradigm of topic modeling, the *Dirichlet Genre Model* (DGM) is proposed to capture the generative process of both the individual points and the groups. Two level of concepts, *genres* and *topics*, are proposed and used to characterize both the distribution of points and the composition of groups. In this way, we can detect various group anomalies.

Two scoring function are proposed as the anomaly score for each group. The first one is the likelihood of the whole group. This score gives an overall measurement for the group, but also lacks of the ability to distinguish "aggregation of anomalous points" and "anomalous aggregation of points". The second scoring function on the other hand focus on the topic distribution in the group and is effective finding anomalies of the second type. In the future, we are hoping to find a trade-off between these two scores, so that the user can specify which type of group anomalies his/her interest is on.

Plenty of future work can be done for DGM. First, currently the model is not Bayesian *i.e.* we get point estimates for all the parameters. We can easily incorporate Bayesian treatment to make the model fitting more stable. Second, as mentioned in section 4.3.4, we can further use non-parametric Bayesian methods to eliminate the parameters T or even K. Thirdly, though efficient, the variational learning method has been shown to be inaccurate (*e.g.* Minka and Lafferty (2002)). We can use other methods such as *Markov Chain Monte Carlo* (MCMC) or *Expectation Propagation* (EP) (Minka and Lafferty (2002)) for learning. The forth interesting problem is how to fit robust Dirichlet distributions in DGM. In practice we observe that the estimated χ parameter tends to cover all topic distributions in the data including the anomalies, making the genres overly smooth. For our anomaly detection purpose, the enhancement on robustness is necessary and expected to improve performance.

The experimental results of the DGM are shown in section 5.3.

5. Experiments

5.1 Point Anomaly Detection using MEMF

In this section we show the empirical effectiveness of MEMF on both simulation and real-world data sets. We compare our method to the following the state-of-the-art competitors:

- **L_1 Factorization (L1F) by Ke and Kanade (2005)** We use the Matlab's *Linear Programming* as the base solver.
- **Robust PCA (RPCA) by Wright et al. (2009)** We use the code from the original author[5]. The faster "inexact" implementation is used for speed.
- **Stable PCA (SPCA) by Zhou et al. (2010)** We implemented SPCA in Matlab and use block coordinate descent to optimize.

The partial SVD results are also provided as a baseline. To see more details about these methods please refer to section 3.3.

The MEMF algorithms are implemented in Matlab. Partial SVD is done using PROPACK by Larsen. For all algorithms, we terminate the iteration when the change of the objective function value from (5) $\frac{f^{t-1}-f^t}{f^{t-1}} \leq \varepsilon = 10^{-5}$.

5.1.1 Simulation Data

First, we compare different robust factorization methods in detail on simulated data sets. Following the set up in the work of Zhou et al. (2010), the test matrix is constructed as the sum of the background $\mathbf{G_0}$, the noise $\mathbf{E_0}$, and the outliers $\mathbf{O_0}$. $\mathbf{G_0} = \mathbf{U_0^T V_0} \in \mathbb{R}^{\mathbf{m}\times\mathbf{m}}$ where $\mathbf{U_0}, \mathbf{V_0} \in \mathbb{R}^{\mathbf{r}\times\mathbf{m}}$ have *i.i.d.* entries from Gaussian $\mathcal{N}\left(0, \sigma_n^2\right)$. Noise $\mathbf{E_0} \in \mathbb{R}^{\mathbf{m}\times\mathbf{m}}$ has *i.i.d* Gaussian entries from $\mathcal{N}\left(0, \sigma^2\right)$ where $\sigma = \sqrt{m}\sigma_n/10$. Outlier $\mathbf{O_0} \in \mathbb{R}^{\mathbf{m}\times\mathbf{m}}$ is sparse with s *i.i.d* entries from the uniform distribution on $[-c\sigma, c\sigma]$. Here we use $\sigma_n = 1, \sigma = \sqrt{m}/10, r = \sqrt{m}, s = 0.05m^2, c = 10$.

For MEMF, we use the un-constrained factorization mentioned in section 3.2.2. The iteration is initialized by partial SVD. The value of λ is set so that residuals larger than 3σ is thresholded. For all the competing algorithms, parameters are set to their suggested values. For SPCA we use a thresholding parameter that is equivalent to the one used in MEMF. The true rank r is specified for all the factorization algorithms.

We compare the performances on three different indices. To measure the performance of robust factorization, we compute the *root mean squared error* (RMSE) of the low-rank reconstruction w.r.t. $\mathbf{G_0}$. Since the factorization obtained by SPCA is biased due to the shrinkage of singular values, its performance was measured after "debiasing" as in Ma et al. (to appear). To measure the outlier detection performance, we compute the *average precision* when we retrieve the outlier entries according to their reconstruction residuals. Finally, the running time is measured for a speed comparison. Mean performances of 20 random runs are reported.

5. `http://perception.csl.uiuc.edu/matrix-rank/sample_code.html`

Individual Outliers First we test the performance when handling *entry-wise outliers*. In this case, the non-zero entries in $\mathbf{O}$ are uniform over all positions. This situation satisfies the assumption made by RPCA and SPCA. For MEMF, both the L_0 and L_1 norms are used to measure outliers, denoted as *MEMF-E0* and *MEMF-E1* respectively. We simulated for 10 sizes that lie uniformly in the log-space between 50 and 5000. The performance curves are shown in the first row of Figure 3.

From (a)(b) of Figure 3, we see that MEMF-E1 and MEMF-E0 performs the best in terms of both robust factorization and anomaly detection. Particularly MEMF-E0 is better than MEMF-E1 because no shrinkage is applied so the impact of outliers is minimum. SVD is clearly not robust. RPCA is doing a poor job because it is not designed to handle the ubiquitous Gaussian noise, which makes the estimated rank wrong. L1F has similar performance with MEMF-E1 in the beginning, but is too slow when $m > 100$. SPCA gives good performance similar to MEMF-E1, confirming their similarity in the L_1-norm case as discussed in section 3.3. From (c) we see that MEMF-E algorithms are much faster than convex algorithms RPCA and SPCA, and is only slightly worse than the state-of-the-art partial SVD. The slower speed of convex algorithms is mainly caused by the repeated use of partial SVD and sometimes the over-estimation of the rank. From these result, we see that MEMF is better than the competitors in both quality and speed.

Group Outliers We then examine the performance when handling grouped outliers. Here, the outliers in $\mathbf{O}$ concentrate in rows *i.e.* we first select a few random rows in $\mathbf{O}$ and then fill them with outliers. Note that now the assumption of RPCA and SPCA has been violated. To accommodate the row patterns, we add the $L_{0\text{-}1}$ and $L_{2\text{-}1}$ norms for MEMF, denoted as *MEMF-R0* and *MEMF-R1* respectively. Note that the set up of this simulation is identical to the previous one except for the positions of the outliers. The performance curves are shown in the second row of Figure 3.

A similar comparison can be observed as in the individual outlier case, but several interesting things can be observed. From (d)(e), we see that the performance of SPCA was severely compromised by the grouped outliers that inflates the estimated rank. The entry-wise MEMF-E1 and particularly MEMF-E0 are also affected. They still have low reconstruction error but do not perform well in the anomaly detection task, showing that the estimations are distorted by structured outliers. On the other hand, the structured MEMF (MEMF-R0 and MEMF-R1), having a good knowledge of the structure of outliers, show superior performance in both reconstruction and anomaly detection, especially that their detection rate almost always achieves the optimum. Structured MEMF also have speed advantage because they usually converge in fewer iterations.

Based on the results in this simulation, we conclude that MEMF-E1 provides similar performance to the state-of-the-art SPCA at a faster speed. MEMF-E0 shows better performance than the L_1-norm based methods, demonstrating the advantage of the not so often used L_0-norm. Moreover, we observe that entry-wise methods suffer from the distortion caused by grouped outliers. For this situation, MEMF-R algorithms are able to deliver much better results using outlier measurements that are designed to match the structure of the outliers. On the other hand, we observe that the convex methods are broken by these structured outliers which defies the assumption that outliers are uniformly distributed on the entries.

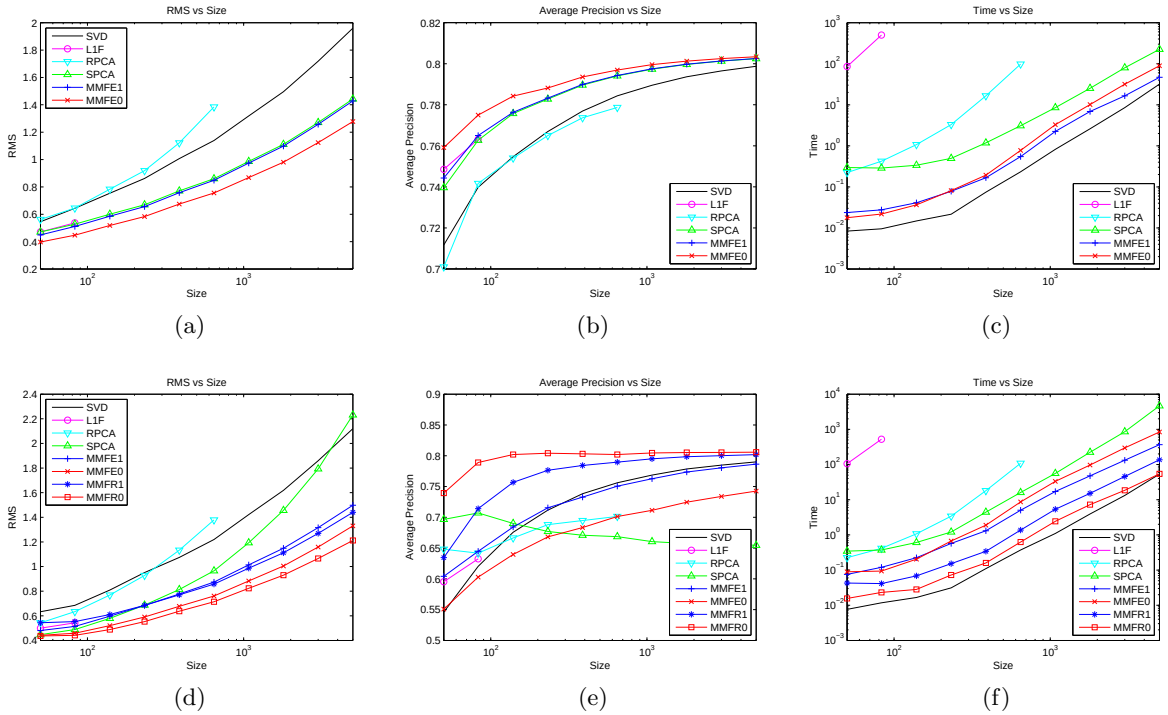


Figure 3: Robust factorization performances on simulated data. The columns are: reconstruction RMSE on the normal entries; average precision for retrieving the outliers; running time in log-scale. In the first row, the outliers are uniform distributed. In the second row, the outliers concentrate on several rows. See text for details.

5.1.2 Video Background Modeling and Activity Detection

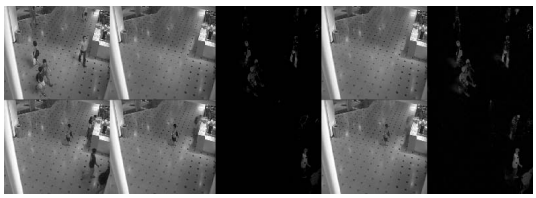
In this experiment we consider modeling the background of video clips. Estimating the background accurately is important for activity detection in videos, yet also difficult because of the variability of the background (*e.g.* due to lighting conditions) and the presence of foreground objects such as moving people. Here we assume that the background variations are of low-rank and the foreground objects are sparse outliers in the video. Then we can solve this problem using robust low-rank factorization, in which the background is modeled by the low-rank part and foreground is captured as outliers.

Video sequences "Hall" (size 128×160, frames 2100-2400), "Lobby" (size 144×176, frames 1300-1700), "Restaurant" (size 120×160, frames 2500-3000), and "ShoppingMall" (size 128×160, frames 1500-2000) from Li et al. (2004) are used. The "Hall" data contains a scene in an airport with relatively static background and many foreground activities. The "Lobby" data contains a scene in an office lobby with few foreground activities and large background variations. The "Restaurant" and "ShoppingMall" data are noisier and contain much more foreground activities. Sample images are shown in Figure 4. We stretch and stack the frames into a matrix, and compare RPCA, SPCA, MEMF-E1, MEMF-E0

(a) Results for "Hall"

(b) Results for "Lobby"

(c) Results for "Restaurant"



(d) Results for "ShoppingMall"

Figure 4: Video activity detection result frames. The columns from left to right are: the original frame, background and foreground generated by MEMF-E0, background and foreground generated by RPCA.

on this problem. The performance is measured again on the reconstruction RMSE of the background, and the average precision of retrieving foreground pixels on given ground truth frames.

We use the suggested parameters for RPCA and SPCA. The median of pixels' standard deviation is used to estimate the background Gaussian noise level σ, and we then set the thresholding parameters of MEMF and SPCA to get the residuals larger than 3σ. For SVD and MEMF models, rank-5 models are used for "Hall", "Lobby" and rank-7 models are used for "Restaurant", "ShoppingMall" to capture the background variations.

Detection results of MEMF-E0 and RPCA for some ground-truth frames are shown in Figure 4. Visually the result from both are quite good and similar. The background are well reconstructed and the detected foreground objects are correct. A close examination shows that the foreground detected by RPCA usually have more pixels than MEMF. This is because that foreground pixels that have small deviations from the background are absorbed by the Gaussian noise part of MEMF. The benefit we get is that MEMF (also SPCA) usually generates models whose ranks are much lower than RPCA's because we do not have to fit the noise in the background.

The normalized performance diagrams are shown in Figure 5, in which we re-scaled the performance values so that the largest one is 100%. On the "Hall" and "Lobby" data, we can see that the algorithms achieve similar RMSE and average precisions. This is probably because that the data here are relatively simple and the number of ground-truth frames is small. Yet, we can still see that MEMF methods are slightly better than SPCA and maintained the speed advantage. On the other hand, on the more complex "Restaurant" and "ShoppingMall" data, the MEMF-E methods achieves significant improvement, especially for the detection rate of foreground pixels. The implication is that MEMF can better re-

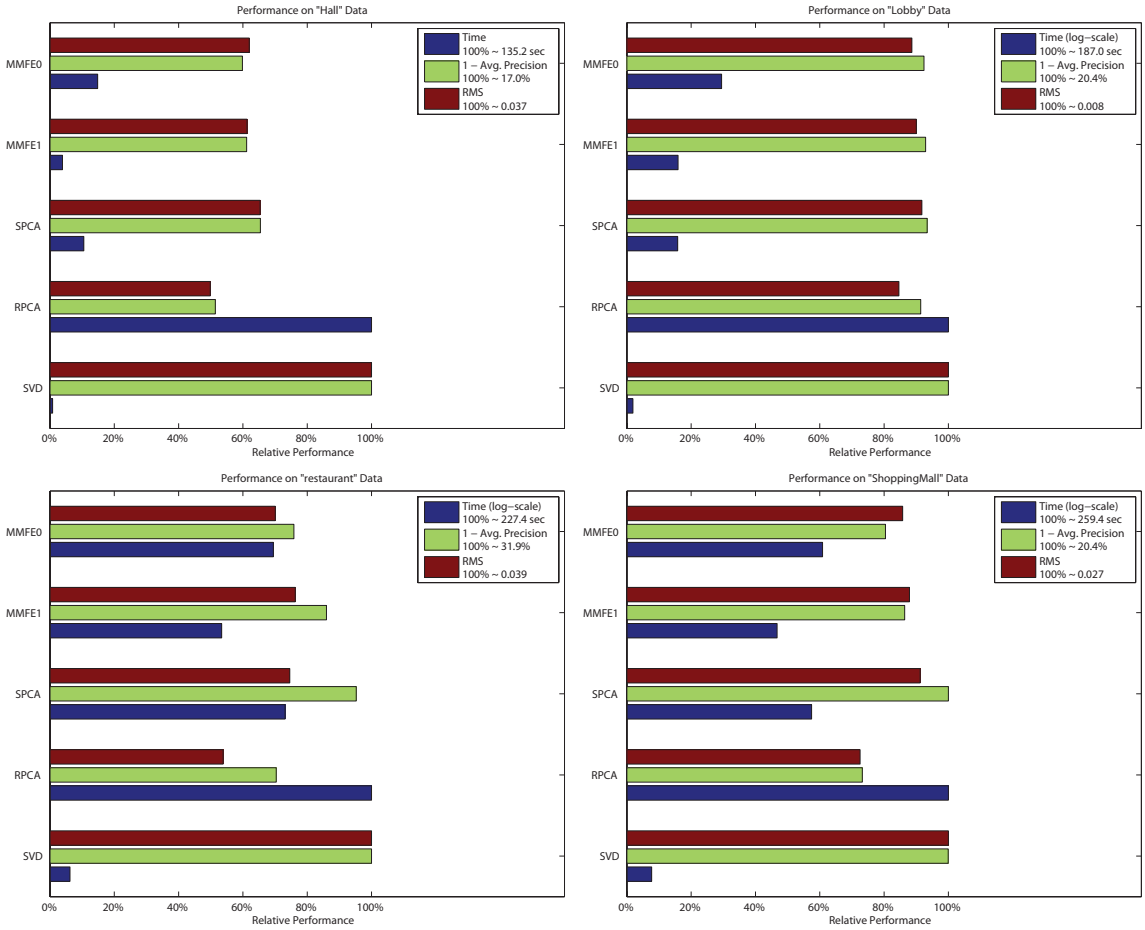


Figure 5: Video background modeling, activity detection, and running time performances. Shorter bars are better. Performance values are normalized so that the largest one is 100%.

construct the background regions that are occluded by foreground objects and not included in the RMSE measurement.

We also see that in this experiment, RPCA performs the best, which is not surprising due to its theoretical soundness. Yet the advantage comes at the price of significantly longer running time (note the time shown is in log-scale) and more complex background models. For example, the ranks of RPCA models are 150, 231, 276, and 282 for the 4 sequences respectively.

5.1.3 Text Data Clustering

In this experiment we show how MEMF help improve the NMF-based clustering algorithm for text data. We demonstrate here the robustness as well as flexibility of MEMF that is obtained by allowing constrained factorization, which is not supported by RPCA/SPCA.

The data set we adopt is a subset of 20-newsgroup[6]. We choose the *rec* topic which contains *autos*, *motorcycles*, *baseball*, and *hockey*. The TF-IDF representation is used, and each document is normalized to have a unit length. The final document-term matrix is of size 3970×8014.

Here we use the original NMF as in Lee and Seung (1999), and solve it using the fast algorithm proposed in Li and jin Zhang (2009). We use the NMF clustering method proposed by Xu et al. (2003) to partition the documents into 4 clusters. To measure the clustering performance, we first compute the confusion matrix from the class labels and cluster labels, then permute the columns to maximize the trace of this matrix, and finally the portion of documents on the diagonal is counted as the clustering accuracy.

We found that this data set is clean for clustering. The original NMF can achieve an accuracy of 0.9217, and the MEMF methods can bring it up to around 0.923. To further demonstrate the impact of outliers and how MEMF can help, we contaminate the data set with artificial outliers. Denoting the number of non-zero entries and the maximal value in the document-term matrix $\mathbf{X}$ as nz and e_{max} respectively, we randomly pick $0.01nz$ entries in $\mathbf{X}$ and set their values to random numbers from the uniform distribution on $[0, e_{max}]$. On this data set, the clustering accuracies of NMF, MEMF-E1, and MEMF-E0 are compared.

For the MEMF methods, we compute the path of accuracy for different values of parameter λ using warm start. Letting the λ_{max} be the smallest λ that picks no outliers out, the values of λ are chosen so that they lie log-uniformly between λ_{max} and $0.01\lambda_{max}$, and are denoted as $\{10, 9, \cdots, 1\}$. In each random trial, we re-add the outliers and the best NMF result from 10 random initializations are used. The mean performance and standard deviation of 20 random trials are shown in Figure 6.

The impact of outlier on the clustering result is obvious: the accuracy of NMF has dropped from 0.92 to 0.83. Yet after applying the MEMF methods with certain λs, the accuracy can be brought back to its original value like the outliers are not there. This shows the power of robust methods against outliers.

We also show the numbers of entries identified as outliers by the MEMF methods. A clear "elbow" point can be found in the curve, and that point coincides with the optimum of MEMF-E1. The interpretation is that once we use a λ that is too small, many normal data are wrongly regarded as outliers, making the number of outliers grow rapidly. This

6. http://people.csail.mit.edu/jrennie/20Newsgroups

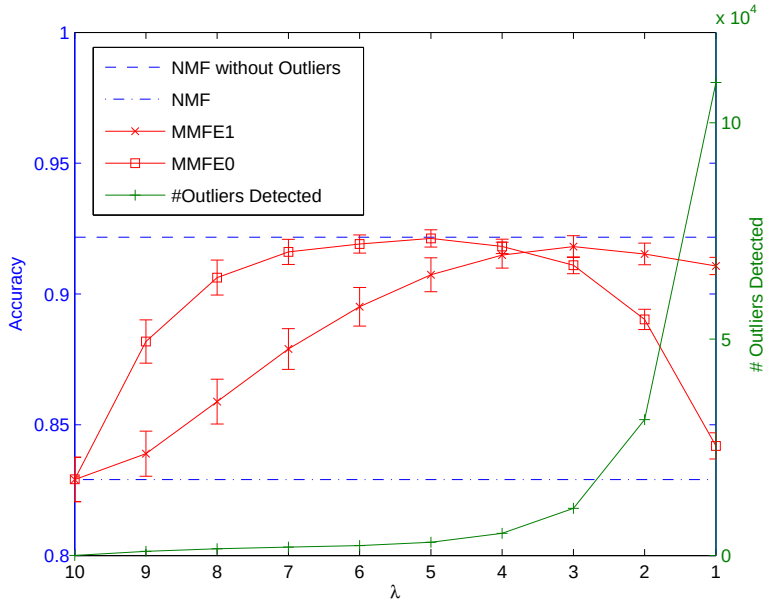


Figure 6: Accuracy path of MEMF for text clustering. MEMF methods is able to overcome the outliers and achieve the un-contaminated performance. Also shown is the number of entries identified as outliers by MEMF (there are about 2.4×10^3 true outliers). See text for details.

phenomenon gives us a useful heuristic for choosing the right λ. It is also interesting to observe the different behaviors of MEMF-E1 and MEMF-E0. MEMF-E0 can achieve higher accuracy for large λs but is unreliable when λ is small. MEMF-E1 on the other hand is more stable, because it shrinks the outliers and pushes the non-outliers back into the normal part of the model.

5.2 Point Anomaly Detection in Astronomical Data

Now we apply the MEMF algorithm to detect point anomalies on the SDSS data and compare its performance with the non-robust method. We use SVD as our base factorization method. The four MEMF algorithms and the original SVD are used. RPCA is not tested here because it does not scale well enough for the astronomical data, and SPCA in general performs very similar as MEMF-E1 when the parameters are tuned. To do detection the anomalies, we stack the objects' features as rows to form a matrix, then find a rank-k decomposition of the data matrix, and finally compute the reconstruction error using this decomposition. The anomaly scores for each object is calculated as the sum of squared errors on each row.

The stars' features $\mathcal{S}$ (the raw spectrum) are used. The resulting data matrix has a size 49529×500, and the labels for anomalies are obtained as follows. For a small subset of stars, the SIMBAD system[7] has a catalog of their categorization. We search for our stars

7. http://simbad.u-strasbg.fr/simbad/

in the SIMBAD system and label the stars that belong to a special category (not just a "star"). Then, these labeled stars are treated as anomalies. In this way, among the 49529 stars in our pool 1144 anomalies are labeled. Note that this does not give us a complete list of anomalies.

We test the performances of algorithm in multiple random runs. In each run, 10^4 stars are randomly selected from the pool. To determine k the rank used for factorization, we first do a PCA on the data and select k to keep 97% of the total variance. The values of λ the regularization coefficients of outliers are selected so that about 3% of the data are regarded as outliers. Performances are measured in both *average precision* (AP) and the *area under the ROC curve* (AUC). The results from 30 random runs are shown in Figure 7.

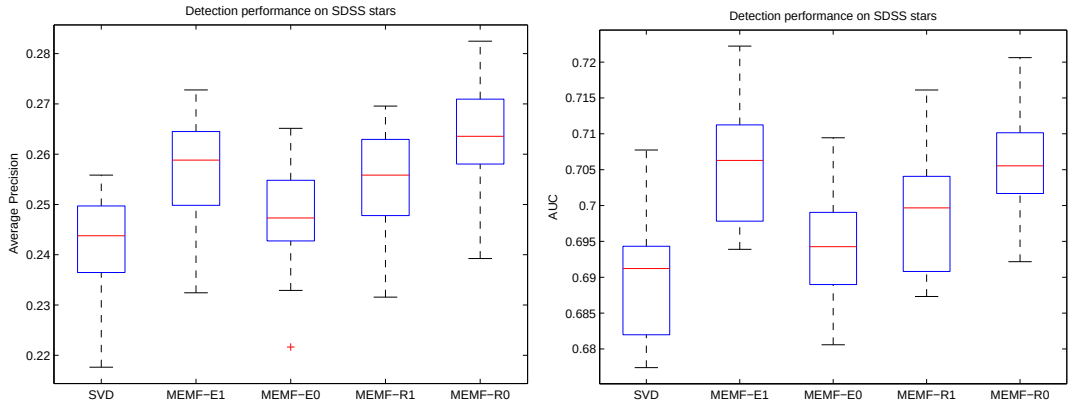


Figure 7: Point anomaly detection performance on the SDSS star data.

We can see that clear improvements are achieved by the MEMF algorithms, showing the benefit of robust modeling. Especially, we observe that the MEMF-R0 method produces the best result. We conclude that by utilizing the intrinsic structure of the data, the MEMF algorithm can find more reliable decompositions and thus gives better anomaly detection results.

5.3 Group Anomaly Detection

We show some experimental results to demonstrate the effectiveness of the proposed Dirichlet Genre Model (DG). We compared it with a simple *point anomaly* detectors: the *Gaussian mixture model* (GMM). In the experiment on astronomical data we also compared it with a histogram based method.

For the DG model, we score the anomalies using both the perplexity score and the genre score. The anomaly score of a group using GMM and KNN is just simply the mean of the anomaly scores of its member points, which can be considered variants of the perplexity. Note that all these scores are normalized by the group sizes.

5.3.1 Synthetic Problems

First, we test the effectiveness of the algorithms on some synthetic data sets. These experiments are designed particularly to demonstrate the weakness of the existing algorithms

and how our proposal solves it. Performances of DG using the genre score (DG-Genre), DG using the perplexity score (DG-Perplexity), and GMM are compared.

We generate the data sets according to the process described in Algorithm 2. The data points are sampled from three 2-dimensional isometric Gaussian components (*i.e.* $K = 3$), whose means are $[-1, -1]$, $[1, -1]$, $[0, 1]$ and the covariances are all $\Sigma = 0.1 \times \mathbf{I}_2$, where $\mathbf{I}_2$ denotes the 2D identity matrix. These components are the *'topics'*. Then we design two normal *genres* ($T = 2$), which are specified by two topic distributions ($\chi_1 \in \mathbb{S}^3$, and $\chi_2 \in \mathbb{S}^3$) respectively. We generated $M = 40$ groups, and $N_m \sim Poisson(50)$ instances in each of the groups. Note that the resulting points individually are perfectly normal *w.r.t.* other points.

To test the detection performance of these models, we inject two types of anomalies in the data set. The first kind is a group of point anomalies, which are sampled from $\mathcal{N}([0, 0], \Sigma)$ (the anomalous topic). This kind of anomaly should be easy to find for all methods. We corrupted one group with this anomaly. The second kind of injection is the group anomalies, where the individual points are normal, but they together as a group look anomalous. In this anomalous group the individual points were sampled from one of the $K = 3$ normal topics, but the distribution of these topics were different from the normal genres χ_1, χ_2. The one realization of the simulation is shown in Figure 8.

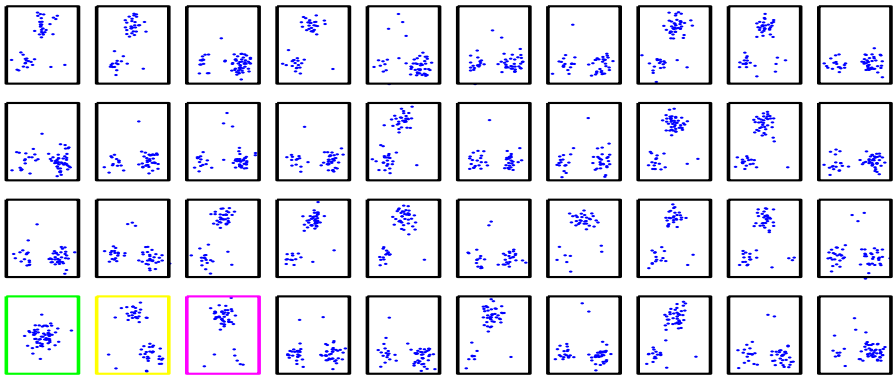


Figure 8: A simulated data set for group anomaly detection. Green box contains the point anomaly group. Yellow and magenta boxes contains the group anomaly groups. Black boxes contains normal groups.

We test the performances of different methods on a data set whose topic distributions has a clear two-modes structure *i.e.* there are two well-separated genres. Concretely the two genres have topic distributions $\chi_1 = (0.33, 0.64, 0.03)$ and $\chi_2 = (0.33, 0.03, 0.64)$ respectively, and the distribution of genres is $\pi = (0.48, 0.52)$. According to these parameters, there are two types of normal groups. One consists mainly of topics 2&3, and the other consists mainly of topics 3. We corrupted three groups of normal data using the point anomaly and the two group anomalies.

The detection results are shown in Figure 9. Here each box contains one group, and we shown 20 out of the 40 groups. The colors of the boxes are: black for normal group, green for point anomaly, and yellow/magenta for group anomaly. The instances of the groups are

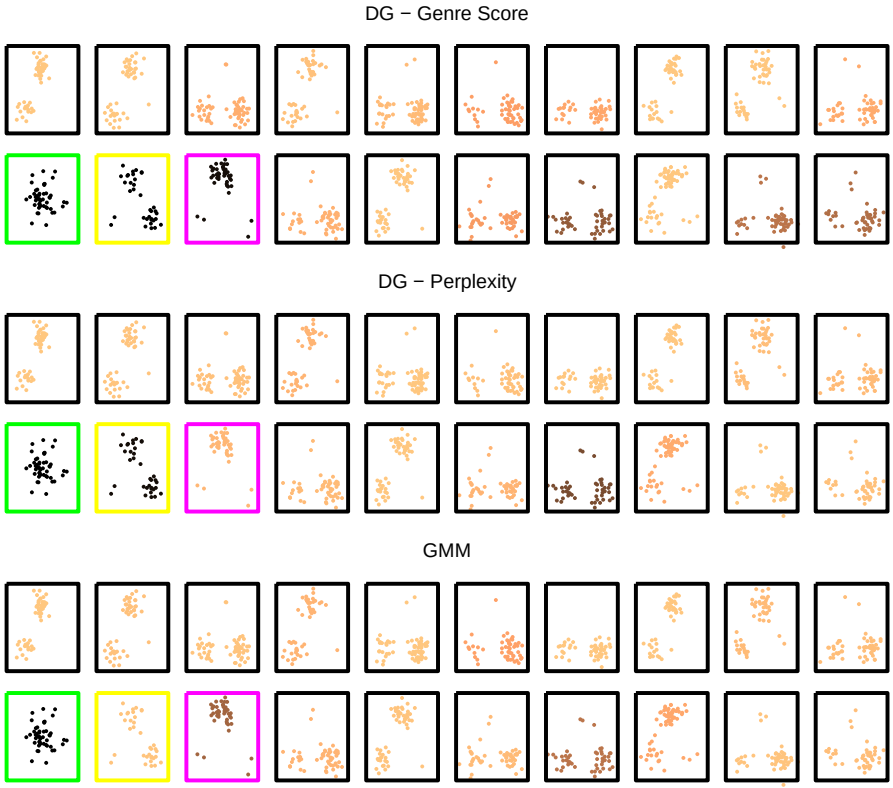


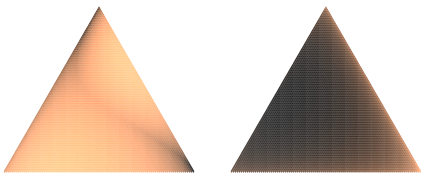
Figure 9: Detection results on a toy data set. Normal groups are in black boxes, and anomalies are in the colored boxes. Darker color means higher anomaly score.

plotted and colored according to their anomaly scores given by the corresponding algorithms (the darker colors indicate higher anomaly scores). The anomaly detection is successful, if the green, yellow/magenta boxes contain points with high anomaly scores (dark points), and the black boxes contain points with low anomaly scores (light gray points).

We can see that the group of aggregated point anomalies is easily identified by all methods. But the point-wise detectors GMM are not aware of the group anomalies, since each individual point is indeed normal. On the other hand, the proposed DG models examine both the topics and the genres, and discovers the eccentric behaviors at the group level. We note that one anomaly is not ranked top-3 by the DG model with the perplexity score. The reason is that the other group at (bottom row, 7-th column) happens to have some points with higher anomaly scores. In general the DG model with the perplexity score is not very stable due to this phenomenon. We suggest only use the genre score if interested in the aggregation behavior, since many mature point anomaly detectors exist already.

The learned genres, which are represented by a mixture of Dirichlet distributions, are shown in Figure 10(a). The triangle represents the 3-dimensional probability simplex, on which each topic corresponds to a corner, and each point corresponds to a topic distribution. We can see that the model clearly captured the two genres. As a comparison, if we directly

apply LDA, which is equivalent to the 1-genre ($T = 1$) DG model, the learned distribution of topic distributions are shown in Figure 10(b). We see it is apparently not faithful to the behavior of data.



(a) (b)

Figure 10: The genres learned by DG. The genres are actually a mixture of Dirichlet distributions, and here we show its log-density. (a) Result from a 2-genre ($T = 2$) model. (b) Result from a 1-genre ($T = 1$) model.

Finally, we demonstrate the effect of the group size. We re-use the settings in the previous experiment, except that now the group sizes are sampled from a exponential distribution $Exponential(100)$. Figure 11 shows the result of the DG based detectors. Since now there are many small groups, the genres are not as well-defined as before. We can see that the DG results are still acceptable. Through multiple runs we further observe that the performance of DG using the genre score is more stable than the perplexity score. Indeed, simply normalizing the scores by group sizes is not the best strategy, and lacks sound probabilistic interpretation.

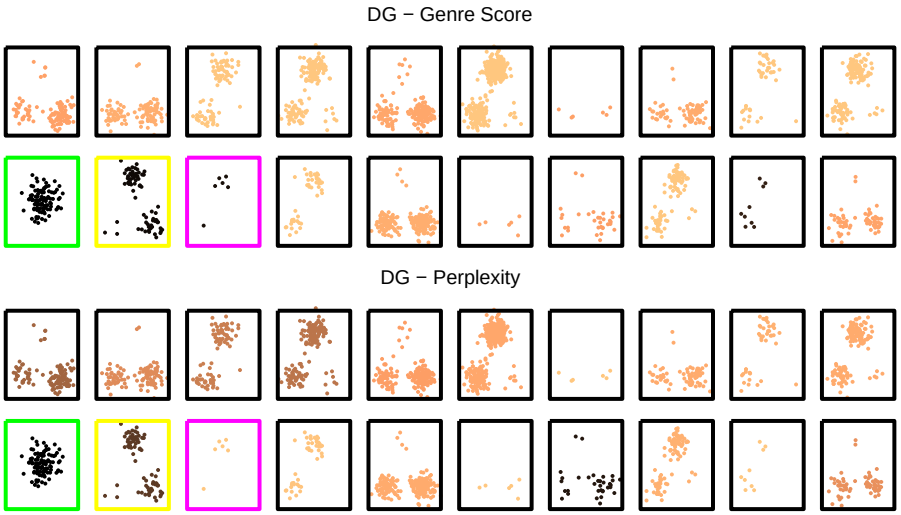


Figure 11: Detection results on a toy data set where the group sizes follow a exponential distribution.

5.4 Group Anomaly Detection in Astronomical Data

We also use the algorithms on the *Sloan Digital Sky Survey* data set to find group anomalies. Again, due to the lack of labeling information on this huge data set, we have to use artificial injections to evaluate the performances of algorithms.

To find the spatial clusters, we first construct a graph by adding edges between nearby galaxies, and then treat the connected components in the graph as spatial clusters. After this preprocessing, 518 spatial groups (7712 galaxies) with sizes between $[10, 50]$ were found. Then we compressed the 500-dimensional feature $\mathcal{C}_{s1}$ (normalized continuum) by PCA into a 2-dimensional space, preserving 95% of the total variance.

We injected artificial group anomalies to test the algorithms. These injections are constructed from random points, such that they were required to lie in a low density region of the 'topic' distribution. Concretely, we first quantize the galaxies into three types/topics. Then a distribution of these topics is computed for each group and plotted on the probability simplex. Then 5 random points, which represent 5 topic distributions, are selected from the low-density region on this simplex. These are the anomalous topic distributions. Finally, 10 injection groups (corresponds to about 2.5% of the normal data) are formed by selecting random galaxies from the same data set according to the chosen anomalous topic distributions.

We compared the DG and GMM models together with a histogram based methods in this experiment. The histogram based methods (H) is repeating the process of the injection: we first quantize the galaxies into several topics, compute the topic distributions for each group, and then on the simplex we find points that are in low-density regions. This is a typical realization of the transformation based methods mentioned in section 4.1. Note that this H detector should be good since it matches our injection process, except that the number of topics been used is different.

The algorithms were compared by the retrieval performance on the injected anomalies. The *average precision* (AP) was calculated using the anomaly scores produced by the models. For DG, we use the genre scores. For GMM, the perplexity score is used. Parameters $T = 4, K = 5$ were used for all methods. The results from 30 random trials are shown in figure 12. Note that the performance values have large variances because each time the injections are very random, and we only added 2.5% anomaly groups.

We see that GMM can hardly do better than a detector that randomly pick out anomalies since every point in the injected groups is random and normal. On the other hand, both H and DG is able to pick out these injections. Further, we see that DG achieves better performance than H. The *paired t-test* on the results of H and DG shows a p-value of 0.0091. This demonstrates the detection power of the DG model given that the H method matches the injection process. We believe the reason is that we are essentially learning the best way to "transform" the groups in the integral process of learning the generative mechanism, which is better than doing the transformation as a separate step.

6. Conclusion and Future Work

In this project, we investigated the problem of anomaly detection on astronomical data sets. Our goal is to design algorithms to assist the astronomers to handle the complex and large scale survey data.

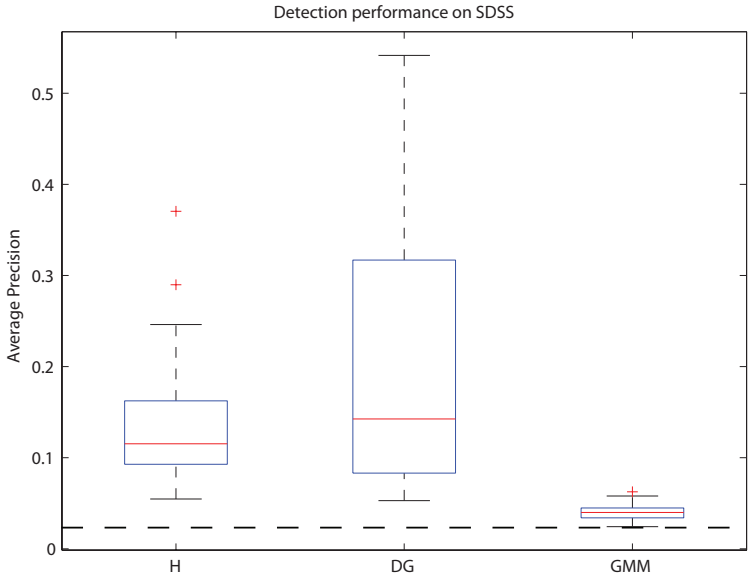


Figure 12: Group anomaly detection performance on the SDSS galaxy data. DG achieves similar performance as H, which repeats the injection process. The dashed line shows the baseline of a random detector.

We propose two algorithms to handle different types of anomalies. The first algorithm, called *mixed-error matrix factorization* (MEMF), is a simple framework for robust low-rank matrix factorization. Given noisy data sets, it is able to identify the outliers and find a reliable principal subspace, which is further used for subspace-based anomaly detection. And its flexibility allows researchers to use it to 'robustify' various factorization methods. More discussion about MEMF is in section 3.4.

The second algorithm we proposed is used to detect group anomalies, even if all of their member points are normal. In this model we adopt the hierarchical generative modeling method, and propose a two-level model based on the concepts of *topic* and *genre*. It is an extension of the *Latent Dirichlet Allocation* (LDA) model so that more complex distribution of the topic distributions can be captured and used for anomaly detection. An efficient learning procedure based on *variational EM* is implemented. More discussion about MEMF is in section 4.4.

There are many possibilities to explore on astronomical data. Unsupervised anomaly detection is just the first step to facilitate gathering human knowledge. For this purpose, we build a web site at `http://www.autonlab.org/sdss` to present the results to and gather feedback from the astronomers. When the initial knowledge is gained, we can employ supervised methods such as *active learning* to further build our knowledge base. Eventually, our hope is to use machine learning to help tame the vast-scale astronomical data, so that we can know the universe better.

References

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, (19-6):716–723, 1974.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

P. Bloomfield and W. L. Steiger. *Least Absolute Deviations: Theory, Applications, and Algorithms (Progress in Probability)*. Birkh'auser Boston, Mass, USA, 1983.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and J'org Sander. Lof: Identifying density-based local outliers. In *ACM SIGMOD Record*, 2000.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Information Theory*, 56(5):2053–2080, 2009.

Philip K. Chan and Matthew V. Mahoney. Modeling multiple time series for anomaly detection. In *IEEE International Conference on Data Mining*, 2005.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–72, 2009.

Kaustav Das, Jeff Schneider, and Daniel Neill. Anomaly pattern detection in categorical datasets. In *Knowledge Discovery and Data Mining (KDD)*, 2008.

Kaustav Das, Jeff Schneider, and Daniel Neill. Detecting anomalous groups in categorical datasets. Technical Report 09-104, CMU-ML, 2009.

B. de Finetti. Funzione caratteristica di un fenomeno aleatorio. *Atti della R. Academia Nazionale dei Lincei, Serie 6. Memorie, Classe di Scienze Fisiche, Mathematice e Naturale*, 4:251–299, 1931.

Chris Ding, Tao Li, and Michael I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE T-PAMI*, 32(1):45–55, 2010.

Maryam Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209 – 230, 1973.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.

F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York., 1986.

Douglas M Hawkins, Li Liu, and S. Stanley Young. Robust singular value decomposition. Technical report, National Institute of Statistical Sciences, 2001.

Geoffrey G. Hazel. Multivariate gaussian MRF for multispectral scene segmentation and anomaly detection. *IEEE Trans. Geoscience and Remote Sensing*, 38-3:1199 – 1211, 2000.

Huber and Peter J. Robust estimation of a location parameter. *Annals of Statistics*, 53: 73–101, 1964.

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, INRIA, 2009.

Michael I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.

Qifa Ke and Takeo Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Eamonn Keogh, Jessica Lin, and Ada Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *IEEE International Conference on Data Mining*, 2005.

M. Kuss, T. Pfingsten, L. Csato, and Rasmussen. Approximate inference for robust gaussian process regression. Technical report, Max Planck Institute: Biological Cybernetics, Tbingen, Germany., 2005.

Fernando De la Torre and Michael J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54:117–142, 2003.

R. M. Larsen. Propack - software for large and sparse svd calculations. URL `http://soi.stanford.edu/~rmunk/PROPACK`.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

Le Li and Yu jin Zhang. Fastnmf: Highly efficient monotonic fixed-point nonnegative matrix factorization algorithm with good applicability. *Journal of Electronic Imaging*, 18 (3), 2009.

Liyuan Li, Weimin Huang, Irene Yu-Hua Gu, and Qi Tian. Statistical modeling of complex backgrounds. *IEEE Trans. Image Processing*, 13(11):1459–1472, 2004.

Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML*, 2006.

Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l2,1-norm minimization. In *The Twenty-fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

Shiqian Ma, Donald Goldfarb, and Lifeng Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Math. Program., Ser. A*, to appear.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 2009.

Geoffrey J. Mclachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. John Wiley and Sons, 1996.

Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Uncertainty in Artificial Intelligence (UAI)*, 2002.

Daniel B. Neill and Gregory F. Cooper. A multivariate bayesian scan statistic for early event detection and characterization. *Machine Learning*, 79:261 – 282, 2010.

Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical report, CORE, Université catholique de Louvain, 2007.

Minh Hoai Nguyen and Fernando De la Torre. Robust kernel principal components analysis. In *NIPS*, 2009.

Nam H. Nguyen, Thong T. Do, and Trac D. Tran. A fast and efficient algorithm for low-rank approximation of a matrix. In *STOC*, 2009.

Jason D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

Ruslan Salakhutdinov and Andriy Minh. Probabilistic matrix factorization. In *NIPS*, 2007.

Gideon E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, (6-2): 461–464, 1974.

Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *ICML*, 2003.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

Hansheng Wang and Chenlei Leng. A note on adaptive group lasso. *Computational Statistics and Data Analysis*, 52:5277–5286, 2008.

John Wright, Yigang Peng, Yi Ma, Arvind Ganesh, and Shankar Rao. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In *NIPS*, 2009.

Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *SIGIR*, 2003.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candès, and Yi Ma. Stable principal component pursuit. In *International Symposium on Information Theory*, 2010.