Partially Generative Neural Networks for Gang Crime Classification with Partial Information

Sungyong Seo[1], Hau Chan[2], P. Jeffrey Brantingham[3], Jorja Leap[3], Phebe Vayanos[1], Milind Tambe[1], Yan Liu[1]
[1]{sungyons,phebe.vayanos,tambe,yanliu.cs}@usc.edu, University of Southern California, CA 90007, USA
[2]{hchan3}@unl.edu, University of Nebraska–Lincoln, NE 68588, USA
[3]{branting,jleap}@ucla.edu, University of California, Los Angeles, CA 90095, USA

Abstract

More than 1 million homicides, robberies, and aggravated assaults occur in the United States each year. These crimes are often further classified into different types based on the circumstances surrounding the crime (e.g., domestic violence, gang-related). Despite recent technological advances in AI and machine learning, these additional classification tasks are still done manually by specially trained police officers. In this paper, we provide the first attempt to develop a more automatic system for classifying crimes. In particular, we study the question of classifying whether a given violent crime is gang-related. We introduce a novel Partially Generative Neural Networks (PGNN) that is able to accurately classify gang-related crimes both when full information is available and when there is only partial information. Our PGNN is the first generative-classification model that enables to work when some features of the test examples are missing. Using a crime event dataset from Los Angeles covering 2014-2016, we experimentally show that our PGNN outperforms all other typically used classifiers for the problem of classifying gang-related violent crimes.

1 Introduction

There are more than 1 million violent crimes reported in the United States each year (Federal Bureau of Investigation 2015). In 2015, for example, there were more 15,696 reported homicides, 764,449 reported aggravated assaults and 327,374 reported robberies (Federal Bureau of Investigation 2015). It is no surprise, therefore, that law enforcement agencies are interested in reducing crimes by leveraging crime prediction models (Mohler et al. 2011; Green, Horel, and Papachristos 2017) and intervention strategies optimized for deterrence (Loughran et al. 2011; Mohler et al. 2015). Of particular concern is violent crime associated with gangs. In many large cities, a substantial fraction of the violent crime can be attributed to the activities of gangs or gang members. In both Los Angeles and Chicago, for example, time-intensive investigative work shows that more than 50% of all homicides appear tied to gangs. Development of real-time predictive models to deal with the gang violence problem hangs on the ability to identify gang crimes quickly and accurately. This is a challenging problem in part because of the fragmentary and heterogeneous nature of the data collected about crime event. While investigative effort is often able to close the gap, this typically takes time that could be otherwise directed towards intervention. Here we develop a neural network model that is able to successfully distinguish gang-related from non-gang crimes given partial data.

Current process of classifying gang-related crimes. The classification process for identifying gang crimes is rooted in data collected at the scene of the crime, contextual details that arise during investigation and information about recent gang activity. The law enforcement approach is to classify a crime as gang-related if there is any evidence that the suspect/victim is a gang member, or that the crime is consistent with gang activity. An individual might self-identify as a gang member (Decker et al. 2014), or might bear gang tattoos that signal gang membership (Klein and Maxson 2006). Evidence that a crime is consistent with gang activity may include the type of crime (e.g., gun homicide or assault) (Bjerregaard and Lizotte 1995), the location of the crime within or near the boundary of a gang territory (Brantingham et al. 2012), the appearance that the crime is a retaliation for another previous crime (Jacobs and Wright 2006; Short et al. 2014), or a connection between the crime and gang social networks (Green, Horel, and Papachristos 2017; Papachristos 2009). It is important to recognize the limitations of a binary classification of crimes as gang-related or not. Criminologists recognize a difference between crimes that are gang-related, which originate in activities in support the gang, and crimes that are merely gang-affiliated, meaning that the crime is not connected to gang activities other than having a victim or suspect tied to gangs (Rosenfeld, Bray, and Egley 1999). Moreover, it is clear now that individuals might be socially embedded to different degrees in a gang (Decker et al. 2014), which presumably translates into events between differentially tied to gang activity.

Our goal. In present law enforcement contexts, the classification of gang-related crimes is done manually by some trained police officers in the criminal gang division. The process of classification is both labor and time intensive. Evidence immediately available upon reporting of a crime may provide strong indications that it is tied to gangs. Data collected over the process of investigation may reverse this determination or reinforce it. This information gathering process may take considerable time (e.g., interviewing vic-

tim(s)/suspect(s) and investigating the crime scenes). Other demands on time may pull officers away from investigative activities, the result being different degrees of information about any individual crime. Moreover, since each crime is unique in some sense, it is also common for the final information to display different degrees of completeness beyond any effects of the investigative activity. Overall, crimes with incomplete or partial information are quite common. Motivated by these facts, our goal is to build classifiers to automatically classify gang-related crimes where some crucial pieces of crime information are not currently available or are missing. In particular, we:

- Study the classification problems of determining whether a crime with partial information is gang-related
- Introduce a novel Partially Generative Neural Network (PGNN) for general supervised learning problems where some features of test/new examples are missing
- Show experimentally that our PGNN is effective in classifying gang-related crimes with full and partial information and outperforms other baseline classifiers.

To the best of our knowledge, we are the first to consider gang-crime classification problem in the AI, machine learning, and criminologist domains. More broadly, our PGNN is the first generative-classification model of its kind for general supervised learning tasks when some features of the test examples are missing, which is incredibly common in real-life settings. Our PGNN works by generating the missing feature values and makes a prediction using the generated feature values and the available features. Our PGNN can also be used to make a prediction when there are no missing features. As we will show in Section 4, PGNN outperforms other classifiers in the standard supervised learning setting and our setting with missing information.

1.1 The General Problem Statement

In our work, we assume that there are two types of features related to a given task, 1) base features F_b and 2) additional features F_a. As the names denote, F_b has basic features regarded as easily-collectible features and expected that this set of features is always available. On the other hand, it is expensive to constantly observe F_a which has additional information. G is a set of targets such as labels for classification. Although our model does not assume the types of tasks, i.e., G can be any forms, we assume that the task we are interested in is a classification and G is a class label in this work. In the training phase, for each training example l, we are giving the base features $F_b^{(l)}$, the additional features $F_a^{(l)}$, and the label $G^{(l)}$ of l. In the test/prediction phrase, we are given a set of T examples such that, for each $t \in T$, only $F_b^{(t)}$ is available. Our goal is to learn a classifier that would predict the (true) label of each $t \in T$.

In the context of gang-related crime prediction, l is a previous crime record with information F_b, F_a, and G (i.e., gang-related or not). For each new crime $t \in T$, F_b are available but F_a (such as the narrative of the crime as in Table 1) are missing in the current police data collection process. As

```
DO-S1 AND S2 BECAME INVOLVED IN AN ARGUMENT WITH
MUTIPLE V S2 SWUNG AT V,WITH A BAT SS ARRESTED
AND ARE *** GANG MEMBERS
```

Table 1: An example of crime narrative

we mentioned, F_a such as narrative information may take considerable time to gather from the investigation.

1.2 Related Work

Gang-related works. Relatively little attention has yet been directed to using neural networks or machine learning to classify or predict crime. Some notable exceptions include point process approaches that use the time and location of known crimes to predict where and when future crimes will occur (Mohler et al. 2015). In (Stomakhin, Short, and Bertozzi 2011) they use a related point process model to identify which gangs may be responsible for a given crime over a network of rivals. Similarly, (Green, Horel, and Papachristos 2017) use a point process on a social network of an individual to map the contagion of gun violence. Moving beyond crime data, (Wang, Gerber, and Brown 2012) uses latent Dirichlet allocation (LDA) to extract topics from Twitter posts and then develops a general linear model to predict hit-and-run traffic incidents from the topic models. Similarly, (Gerber 2014) also uses LDA to extract topics from Twitter but then fuses those topics with kernel density estimation estimates of crime density to show that the Twitter features improve predictability of crime. Most related to the present work is (Kuang, Brantingham, and Bertozzi 2017), which uses NMF-based topic modeling to investigate the relationships between the full array formal crime type classifications used by police and narrative texts associated with crime events. We leverage similar features below. However, (Kuang, Brantingham, and Bertozzi 2017) do not build a classifier based on topic structures.

Missing values. In the past decades, many approaches (Kreindler and Lumsden 2012) have been investigated to handle missing values in different domains. For instance, wavelet variance analysis (Mondal and Percival 2010), correlation analysis (Rehfeld et al. 2011), sequential regression (Raghunathan et al. 2001), and multiple imputation (White, Royston, and Wood 2011) have been proposed to impute missing values. Some of these works are based on the temporal dependencies and others require domain experts to build the technique, and therefore, it is not easy to directly use these approaches to our domain.

Recently, data-driven models based on deep learning have been introduced to handle the missing gaps. (Che et al. 2016) exploit two representations of informative missing patterns by masking and time interval with Gated Recurrent Unit (GRU). However, this work does not address how to impute the random missing gaps but how to utilize the gaps. Most related to our model is (Hoffman, Gupta, and Darrell 2016) that hallucinate expensive depth images from RGB images. They propose discriminative networks to mimic the depth image and illustrate how to define joint loss functions to facilitate the transferring. Our approach is different, we focus on the relationship between categorical features and natural language text summarizing crimes. This relationship is not very intuitive and we introduce the generative mod-

ule (Kingma and Welling 2013) to transfer the categorical features to the latent representation of the text.

2 Partially Generative Neural Network

In this section, we propose a novel Partially Generative Neural Network (PGNN) architecture that enables us to learn mapping functions not only between an input vector (F_b and F_a) and an output label (G) but also between a subset of the input vector (F_b) and another subset of the input vector (F_a). The former mapping function can be modeled by a simple classifier such as a logistic regression or a multilayer neural network. On the other hand, the latter mapping function is based on the generative variational model to mimic a set of (relatively expensive) additional features (F_a) which might not be available at test time from a set of easily available features. Once all parameters in the proposed model are trained, it is flexible to use the model no matter whether the additional features are available.

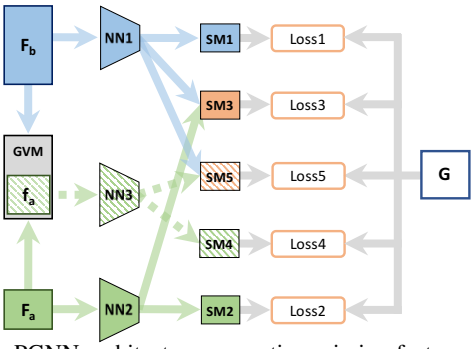


Figure 1: PGNN architecture generating missing features at training time. NN is neural networks and SM is for softmax activations.

2.1 Architecture

Modules for full features. Figure 1 describes the overall architecture internally generating F_a from F_b. First, there are two input sets, F_b and F_a, and two neural networks denoted as NN1 and NN2. These neural networks consist of fully connected layers and they reduce the dimension of features. Three loss functions denoted as Loss1($\mathcal{L}_1$), Loss2($\mathcal{L}_2$), and Loss3($\mathcal{L}_3$) correspond to the neural networks (NN1 and NN2) and their forms are defined by a given task. NN1 and NN2 read F_b and F_a as their inputs, respectively, and they do not require opposite features for the training. Each output from NN1 and NN2 is used for the classification separately. For example, $\mathcal{L}_1$ is computed by the output of NN1 and the true label G. Similarly, $\mathcal{L}_2$ only requires the output of NN2. Thus, $\mathcal{L}_1$ and $\mathcal{L}_2$ can be considered as loss functions based on F_b and F_a, independently. On the other hand, $\mathcal{L}_3$ requires both F_b and F_a features. The outputs of NN1 and NN2 are concatenated and used to compute $\mathcal{L}_3$. Since $\mathcal{L}_3$ needs entire features and two neural networks, it is expected to show the least loss value. Overall, these three loss functions drive parameters in NN1 and NN2 to minimize the objective, and particularly $\mathcal{L}_3$ provides the classification error when entire features are available.

Modules for missing features. Let f_a be the generated feature from GVM and this is going to replace the role of F_a when it is not available at the test time. Figure 1 illustrates how Generative Variational Module (GVM) is used to generate f_a using NN3 and f_a is propagated in PGNN (dashed arrows). We will discuss GVM below. NN3 is a fully connected layer which reads f_a and encodes it to a low dimensional vector. The encoding vector is used in two loss functions, (Loss4($\mathcal{L}_4$) and Loss5($\mathcal{L}_5$)). First, $\mathcal{L}_4$ is similar with $\mathcal{L}_2$. $\mathcal{L}_4$ shows the classification error when f_a is the only available feature. $\mathcal{L}_5$ is particularly important since it provides the error when F_a is unavailable at test time. As like $\mathcal{L}_3$, it reads the concatenated vector from outputs of NN1 and NN3, and turns out discrepancy from the true label.

2.2 Generative Variational Module (GVM)

To generate missing features F_a, we modify the generative process in variational autoencoder (VAE) (Kingma and Welling 2013) which enables latent representations and is trainable with other neural networks jointly. Specifically, we derive the appropriate loss function to generate desired features in the context of supervised learning.

Latent variable model. Latent variable models describe a stochastic process which governs the generative process of an observation from the latent space. Let $\mathbf{x}$ and $\mathbf{y}$ be "observable" variables and $\mathbf{z}$ denote "latent" variables, respectively. Then, Bayes' theorem tells us how to infer $\mathbf{z}$ from $\mathbf{x}$ and $\mathbf{y}$. In other words, the latent variable from its distribution, $\mathbf{z} \sim P(\mathbf{z})$, gives a conditional distribution of an observation, $\mathbf{y} \sim P(\mathbf{y}|\mathbf{x}, \mathbf{z})$, and the set of parameters governing the distribution is decided by our model assumption (e.g., neural networks) and datasets. Since the observations $\mathbf{x}$ and $\mathbf{y}$ are given, we are interested in building the latent model to infer $\mathbf{z}$ from observations $\mathbf{x}$, hence the posterior inference, $P(\mathbf{z}|\mathbf{x})$, and to reconstruct $\mathbf{y}$ from the $\mathbf{x}, \mathbf{z}$, the likelihood. We follow the variational inference (Wainwright, Jordan, and others 2008) to infer the posterior. In our model, $\mathbf{x}$ and $\mathbf{y}$ correspond to F_b and F_a, respectively.

Variational lower bound. The idea behind variational inference is to (1) propose a parametric distribution $Q_\theta(\mathbf{z}|\mathbf{x})$ with the parameter θ which we know, and to (2) adjust the parameter θ so that $Q_\theta(\mathbf{z}|\mathbf{x})$ is as close to $P(\mathbf{z}|\mathbf{x})$ as possible.

To measure the closest/distance of the two distributions, we use the typical Kullback-Leibler (KL) divergence, which is defined below. Let $\mathbf{x}$ and $\mathbf{y}$ have relations which are able to be represented as a function $\mathcal{F}$ that $\mathcal{F}(\mathbf{x}) \approx \mathbf{y}$. Under this assumption, the KL divergence between two distributions $Q(\mathbf{z}|\mathbf{x})$ and $P(\mathbf{z}|\mathbf{x}, \mathbf{y})$ is re-written as:

$$\begin{aligned} \mathrm{KL}(Q||P) &= \sum_{\mathbf{z}} q_\theta(\mathbf{z}|\mathbf{x}) \log \frac{q_\theta(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x}, \mathbf{y})} \\ &= \left(\sum_{\mathbf{z}} q_\theta(\mathbf{z}|\mathbf{x}) \log \frac{q_\theta(\mathbf{z}|\mathbf{x})}{p(\mathbf{z},\mathbf{y}|\mathbf{x})} \right) + \log p(\mathbf{y}|\mathbf{x}) \end{aligned} \quad (1)$$

where p and q denote the densities of P and Q. To minimize $\mathrm{KL}(Q||P)$ with respect to the parameters θ, we just have to minimize the first term of above equation since $p(\mathbf{y}|\mathbf{x})$ is

fixed with respect to θ.

$$\sum_{\mathbf{z}} q_\theta(\mathbf{z}|\mathbf{x}) \log \frac{q_\theta(\mathbf{z}|\mathbf{x})}{p(\mathbf{z},\mathbf{y}|\mathbf{x})} = \mathbb{E}_Q \left[\log \frac{q_\theta(\mathbf{z}|\mathbf{x})}{p(\mathbf{z},\mathbf{y}|\mathbf{x})} \right]$$
$$= \mathbb{E}_Q[\log q_\theta(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{y}|\mathbf{z},\mathbf{x}) - \log p(\mathbf{z}|\mathbf{x})] = -\mathcal{L} \quad (2)$$

We call $\mathcal{L}$ as the variational lower bound (we will see soon why) and $\mathcal{L}$ is required to be maximized to minimize Eq. 1. Then, $\mathcal{L}$ can be further rearranged as:

$$\mathcal{L} = \mathbb{E}_Q[\log p(\mathbf{y}|\mathbf{z},\mathbf{x})] - \mathrm{KL}(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z}|\mathbf{x})) \quad (3)$$

Eq. 3 provides the loss function of GVM. First, $\log p(\mathbf{y}|\mathbf{z},\mathbf{x})$ is involved with the reconstruction error in supervised learning approach on $(\mathbf{x},\mathbf{y})$ pairs. The second term, $\mathrm{KL}(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z}|\mathbf{x}))$, is exactly same as Eq.(1) in (Kingma and Welling 2013), and therefore, it can be minimized by maximizing $\mathbb{E}_Q[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z}))$. Since the likelihood $\mathbb{E}_Q[\log p(\mathbf{x}|\mathbf{z})]$ is not considered in GVM (reconstruction error is in $\log p(\mathbf{y}|\mathbf{z},\mathbf{x})$ instead), the optimization is realized by minimizing $\mathrm{KL}(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z}))$.

Finally, the probability distribution of $\log p(\mathbf{y}|\mathbf{x})$ can be written as: $\log p(\mathbf{y}|\mathbf{x}) = \mathrm{KL}(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z}|\mathbf{x},\mathbf{y})) + \mathcal{L}$. Since the KL divergence is always positive, $\log p(\mathbf{y}|\mathbf{x})$ must be larger than $\mathcal{L}$. Therefore, it is possible to minimize the distance between two posterior distribution Q and P by maximizing $\mathcal{L}$ indirectly.

Reparameterization. As $\mathbf{z}$ is a random variable following the approximate posterior $q_{\theta_e}(\mathbf{z}|\mathbf{x})$, it is possible to directly sample the random variable $\mathbf{z}$ from a neural network with an input $\mathbf{x}$. Since the backpropagation cannot pass the random variable nodes, (Kingma and Welling 2013) introduce a new parameter ϵ which allows to reparameterize $\mathbf{z}$ that allows the backpropagation to flow through the deterministic nodes.

$$(\mu(\mathbf{x}), \Sigma(\mathbf{x})) = \mathrm{Encoder}(\mathbf{x}; \theta_e), \quad (4)$$
$$\mathbf{z} = \mu(\mathbf{x}) + \Sigma^{\frac{1}{2}}(\mathbf{x}) \cdot \epsilon, \text{where } \epsilon \sim \mathcal{N}(0,1).$$

Optimization. The objective of the optimization consists of two parts, 1) maximization of the likelihood and 2) minimization of the KL divergence in Eq. 3. The first objective is realized by two neural networks, an encoder (θ_e) and a decoder (θ_d). The latent representation $\mathbf{z}$ is obtained from $\mathbf{x}$ through the encoder network and $\hat{\mathbf{y}}$ is reconstructed through the decoder network. To maximize the conditional distribution $p(\mathbf{y}|\mathbf{z},\mathbf{x})$, we implement a loss function that depends on the difference between $\hat{\mathbf{y}}$ and $\mathbf{y}$. Then, all trainable parameters in the encoder and the decoder are updated through the backpropagation from the loss function to minimize itself.

The second objective enforces the approximate posterior $q(\mathbf{z}|\mathbf{x})$ to be close to the latent variable distribution $p(\mathbf{z})$. We can assume $p(\mathbf{z})$ as simple as possible, e.g., $\mathcal{N}(0,1)$, and the KL divergence between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$ could be provided in a closed form: $\mathrm{KL}(q(\mathbf{z}|\mathbf{x})||\mathcal{N}(0,1)) = \frac{k}{2}\left(\Sigma(\mathbf{x})) + \mu^2(\mathbf{x}) - 1 - \log \Sigma(\mathbf{x})\right)$ where k is the dimension of the latent variable. These two objectives can be optimized through the gradient descent.

Usage. GVM is trained by given pairs of (F_b,F_a) and once the parameters in the module are updated, it can generate f_a from F_b regardless of corresponding F_a.

2.3 Model Optimization

As noted above, there are a number of loss functions considered in the model optimization. First, there are two loss functions in GVM, reconstruction error and KL divergence:

$$\mathcal{L}_{GVM} = \mathbb{E}_Q[\log p(\mathbf{y}|\mathbf{z},\mathbf{x})] - KL(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z})) \quad (5)$$

In Eq. 5, we use mean squared error (MSE) to maximize $\mathbb{E}_Q[\log p(\mathbf{y}|\mathbf{z},\mathbf{x})]$ and $\mathcal{N}(0,1)$ for $P(\mathbf{z})$. Another set of loss functions is about the classification error, $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$, and $\mathcal{L}_5$. In this work, we use the softmax cross entropy for these losses. Thus, the overall joint cost function of our model is represented as: $\mathcal{L}_{tot} = \mathcal{L}_{GVM} + \lambda_1(\mathcal{L}_1 + \mathcal{L}_2) + \lambda_2(\mathcal{L}_3 + \mathcal{L}_5) + \lambda_3\mathcal{L}_4$ where λ_1, λ_2, and λ_3 are regularization parameters. In general, we can assign different regularization parameters for each loss function. For simplicity, we assign same parameter, λ_1, for $\mathcal{L}_1$ and $\mathcal{L}_2$ which are based on one feature (F_b or F_a) only. λ_2 is assigned for the loss functions handling concatenated vectors, $\mathcal{L}_3$ and $\mathcal{L}_5$. Finally, we separate $\mathcal{L}_4$ since the highest loss is expected from the loss function. The total loss function will be minimized jointly by stochastic gradient descent.

3 Gang-Related Crime Prediction

As discussed in the introduction, we are interested in identifying and predicting gang-related crimes. We begin by discussing our crime data and features. We then learn and evaluate our PGNN presented in Section 2 using the crime data.

3.1 LAPD Crime Data

We use a dataset provided by the Los Angeles Police Department. Our dataset consists of different types of crimes occurred in 2014, 2015, and 2016. Each crime record may have associated modus operandi codes (mocodes) providing some identification of unique behaviors or attributes associated with the crime. In the LAPD crime data, an mocode is used to flag a gang-related crime, which, as discussed, is labeled by officers in the criminal gang division. As discussed above, there is some debate about the relative importance of gang-related versus gang-affiliated crimes (Rosenfeld, Bray, and Egley 1999). We there concentrate on those crimes that are more consistently related to gang activity including homicide, aggravated assaults, and robberies, and do not focus on crimes are more likely to simply be committed by a gang member (e.g., burglary). These more serious violent crimes are expected to more consistently apply the correct gang-related labels compared to other types of crimes. Moreover, homicides, aggravated assaults, and robberies have the highest numbers of labeled gang-related crimes among other types of crimes (see Table 2). In particular, only about 1.3 percent of other crimes are gang-related.

Features. In the dataset, each crime has a set of categorical, textual, and numerical attributes/features recorded by the police officers or other analytical processes. Note that some of the features might not be applicable to some crimes, and they are treated as "blank" entries by the police officers. Similarly, other features might be recorded but are relevant to other bureaucratic processes. For example, the variable `LOC_TIER` is an ordinal measure of the quality

Crime	2014	2015	2016
AGG	10,625	13,808	15,585
(gang-related)	(1,873)	(2,431)	(2,643)
	17.63%	17.61%	16.64%
ROBB	7,933	8,994	10,283
(gang-related)	(1,056)	(1,148)	(1,213)
	13.31%	12.76%	11.80%
HOM	260	283	294
(gang-related)	(158)	(167)	(171)
	60.77 %	59.01%	58.16%
OTH	169,468	185,251	190,597
(gang-related)	(2,270)	(2,421)	(2,415)
	1.34%	1.31%	1.27%

Table 2: Crimes in 2014-2016. AGG = Aggravated Assaults, ROBB = Robberies, HOM = Homicides, OTH = Other Crimes

of the location coordinates generated automatically by the geocoding engine. The features of a type of crime consist of the premise (PREMIS), the point of entry (POENTRY), the street name (STREETNAME), the location region (RD) and division (DIV) in the LAPD system, the primary weapon (PRIMARYWEAPON), the property missing/stolen/destroyed (PROPERTY), the tier (LOC_TIER), the maker of the suspect vehicle (SUSPVEHMAKE), the maker of the victim vehicle information (VICVEHMAKE), the number of suspects (SUSPECTS), the sex of the victim (VICSEX), the narrative (NARRATIVE), the approximated day of the week (DOW), the date (MONTH and DAY), the hour (HOUR), the case status (CASESTATUS), the approximate day-span (WINDOW), and the match score (SCORE) of the crime.

Gang territory feature. In addition to using crime features, we use the 2009 gang territory data to specify the gang territory of each crime. In particular, for each crime, we record the name of the gang territory (GANGTERR) that corresponds to the location of the crime.

Collecting missing features. Each crime record is collected by police officers. Some features of the crimes are harder and require more effort to obtain than others. This is particularly the case for collecting narrative details about a crime, which is a time-consuming task. Moreover, such feature might not be readily available at test time when all other features are present. Yet, the narrative feature, as we will see later in the section, is a powerful feature for predicting gang-related crimes. As a result, we aim to build a classifier that would perform well on predicting crimes with (possibly) missing narratives. If the classifier performs exceptionally better with narrative text, then a policy recommendation would be to ensure that narrative text is always collected, perhaps in place of other data types that do not seem to offer as much information.

3.2 Feature Encoding and Selection

Since some features (e.g., VICSEX) are recorded as categorical types, we use one-hot encoding for the features. For the NARRATIVE feature, it is inappropriate to encode as a one-hot vector since the feature is described by natural language. To encode the description written in natural language to a numerical dense vector, we use *Word2vec* (Mikolov et al. 2013) as word embeddings. For each word of the narrative is represented by a vector, and we take the average of all the word vectors in a text record as a NARRATIVE-embedding.

Only some features are important to classify gang-related crimes. Some features are not much better than random guess (i.e., independent with the target), while other features show strong relationship with the target. Hence, we want to select *dependent* attributes as a set of input features before simply feeding all attributes into a classifier. To verify the dependency of each attribute, we repeatedly train a classifier with one attribute and validate the dependency of the attribute on the held-out dataset.

Feature	Type	Encoding	Dimension
GANGTERR	CAT	One-hot	262
NARRATIVE	Text	Word2vec	300
PREMISE	CAT	One-hot	143
PRIMARYWEAPON	CAT	One-hot	75
SUSPECTS	INT	Scalar	1

Table 3: Important features. CAT = Categorical, Int = Integer.

The attributes PREMISE, PRIMARYWEAPON, SUSPECTS, NARRATIVE, and GANGTERR yield AUROC$>$0.6 with low dimensionality, and we select them as the final set of features of interest (Table 3 lists their encodings and dimensions).

4 Experiments

In this section, we evaluate the proposed model PGNN on the LAPD crime datasets over 3 years (2014 - 2016). We extract three types of crimes (AGG, ROBB, and HOM) which are highly gang-related crimes as datasets for the evaluation.

Baseline. Many existing models for classification tasks can be directly used as baselines. Among these models, we choose representative models such as Logistic regression (LR), Support vector machine (SVM), Decision tree (DT), and Neural networks (NN) to compare with PGNN. We use cross validation to find the optimal hyperparameters for each model. For the neural networks model, we use the same networks (NN1 and NN2) in PGNN for the fair comparison. Finally, we evaluate a model replacing GVM in PGNN by a discriminative model (PDNN) to verify the effectiveness of the generative module. The discriminative module in PDNN is composed of several fully connected layers. First, F_a is connected with a layer reducing the dimension of the input vector to 50. Then, F_b is transferred to the reduced feature vector by passing the two fully connected layers. With these two layers, F_b can mimic F_a when F_a is unavailable and the mapped feature is used for filling the missing part.

Model setting. There are several tunable hyperparameters in PGNN. We first need to define the internal neural networks, NN1, NN2, and NN3. Since the size of F_b is around 500 (depending on years), we use two fully-connected layers for NN1 where the size of the output of each layer is reduced by half. For NN2 and NN3, a single layer is used to return output vectors. The size of the output dimension is a third of the input (*Word2vec*) dimension which is 300. The layers in NN1, NN2, and NN3 are trained with 0.5 dropout probability.

In GVM, we need to define parameters in the encoder and decoder. For the encoder, two fully-connected layers are used to reduce the dimension of the input vector. The dimension of the latent variable $\mathbf{z}$ is 10. The decoder composed

		LR	SVM	DT	NN	PDNN	PGNN
Full	2014	0.8437 (0.0142)	0.7615 (0.0181)	0.6304 (0.0204)	0.8677 (0.0102)	0.8898 (0.0182)	**0.9180** (0.0056)
	2015	0.8462 (0.0141)	0.7759 (0.0170)	0.6411 (0.0141)	0.8697 (0.0077)	0.8843 (0.0086)	**0.9239** (0.0087)
	2016	0.8449 (0.0134)	0.7730 (0.0139)	0.6196 (0.0413)	0.8649 (0.0066)	0.8742 (0.0117)	**0.9157** (0.0116)
Partial	2014	0.7566 (0.0192)	0.7377 (0.0172)	0.6115 (0.0380)	0.7905 (0.0131)	0.8214 (0.0188)	**0.8416** (0.0103)
	2015	0.7673 (0.0195)	0.7599 (0.0163)	0.6380 (0.0518)	0.7812 (0.0156)	0.8028 (0.0142)	**0.8595** (0.0152)
	2016	0.7725 (0.0147)	0.7536 (0.0160)	0.6195 (0.0414)	0.7911 (0.0079)	0.8018 (0.0117)	**0.8538** (0.0124)

Table 4: Experiment results. AUROCs are provided with the standard deviation.

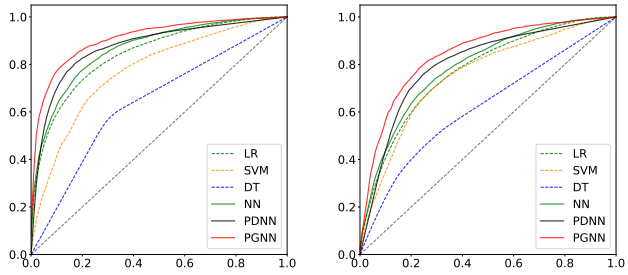
of two layers reconstructs F_a by increasing dimensions of outputs. A hyperbolic tangent function is used for all activation functions excluding the encoder (Eq. 4) which uses a linear activation function. We set the regularization parameters, $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, and $\lambda_3 = 0.1$ obtained from cross validation. Since $\mathcal{L}_4$ is expected to be larger, it is required to assign a smaller constant to reduce influence of the loss.

Experiment setting. We split the dataset into training and test sets. For handling the issue of imbalanced labels, we randomly sample 10% of gang-related crimes and the same number of non-gang-related samples for the test set, and the left samples are used for training. We repeat the evaluation 100 times to obtain robust results. We set the learning rate as 0.01 and use Adam optimizer (Kingma and Ba 2014).

There are two types of inputs at test time: 1) full information and 2) partial information. We train all models based on the full information (F_b and F_a). Once the models are trained, we use F_b and F_a for the former setting. We only use F_b for the latter case to see how models can handle the missing information. Since some baselines excluding PDNN are not flexible to read different dimensional inputs, we feed a zero vector for the missing part at test time for them.

Missing feature. For the gang-related crime classification, all features are not identically important. As Section 3 shows, we specifically use the important features for the classification. Although these features are essential for understanding details of a given crime, it is hard to expect that all features are available in the realistic setting. Among these features, `NARRATIVE` is a powerful feature for predicting gang-related crimes, however, collecting the feature requires much effort of domain experts and delays rapid predictions. Hence, we are interested in a case that the most important feature, `NARRATIVE`, is unavailable at test time. Under this setting, we can see how PGNN can improve the overall performance of the classification task by generating the missing feature and provide the potential for practical applications.

Result. Table 4 reports classification performance (AUROC) from baselines and PGNN with standard deviations on 3 year LADP crime datasets. The AUROCs indicated as 'Full' are results when the important features are fully available. In other words, `NARRATIVE` is available at test time as well. Thus, it is expected to provide the highest AUROCs from each classifier. On the other hand, 'Partial' denotes that `NARRATIVE` is only available at training and it is missed at test time. Thus, all baselines excluding PDNN are trained with the full features and tested on the partial features (`GEANGTERR`, `PREMISE`, `PRIMARYWEAPON`, and `SUSPECTS`). Figure 2 illustrates the ROC curves for every classifier on the 2016 crime data.



(a) Full features (b) Partial features

Figure 2: 2016 Crimes: ROC curves.

As Table 4 shows, PGNN outperforms other baselines on all datasets. Specifically, LR and NN provide comparable results, however, SVM and DT show worse classification quality. As expected, all classifiers show smaller AUROCs on the partial feature set compared to the full feature case. The results from NN tell us that the fully connected layers are effective to encode a given feature into a lower dimensional representation. Furthermore, we can find that the performance of one neural network (NN) can be enhanced by other side neural networks in PDNN or PGNN.

Compared to PDNN, PGNN is more robust and has higher AUROCs. It is believed that KL divergence in a generative model can regularize the model and help not to be susceptible on over-fitting. Hence, it provides that the GVM module can improve the overall classification quality. Interestingly, PGNN provides even better (or comparable) AUROCs under the partial setting than those of LR, SVM, DT, and NN under the full feature setting. This closeness clearly shows that PGNN can successfully generate missing features which are prevalent in the real world. Thus, we could find the effectiveness of generative networks to handle missing features.

5 Conclusion

We have presented the generative module that enables to generate missing features from partially available features and showed that it can be embedded into neural networks to successfully classify gang-related crimes. This procedure readily extends to other domains handling dependent feature sets. Finally, we compare PGNN with the discriminative classifiers to see that the generative module performs well.

Acknowledgement

This work is supported in part by NSF Research Grant IIS-1254206 and MINERVA grant N00014-17-1-2281. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

References

[Bjerregaard and Lizotte 1995] Bjerregaard, B., and Lizotte, A. J. 1995. Gun ownership and gang membership. *The Journal of Criminal Law and Criminology (1973-)* 86(1):37–58. 1

[Brantingham et al. 2012] Brantingham, P. J.; Tita, G. E.; Short, M. B.; and Reid, S. E. 2012. The ecology of gang territorial boundaries. *Criminology* 50(3):851–885. 1

[Che et al. 2016] Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2016. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*. 2

[Decker et al. 2014] Decker, S. H.; Pyrooz, D. C.; Sweeten, G.; and Moule, R. K. 2014. Validating self-nomination in gang research: Assessing differences in gang embeddedness across non-, current, and former gang members. *Journal of Quantitative Criminology* 30(4):577–598. 1

[Federal Bureau of Investigation 2015] Federal Bureau of Investigation. 2015. Crime in the united states. 1

[Gerber 2014] Gerber, M. S. 2014. Predicting crime using twitter and kernel density estimation. *Decision Support Systems* 61:115–125. 2

[Green, Horel, and Papachristos 2017] Green, B.; Horel, T.; and Papachristos, A. V. 2017. Modeling contagion through social networks to explain and predict gunshot violence in chicago, 2006 to 2014. *JAMA Internal Medicine*. 1, 2

[Hoffman, Gupta, and Darrell 2016] Hoffman, J.; Gupta, S.; and Darrell, T. 2016. Learning with side information through modality hallucination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 826–834. 2

[Jacobs and Wright 2006] Jacobs, B. A., and Wright, R. 2006. *Street justice: Retaliation in the criminal underworld*. Cambridge University Press. 1

[Kingma and Ba 2014] Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 6

[Kingma and Welling 2013] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*. 3, 4

[Klein and Maxson 2006] Klein, M. W., and Maxson, C. L. 2006. *Street gang patterns and policies*. New York: Oxford University Press. 1

[Kreindler and Lumsden 2012] Kreindler, D. M., and Lumsden, C. J. 2012. The effects of the irregular sample and missing data in time series analysis. *Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data* 135. 2

[Kuang, Brantingham, and Bertozzi 2017] Kuang, D.; Brantingham, P. J.; and Bertozzi, A. L. 2017. Crime topic modeling. *arXiv preprint arXiv:1701.01505*. 2

[Loughran et al. 2011] Loughran, T. A.; Paternoster, R.; Piquero, A. R.; and Pogarsky, G. 2011. On ambiguity in perceptions of risk: Implications for criminal decision making and deterrence. *Criminology* 49(4):1029–1061. 1

[Mikolov et al. 2013] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119. 5

[Mohler et al. 2011] Mohler, G.; Short, M.; Brantingham, P.; Schoenberg, F.; and Tita, G. 2011. Self-exciting point process modeling of crime. *Journal of the American Statistical Association* 106(493):100–108. 1

[Mohler et al. 2015] Mohler, G.; Short, M. B.; Malinowski, S.; Johnson, M.; Tita, G. E.; Bertozzi, A. L.; and Brantingham, P. J. 2015. Randomized controlled field trials of predictive policing. *Journal of the American Statistical Association* 110(512):1399–1411. 1, 2

[Mondal and Percival 2010] Mondal, D., and Percival, D. B. 2010. Wavelet variance analysis for gappy time series. *Annals of the Institute of Statistical Mathematics* 62(5):943–966. 2

[Papachristos 2009] Papachristos, A. V. 2009. Murder by structure: Dominance relations and the social structure of gang homicide 1. *American Journal of Sociology* 115(1):74–128. 1

[Raghunathan et al. 2001] Raghunathan, T. E.; Lepkowski, J. M.; Van Hoewyk, J.; and Solenberger, P. 2001. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey methodology* 27(1):85–96. 2

[Rehfeld et al. 2011] Rehfeld, K.; Marwan, N.; Heitzig, J.; and Kurths, J. 2011. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics* 18(3):389–404. 2

[Rosenfeld, Bray, and Egley 1999] Rosenfeld, R.; Bray, T. M.; and Egley, A. 1999. Facilitating violence: A comparison of gang-motivated, gang-affiliated, and nongang youth homicides. *Journal of Quantitative Criminology* 15(4):495–516. 1, 4

[Short et al. 2014] Short, M.; Mohler, G.; Brantingham, P. J.; and Tita, G. 2014. Gang rivalry dynamics via coupled point process networks. *Discrete & Continuous Dynamical Systems-Series B* 19(5). 1

[Stomakhin, Short, and Bertozzi 2011] Stomakhin, A.; Short, M. B.; and Bertozzi, A. L. 2011. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems* 27(11):115013. 2

[Wainwright, Jordan, and others 2008] Wainwright, M. J.; Jordan, M. I.; et al. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1(1–2):1–305. 3

[Wang, Gerber, and Brown 2012] Wang, X.; Gerber, M. S.; and Brown, D. E. 2012. Automatic crime prediction using events extracted from twitter posts. *SBP* 12:231–238. 2

[White, Royston, and Wood 2011] White, I. R.; Royston, P.; and Wood, A. M. 2011. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine* 30(4):377–399. 2