CRIME INCIDENTS EMBEDDING USING RESTRICTED BOLTZMANN MACHINES

Shixiang Zhu, Yao Xie

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

We present a new approach for detecting related crime series, by unsupervised learning of the latent feature embeddings from narratives of crime record via the Gaussian-Bernoulli Restricted Boltzmann Machine (GBRBM). This is a drastically different approach from prior work on crime analysis, which typically considers only time and location and at most category information. After the embedding, related cases are closer to each other in the Euclidean feature space, and the unrelated cases are far apart, which is a good property can enable subsequent analysis such as detection and clustering of related cases. Experiments over several series of related crime incidents hand labeled by the Atlanta Police Department reveal the promise of our embedding methods.

Index Terms— Unsupervised learning, crime data analysis, feature embeddings, neural networks

1. INTRODUCTION

A fundamental and one of the most challenging tasks in crime analysis is to find related *crime series* [1], which are committed by the same individual or group. Such series of crimes follow a so-called modus operandi (M.O.), for instance, some criminals always break into houses in the late afternoon from backdoor to steal jewels. Finding crime series based on M.O. critically depend on extracting informative features for crime incidents [1], which is usually done by the human, however, this is not scalable to larger and ever-growing crime data set. For instance, in the City of Atlanta, from the year 2013 to 2017, there are a total of 1,096,961 cases with over 800 categories.

Crime incident reports (a.k.a. police reports) are a large source of data that contains rich information for detecting related crime series, which somehow has not been tapped. Each incident has a unique police report, which contains the time, location (latitude and longitude), and last but the not the least, the free-text narratives entered by police officers. According to the crime analysts, the free-text narrative contains the most useful information form their investigation, but there has yet been a tool to automatically extract useful features and information from the free-text narratives, since the narratives are very noisy and unstructured, written by different police officers, and are sometimes incomplete English sentences since they are written in a haste. Currently, crime analysts identify crime series by hand.

In this paper, we propose a new approach for detecting related crime series that are usually committed by a same group of suspects. This method can directly process the free-text narratives of the police reports, and map them into a feature vector space that automatically captures the similarity of incidents. The main idea is to map the raw feature extracted from the narratives using standard NLP models (such as the bag-of-words model), into latent feature vector space, using Gaussian-Bernoulli Restricted Boltzmann Machines (GBRBMs). The GBRBM is trained from a large number of data without supervision. After training, GBRBM embeds the crime incidents to capture their similarity by vicinity in the Euclidean space. Our work is inspired by the idea of word embeddings [2], we similarly assume We validate the effectiveness of our method over several series of related crime incidents hand labeled by the Atlanta Police Department.

Relation to prior work. A seminal work [1] uses subspace clustering to find crime series and has achieved good performance. However, [1] requires clean features that are entered by the human for each incident. For the larger scale of police report data, there are not many clean hand-entered features. For such dataset, it is highly desirable to be able to directly work with the free-text narratives and find hidden correlations of the crime series.

A recent work [3] explores the possibilities of using natural language processing tools for crime pattern detection, including Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA). In certain cases (as we observed in our experiment in Section 3), embedding via GBRBM can be a better approach than LDA, because GBRBM can capture more subtle distinctions between different narratives. This is

Thanks to the Atlanta Police Foundation for funding. This project is a collaboration with the Atlanta Police Department. The authors would like to thank Mr. Frank Ruben, Lieutenant David Wilson, Major John Quigley at the Atlanta Police Department for technical support for obtaining data and helpful discussions for problem formulation. Also thank Ms. Debra Lam at the Georgia Tech Institute of People and Technology for support.

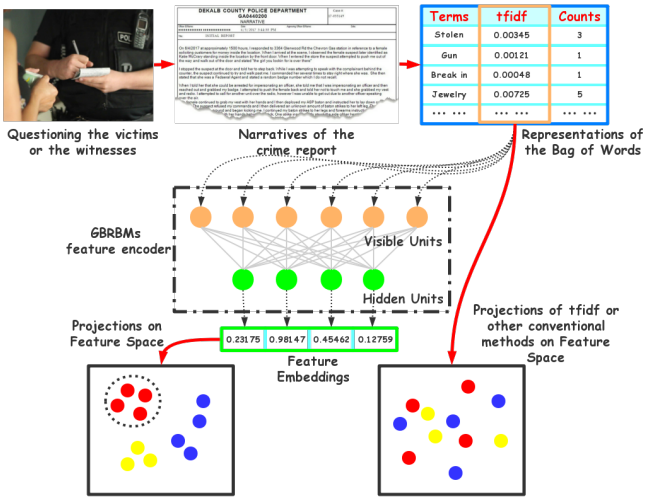


Fig. 1. An overview of the essential idea of our method.

possibly due to that the structure of RBM directly captures the hidden correlation between incidents via nonlinear structures. We also observe that the narratives usually contain professional but limited "police vocabulary" and has distinct but similar writing styles. This may explain why learning good vector representations is possible, as we have observed from experimenting with millions of crime records in the Atlanta Police Department database.

Another line of research from crime analysis and predictive policing focus on the so-called *hotspot prediction* (see, e.g., [4, 5]), which has achieved a lot of success in modeling the dynamics of crimes over space and time. The hotspot prediction aims to model the excitation relationships between crime incidents occurred at different space and time. The idea is that certain types of crimes (such as gang crimes) have a triggering effect: an event happens at certain location and time may trigger future incidents at similar locations in the futures. The hotspot prediction typically only use time, location (and sometimes category) information of the incidents.

2. EMBEDDING USING GBRBM

Our dataset is provided by the Atlanta Police Department (APD), which consist of all crime incidents from 2013 to 2017, with 1,096,961 cases in over 800 different categories. The records are unlabeled and naive clustering will not resolve them into related crime series. The latent feature embedding algorithm that we describe below capture the critical information of the crime and criminal correlations would be very helpful no matter for further classification or clustering of the crime cases in the absence of label information.

The flowchart of the embedding algorithm is shown in Fig. 1. On a high level, we first generate raw features using standard natural language processing tools, such as term-frequency inverse-document-frequency (TF-IDF) for each incident. The core of the algorithm is the GBRBM with the input being the TF-IDF of reports. We train the GBRBM by maximizing the likelihood function (defined in terms of the energy function) for the occurrence of word terms in a police report, and in the end, use the output latent variables as the embedded features for each corresponding crime incident. The embedded features have a nice property that related cases have features in the vicinity in the Euclidean space.

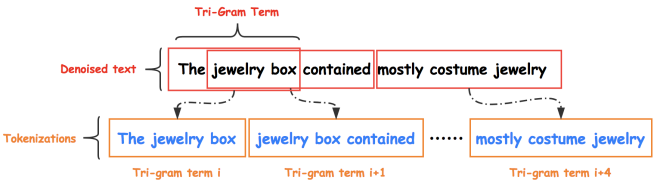


Fig. 2. An example of Tri-Gram.

2.1. Raw Feature Extraction

The free-text narratives are highly unstructured data, meanwhile they also consist typos, irrelevant words or phrases. Our raw feature extraction is designed to be robust to these issues. Several key steps are as follows: **Data cleaning**: we normalize the text to lower-cases so that the distinction between "The" and "the" are ignored; also remove stop-words, independent punctuation, low-frequency terms (low TF) and the terms that appeared in most of the documents (high IDF). **Tokenization**: for each narrative of the crime, the text needs to be tokenized into a list of word-level tri-gram terms, as shown in Fig.2. As a matter of experience, unigram and bigram loss too much context information, while the four-gram or higher gram secures only tiny gains while making the feature vector become too long and increasing the model complexity and training is more time-consuming. **Bag of Words (BoW)**: BoW is a simplifying, orderless document representation commonly used in NLP. In this representation, each document is represented by one vector where each element means the occurrence in association with a specific term. As a result, the entire corpus can be converted to a term-document matrix and a dictionary that keeps the mapping between the terms and their ids. **Term Frequency-Inverse Document Frequency (TF-IDF)** is a conventional method for extracting feature vectors from the term-document matrix to de-emphasize frequent words. In [6], TF-IDF weighting scheme has been used to reduce the impact of the terms that appeared in most of the documents, which means that they have weak discrimination capability across documents.

2.2. Model Architectures

GBRBM is a type of neural networks and it can be viewed as a probabilistic graphical model. GBRBM is a powerful model for the complex joint distribution of real-valued visible variables and binary valued hidden variables [7]. Here,

we consider a standard GBRBM, whose architecture is shown in Fig.1, where there are n hidden variables as embeddings and m visible variables that will be input by tokenized word terms. The weights, represented as a matrix $W = (w_{ij})$, associate the hidden variable h_j and visible variable v_i. There are also bias weights c_j for the hidden variable and b_i for the visible variable. The learned weights and biases define a Gibbs probability distribution over all possible input data via the energy function, denoted as $E(\mathbf{v}, \mathbf{h})$. The energy function for the joint configuration $(\mathbf{v}, \mathbf{h})$ of the visible and hidden units is defined [8]

$$E(v, h) = -\sum_{i,j} w_{ij} h_j \frac{v_i}{\sigma_j} - \sum_i \frac{(v_i - b_i)^2}{2\sigma^2} - \sum_j h_j c_j.$$

Note that the first term of the energy function captures the joint pattern of the hidden and visible variables, and the second and the third terms capture the linear effect of both the hidden and the visible variables. A nice structure of the RBM is that the joint distributions of the visible and hidden variables, conditioned on each other, are mutually independent

$$p(v|h) = \prod_{i=1}^{m} p(v_i|h), \quad p(h|v) = \prod_{j=1}^{n} p(h_j|v).$$

Moreover, the conditional distribution of $v_i|h$ is a normal random variable $\mathcal{N}(b_i + \sigma_i \sum_{j=1}^{n} w_{ij} h_j, \sigma_i^2)$

$$p(v_i = v|h) = \frac{1}{\sigma_i \sqrt{2\pi}} \cdot e^{-\frac{1}{2\sigma_i^2}(v_i - b_i - \sigma_i \sum_{j=1}^{n} w_{ij} h_j)^2},$$

and the conditional distribution of $h_i|v$ is Bernoulli variable

$$p(h_j = 1|v) = \sigma(\sum_{i=1}^{m} w_{ij} \frac{v_i}{\sigma_i} + c_j).$$

where σ is a sigmoid function. Due to this property, it is easily to sample visible or hidden variables via Gibbs sampling [9] in just two steps: sampling a new state h for the hidden units based on $p(h|v)$ and sampling a state v for the visible layer based on $p(v|h)$. The sampling procedure is essential to perform model estimation.

2.3. Model Estimation

We follow the standard training approach for GBRBM. The training objective of the GBRBM is to maximize a likelihood function, which is defined via the energy function. The training result, somehow, in the end, converges to representations such that related cases tend to be close to each other in Euclidean space. More formally, given a set of training narratives $\mathbf{V} = \{v^{(1)}, v^{(2)}, v^{(3)}, \ldots, v^{(N)}\}$, the objective of the model is to maximize the average log likelihood given by $\log \mathcal{L}(\theta|V) = \sum_{i=1}^{N} \log p(v^{(i)}|\theta)$, where the marginal distribution is given by

$$p(v) = \sum_h p(v, h) = \frac{\sum_h e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}.$$

Note that the number of possible values of h vectors is exponential in the number of hidden variables, so in practice, one usually performs sampling approach to calculate the sum approximately.

Directly obtaining unbiased estimates of the log-likelihood gradient using MCMC methods typically requires many sampling steps. In training, we adopt the k-step contrastive divergence (CD-k) approach, which is an approach to approximate the gradient in training GRB via gradient descent [10]. The main idea is to approximate the gradient of the log-likelihood with respect to θ for one training pattern $v^{(0)}$ as

$$\mathrm{CD}_k(\theta, v^{(0)}) = -\sum_h p(h|v^{(0)}) \frac{\partial E(v^{(0)}, h)}{\partial \theta} + \sum_h p(h|v^{(k)}) \frac{\partial E(v^{(k)}, h)}{\partial \theta}.$$

The Gibbs chain is initialized with a training example $v^{(0)}$ of the training set and yields the sample $v^{(k)}$ after k steps. Each step t consists of sampling $h^{(t)}$ from $p(h|v^{(t)})$ and sampling $v^{(t+1)}$ from $p(v|h^{(t)})$ subsequently. The iterations are repeated until certain empirical convergence has achieved.

3. RESULTS

To test the performance of our embedding method, we devise a comprehensive test dataset. The dataset contains five hand-labeled crime series that were identified as committed by five individual arrestees, and 441 randomly selected irrelevant crime cases. Details of the test data are given in Table 1.

Ideally, we hope that the crime records which were committed by the same arrestee tend to be closed to each other in the embedded feature space. However, it is not so easy to show the distance between two crime cases directly without dimensionality reduction on their feature embeddings. For visualization purposes, we apply two-dimensional t-distributed stochastic neighbor embeddings (t-SNE) [11] to the feature embeddings, to convert the high-dimensional feature vectors into a matrix of pairwise similarities. t-SNE is capable of capturing local structure of the high-dimensional data, while also revealing global structures such as the presence of clusters at several scales [11].

In our experiments, the basic parameters for the GBRBM are as follows: the size of the visible layer is fixed to 9863, which is determined by the size of the dictionary. We tried 3 different sizes of the hidden layer for testing the model performance, with 1000, 2000 and 5000 hidden units respectively. In the training stage, the learning rate is 0.05, the batch size is 20, and the number of epochs at fine tune periods is 30. We adopt the Stochastic Gradient Descent (SGD) optimizer [12] to optimize the loss function.

First, we study the effect when increasing the number of hidden variables in GBRBM. This will lead to different dimensions of the feature embeddings. Fig.3 (a) shows that the

Table 1. Details of the test data

Id	Number	Category
Crime Series 1	8	*Robbery at Residence*
Crime Series 2	7	*Robbery at Gas Station*
Crime Series 3	4	*Pedestrian Robbery*
Crime Series 4	15	*Attempt Auto Theft*
Crime Series 5	22	*Burglary*
Random Cases	441	*Over 89 Categories*
Total	497	

Table 2. The comparison of the training time.

Methods	Training Time
GBRBMs with 1000 units	$\sim$ *2 mins*
GBRBMs with 2000 units	$\sim$ *3 mins*
GBRBMs with 5000 units	$\sim$ *7 mins*
LDA with 1000 topics	$\sim$ *5 mins*

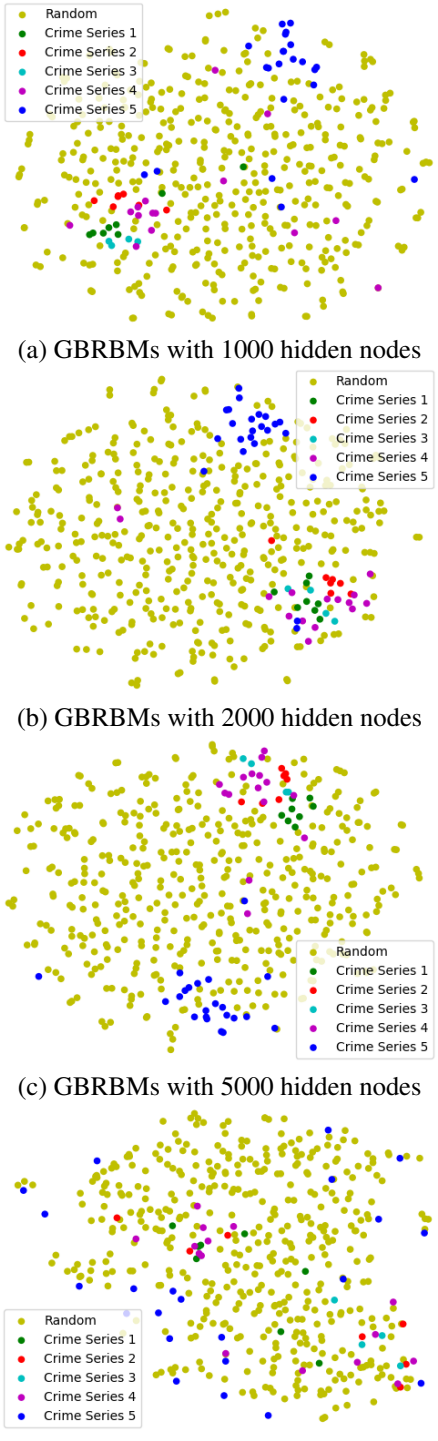
embeddings with 1000 units can successfully map crime series 1, 2, 3, 5 to clusters, which separate them out from random cases. In particular, for crime series 1, 2, 3, the embeddings of same crime series gather closely at some local regions. The clustering does not work quite well for crime series 4. When we increase the number of the hidden units to 2000, the performance for crime series 4 become much better as shown in Fig.3 (b) and (c). The performance of the GBRBM does not seem to have significant further improvement when we further increase the number of the hidden variables (Fig.3 (c)).

Second, we compare the performance of GBRBM with Latent Dirichlet Allocation (LDA) [13] on the same test data set. Fig.3 shows four instances, which are the projection of the embeddings via GBRBM and LDA topic modeling on a 2D t-SNE space. We implement a LDA with 1000 latent topics. It turns out that the LDA does not map crime series into clusters: they are scattered randomly in the feature space without any obvious patterns.

The embedding can be computed efficiently. As reported in Table 2, the training time of reaching the convergence precision for 497 cases are around minutes. This also shows the GBRBM with less than 2000 hidden units have an advantage over the LDA in terms of the training time.

4. CONCLUSION

We have presented a novel approach for detecting crime series that are related, using embedding found by the Gaussian-Bernoulli Restricted Boltzmann Machine (GBRBM). The GBRBM tends to map related cases (that share certain correlation in the raw feature) into features that are in the vicinity in the Euclidean space. Our methods demonstrate very promising results on real police data and demonstrated that the feature embeddings can have advantages over the conventional text processing methods on detecting crime series in certain cases. Ongoing work is to develop an online crime series detection algorithm based on the embedded features.



(a) GBRBMs with 1000 hidden nodes

(b) GBRBMs with 2000 hidden nodes

(c) GBRBMs with 5000 hidden nodes

(d) LDA with 1000 topics

Fig. 3. Visualization of the projections of different embeddings on the 2D t-SNE space.

5. REFERENCES

[1] T. Wang, C. Rudin, D. Wagner, and R. Sevieri, "Finding Patterns with a Rotten Core: Data Mining for Crime Series with Cores," *Big Data*, vol. 3, no. 1, pp. 3–21, 2015.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781*, 2013.

[3] D. Kuang, P. J. Brantingham, and A. L. Bertozzi, "Crime Topic Modeling (Classification)," *arXiv:1701.01505*, p. 47, 2017.

[4] M. B. Short, P. J. Brantingham, A. L. Bertozzi, and G. E. Tita, "Dissipation and displacement of hotspots in reaction-diffusion models of crime," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 107, pp. 3961–3965, mar 2010.

[5] J. T. Woodworth, G. O. Mohler, A. L. Bertozzi, and P. J. Brantingham, "Non-local crime density estimation incorporating housing information," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2028, pp. 20130403–20130403, 2014.

[6] Stanford NLP Group, "Tf-idf weighting." https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html.

[7] A. Fischer and C. Igel, "An Introduction to Restricted Boltzmann Machines," *Lecture Notes in Computer Science: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, vol. 7441, pp. 14–36, 2012.

[8] G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *ICML*, no. 3, 2010.

[9] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 1064–1071, 2008.

[10] G. E. Hinton, "Training products of experts by minimizing constractive divergence," *Neural Computation*, 2002.

[11] L. V. D. Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[12] J. Duchi, J. B. Edu, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization *," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[13] D. M. Blei, B. B. Edu, A. Y. Ng, A. S. Edu, M. I. Jordan, and J. B. Edu, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.