“MONET: Debiasing Graph Embeddings via the Metadata-Orthogonal Training Unit”
“Adversarial Learning for Debiasing Knowledge Graph Embeddings”

Presenter: Sriparna Ghosh

“MONET: Debiasing Graph Embeddings via the Metadata-Orthogonal Training Unit”

John Palowitch, Google Research

Bryan Perozzi, Google Research

Outline

1. Understanding the problem space
2. Understanding the why
3. Getting to what
4. The novel model - MONET
5. Diving deeper - Algorithm and Mathematical model
6. Analysis and Experiment Results

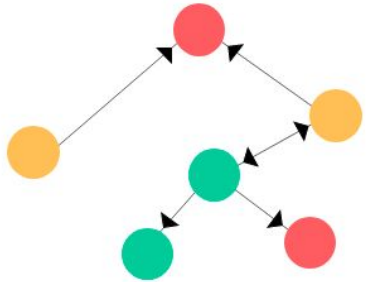
Understanding the problem space

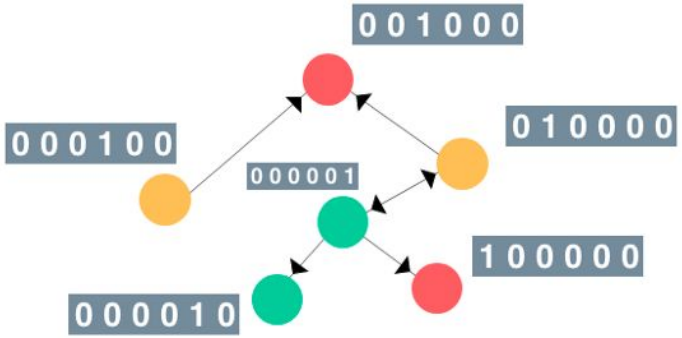
Graph Neural Network: Graph Neural Network, is a neural network that can directly be applied to graphs. It provides a convenient way for node level, edge level, and graph level prediction task

Graph Embeddings: continuous, low-dimensional vector representations of nodes.

Debiasing: reduction of bias, particularly with respect to judgment and decision making

Metadata: While graph embeddings can be estimated directly by unsupervised algorithms using the graph structure, there is often additional (non-relational) information available for each node in the graph, referred to as node attributes or node metadata, can contain information that is useful for prediction tasks including demographic, geo-spatial, and/or textual features.





Understanding the why

1. Despite the usefulness and prevalence of metadata in graph learning, sometimes it is **desirable to design a system to avoid the effects** of a particular kind of **sensitive data**.

   For example, the designers of a recommendation system may want to make recommendations independent of a user's demographic information or location.

2. What if we remove the sensitive data instead?

   Ignoring a sensitive attribute does not control for any existing correlations that may exist between the sensitive metadata and the edges of a node. In other words, if the edges of the graph are correlated with sensitive metadata, then any algorithm which does not explicitly model and remove this correlation will be biased as a result of it.

Getting to what - the proposal

To this end, we propose two simple desiderata for handling sensitive metadata in GNNs:

D1. The influence of metadata on the graph topology is modeled in a **partitioned subset of embedding space**, providing interpretability to the overall graph representation, and the option to remove metadata dimensions.

D2. Non-metadata or “topology" **embeddings are debiased** from metadata embeddings with a **provable guarantee** on the level of remaining bias.

The novel model - MONET

- A natural first approach to controlling metadata effects is to **introduce a partition of the embedding space** that models them explicitly, adding interpretability and separability to the overall representation
- The Metadata-Orthogonal Node Embedding Training (MONET) unit, a general neural network architecture component for performing training-time **linear debiasing of graph embeddings**
- MONET operates by ensuring that the node embeddings are **trained on a hyperplane orthogonal to that of the node features** (metadata).

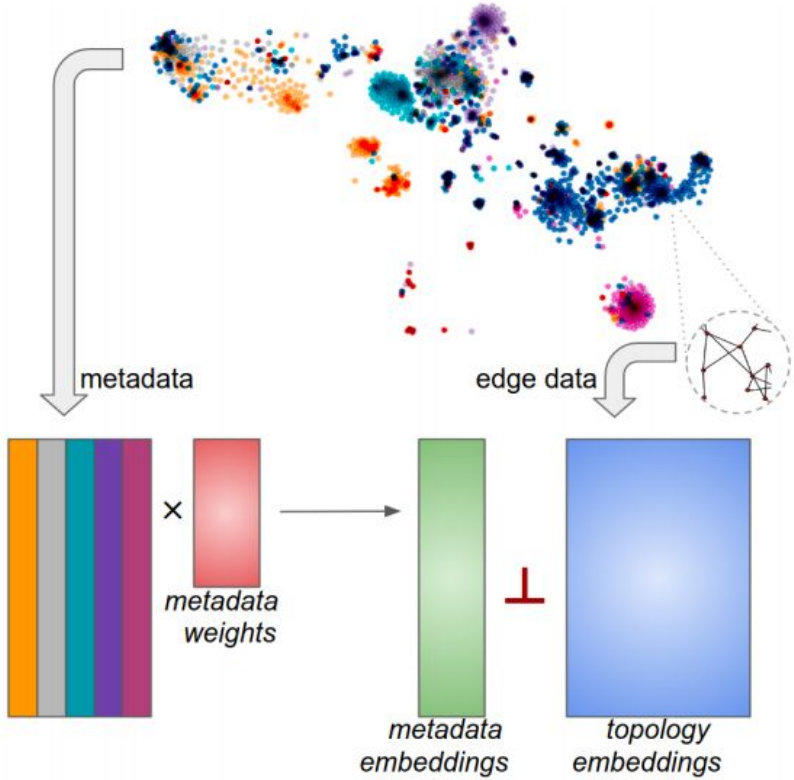


Figure 1: Illustration of the Metadata-Orthogonal Node Embedding Training (MONET) model. Points are nodes from a toy graph, represented in 2-D, colored by known metadata. There is clear correlation and clustering by the metadata. MONET encodes the metadata signal, then debiases standard topology embeddings from the metadata embeddings. As described in Section 3.3, both representations are trained concurrently and orthogonally.

Contributions

1. The **Metadata-Orthogonal Node Embedding Training (MONET)** unit, a novel GNN algorithm which jointly embeds graph topology and graph metadata while enforcing linear decorrelation between the two embedding spaces.

2. **Analysis** which proves that addressing desiderata D1 alone – partitioning a metadata embedding space – still produces a biased topology embedding space, and that the MONET unit corrects this.

3. **Experimental results** on real world graphs which show that MONET can **successfully debias** topology embeddings while relegating metadata information to separate dimensions – achieving superior results to state-of-the-art adversarial and NLP-based debiasing methods

Diving deeper - Algorithm and Mathematical model

How embeddings work

Embeddings are a type of algorithm used in graph pre-processing with the goal to turn a graph into a computationally digestible format

1. Early graph embedding methods involved **dimensionality reduction** techniques like multidimensional scaling and singular value decomposition
2. In this paper we use graph neural networks trained on random walks, similar to **DeepWalk**
3. DeepWalk and many subsequent methods first generate a sequence of random walks from the graph, to create a “corpus" of node “sentences" which are then modeled via word embedding techniques (e.g. word2vec or GloVe) to learn low dimensional representations that preserve the observed co-occurrence similarity.

The Glove Model

$$L_{\text{GloVe}} = \sum_{i,j \leq n} f_\alpha(C_{ij})(a_i + b_j + U_i^T V_j - \log(C_{ij}))^2.$$

Word embedding technique to learn low dimensional representations that preserve the observed co-occurrence similarity. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

- W is a d-dimensional graph embedding matrix, $W \in R^{n \times d}$
- U, V $\in R^{n \times d}$ are the “center" and “context" embeddings
- a, b $\in R^{n \times 1}$ are the biases
- C is the walk-distance-weighted context co-occurrences, and
- f_α is the loss smoothing function

Jointly Modeling Metadata & Topology D1

$$L_{\text{GloVe}_{\text{meta}}} = \sum_{i,j \leq n} f_\alpha(C_{ij})(a_i + b_j + U_i^T V_j + X_i^T Y_j - \log(C_{ij}))^2.$$

To achieve D1, we feed M through a single-layer neural network with weights T

word embedding techniques (e.g. word2vec (17) or GloVe (21)) to learn low dimensional representations that preserve the observed co-occurrence similarity

- Metadata matrix as $M \in R^{n \times m}$
- Two weight matrices T1, T2 are needed to embed the metadata for both center and context nodes
- This network yields metadata embeddings X, Y , which correspond to the standard topology embeddings U, V (respectively), and can be concatenated to form a combined metadata and topology vector

$$L_{\text{GloVe}_{\text{meta}}} = \sum_{i,j \leq n} f_\alpha(C_{ij})(a_i + b_j + U_i^T V_j + X_i^T Y_j - \log(C_{ij}))^2.$$

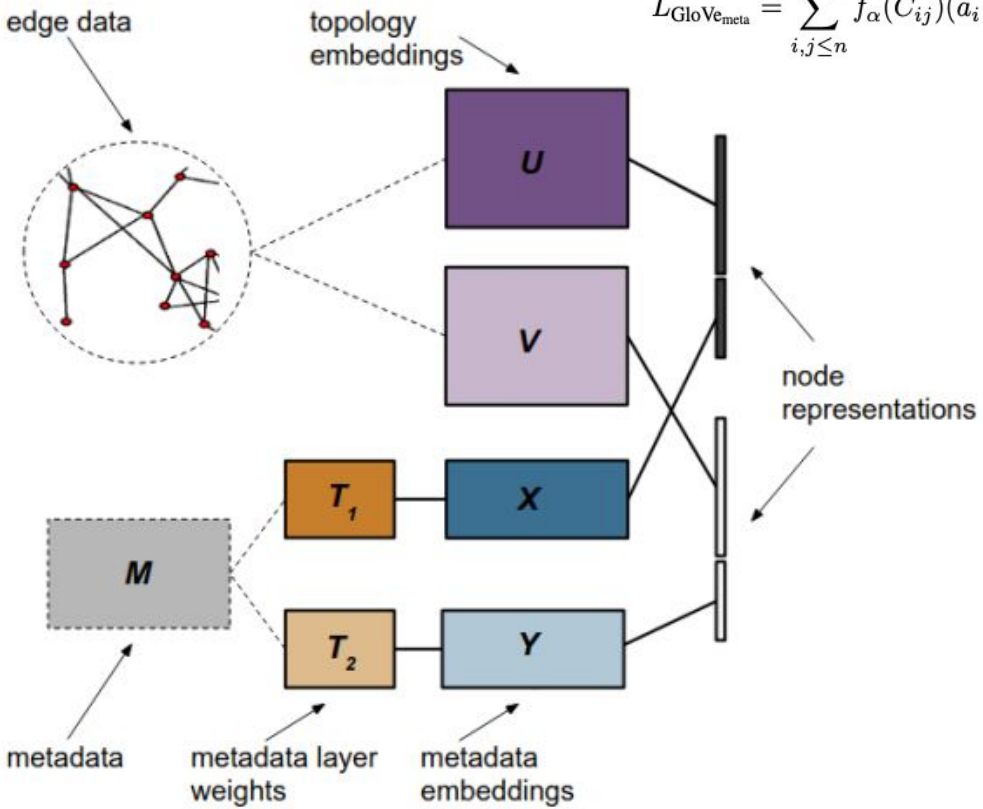


Figure 2: Illustration of GloVe$_{meta}$. U and V are topology embedding layers. A single-layer feed-forward neural network creates metadata embeddings layer X and Y. The concatenations $[U, X]$ and $[V, Y]$ create the output node representations.

Metadata Leakage - Need for D2

- This model provides users of graph embeddings with the **option to include or exclude** metadata learned dimensions
- Suppose that the metadata (e.g. demographic information) are indeed associated with the formation of links in the graph
- We find – empirically and theoretically – that this naïve approach **does not guarantee** that the **topology embeddings are completely decorrelated** from the metadata embeddings
- Surprisingly, we find that most of the metadata **bias** suffered by standard baselines **remain in the topology dimensions**
- This is a phenomenon we call ***metadata leakage***

The metadata leakage of metadata embeddings $Z \in R^{n \times dZ}$ ***into topology embeddings*** $W \in R^{n \times d}$ ***is defined*** $ML(Z, W) := ||Z^T W||^2_F$ ***. We say that there is no metadata leakage if and only if*** $ML(Z, W) = 0$

*$||\cdot||_F$ ***denotes the Frobenius norm***

Metadata leakage Demonstration

$$L^*_{\text{GloVe}_{\text{meta}}} = \sum_{i,j \le n} f_\alpha(C_{ij})(a_i + a_j + W_i^T W_j + Z_i^T Z_j - \log(C_{ij}))^2$$

Metadata-laden GloVe Loss

- Specifically, let (i, j) be a node pair from C, and **define δ_W (i, j) as the incurred Stochastic Gradient Descent update W' ← W + δ_W (i, j).**
- Suppose there is a "ground-truth" metadata transformation B ∈ R m×dB , and define ground-truth metadata embeddings $\tilde{Z}$:= MB, which represent the "true" dimensions of the metadata effect on the co-occurrences C
- Define $\Sigma_B := BB^T$ and $\Sigma_T := T T^T$
- Define $\mu_W := E[W_i]$ and $\Sigma_W := E[W_i W^T{}_i]$ Define μ_M, Σ_M similarly

Theorem 1

- Assume $\Sigma_W = \sigma_W I_d$ for $\sigma_W > 0$, $\mu_W = 0_{d\times 1}$, and $\mu_M = 0_{m\times 1}$
- Suppose for some fixed θ ∈ R we have $\log(C_{ij}) = \theta + Z_i^{\tilde{}T} \tilde{Z}_j$
- Let (i, j) be a randomly sampled co-occurrence pair and W' the incurred update
- Then if $E[M_i W_i^T] = \beta \in R^{m\times d}$

$$\mathbb{E}[\mathcal{ML}(Z,W')] \ge 2||T^T[\Sigma_M(\Sigma_B - \Sigma_T) + (n - \sigma_W)I_m]\beta||_F^2.$$

Corollary 1

- Σ_T and σ_W are neural network hyperparameters
- Under the assumptions of Theorem 1

$$\mathbb{E}[\mathcal{ML}(Z,W)] = \Omega(n||T^T\beta||_F^2) \text{ as } n \to \infty.$$

- Note that under reasonable GNN initialization schemes, T and β are random perturbations.
- Thus, C**orollary 1 implies the surprising result that incorporating a metadata embedding partition is not sufficient to prevent metadata leakage in practical settings**

MONET

- The Metadata-Orthogonal Node Embedding Training (MONET) unit for training joint topology-metadata graph representations [W, Z] without metadata leakage

- MONET explicitly prevents the correlation between topology and metadata, by using the Singular Value Decomposition (SVD) of Z to orthogonalize updates to W during training.

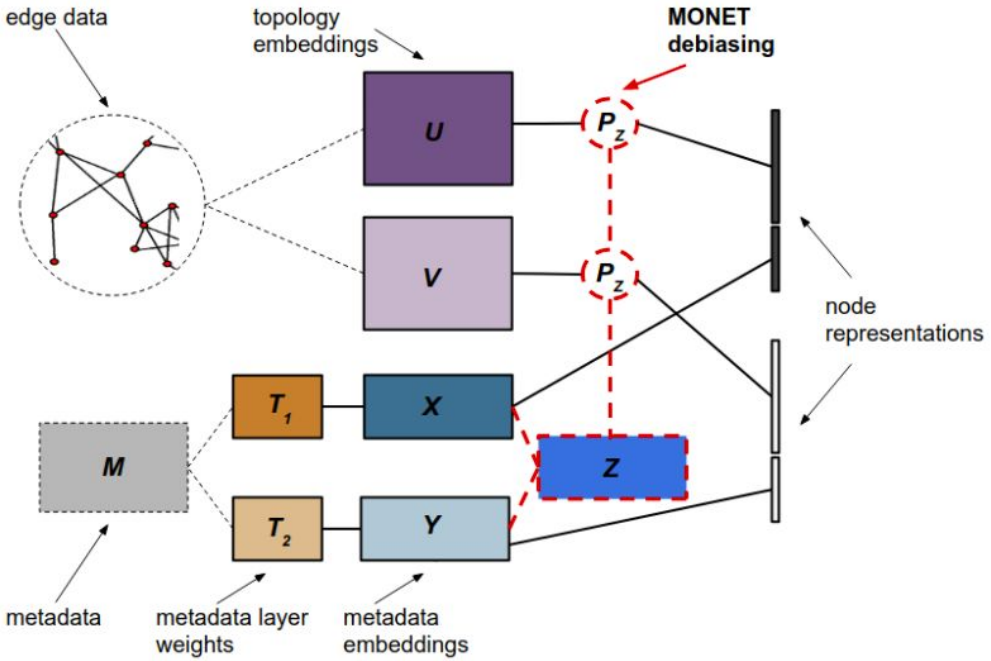


Figure 3: The MONET unit adds a feed-forward transformation of the metadata, resulting in metadata embeddings X and Y. $Z = X + Y$ gives the combined metadata representation, used to debias U and V via P_Z.

MONET Algorithm

- The MONET unit is a two-step algorithm applied to the training of a topology embedding in a neural network
- The input to a MONET unit is a metadata embedding $Z \in R^{n \times dz}$ and a target topology embedding $W \in R^{n \times d}$ for debiasing
- Then, let Q_Z be the left-singular vectors of Z, and define the projection $P_Z := I_{n \times n} - Q_Z Q_Z^T$
- In the forward pass procedure, debiased topology weights are obtained by using the projection $W\perp = P_Z W$
- Similarly $W\perp$ is used in place of W in subsequent GNN layers
- In the backward pass, MONET also debiases the backpropagation update to the topology embedding, δ_W , using $\delta^{\perp}_W = P_Z \delta_W$
- Straightforward properties of the SVD show that MONET directly prevents metadata leakage

Algorithm 1: MONET Unit Training Step

Input: topology embedding W, metadata embedding Z
procedure FORWARD PASS DEBIASING(W, Z)
  Compute Z left-singular vectors Q_Z and projection: $P_Z \leftarrow I_{n \times n} - Q_Z Q_Z^T$
  Compute orthogonal topology embedding: $W^{\perp} \leftarrow P_Z W$
  return debiased graph representation $[W^{\perp}, Z]$
end procedure
procedure BACKWARD PASS DEBIASING(δ_W)
  Compute orthogonal topology embedding update: $\delta_W^{\perp} \leftarrow P_Z \delta_W$
  Apply update: $W^{\perp} \leftarrow W^{\perp} + \delta_W^{\perp}$
  return debiased topology embedding $W^{\perp}$
end procedure

Theorem 2. *Using Algorithm 1,* $\mathcal{ML}(Z, W^{\perp}) = 0$ *and* $\mathcal{ML}(Z, \delta^{\perp}) = 0$.

- Note that in this work we have only considered linear metadata leakage; provable guarantees for debiasing nonlinear topology/metadata associations is an area of future work

Implementation

- We demonstrate MONET in our experiments by applying Algorithm 1 to Eq. (2), which we denote as MONET_G.
- We orthogonalize the input and output topology embeddings U, V with the summed metadata embeddings Z := X + Y
- By linearity, this implies Z-orthogonal training of the summed topology representation W = U + V
- We note that working with the sums of center and context embeddings is the standard way to combine these matrices

Relaxation

- A natural generalization of the MONET unit is to parameterize the level of orthogonalization incurred at each training step
- Introduce a parameter $\lambda \in [0.0, 1.0]$ which controls the extent to which topology embeddings are projected onto the metadata-orthogonal hyperplane

$$P_Z^{(\lambda)} := I_{n \times n} - \lambda Q_Z Q_Z^T$$

- Using MONET with $\lambda = 1.0$ removes all linear bias, whereas using $\lambda = 0.0$ prevents any debiasing

Analysis and Experiment Results

Experiment 1: Debiasing a Political Blogs Graph

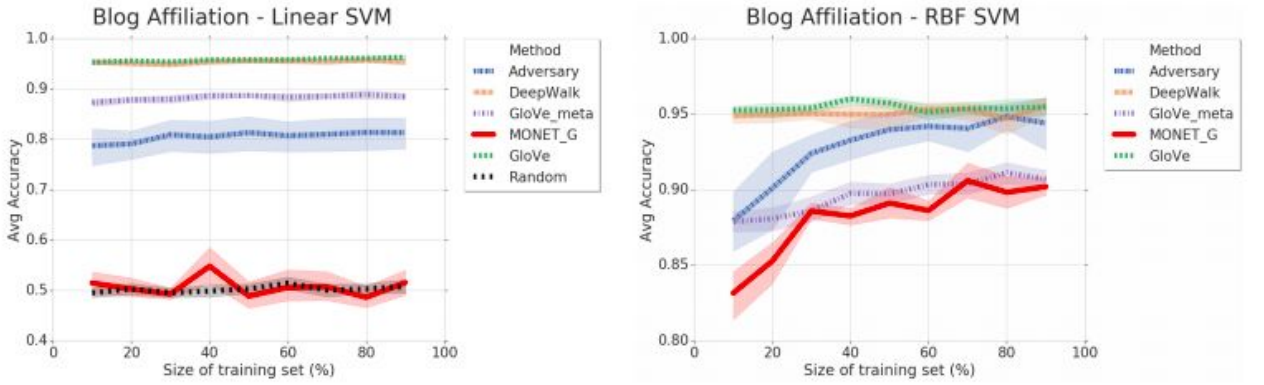
- The effect of MONET debiasing by removing the effect of political ideology from a blogger network.
- The political blog network has has 1,107 nodes corresponding to blog websites, 19,034 hyperlink edges between the blogs (after converting the graph to be undirected), and two clearly defined, equally sized communities of liberal and conservative bloggers.
- Compare MONET$_G$ against the following methods: (1) a random topology embedding generated from a multivariate Normal distribution; (2) DeepWalk; (3) GloVe; (4) GloVemeta; (5) an adversarial version of GloVe, in which a 2-layer MLP discriminator is trained to predict political affiliation

Method	$\mathcal{ML}$	Metadata Importance Σ_T
GloVe	6503 ± 196	N/A
Random	283 ± 22	N/A
Adversary	4430 ± 305	N/A
DeepWalk	2670 ± 104	N/A
GloVe$_{meta}$	2103 ± 183	$\begin{pmatrix} .110 \pm .01 & -.106 \pm .01 \\ -.110 \pm .01 & .106 \pm .01 \end{pmatrix}$
MONET$_G$	$\mathbf{.021 \pm .003}$	$\begin{pmatrix} .187 \pm .01 & -.182 \pm .01 \\ -.187 \pm .01 & .182 \pm .01 \end{pmatrix}$

Table 1: When trained to debias political affiliation, MONET removes metadata leakage to precision error. Metadata Importance values - computed from layers of GloVe$_{meta}$ and MONET$_G$ - show that the MONET unit allows stronger metadata learning.

- This demonstrates the downstream effect of Theorem 1, and shows the necessity of desiderata D2 in a rigorous approach to graph embedding debiasing.
- Note that MONET also outperforms the adversarial extension of GloVe, even though adversarial approaches have performed well in similar domains

Experiment 1: Debiasing a Political Blogs Graph



(a) Sensitive attribute prediction using linear model (b) Sensitive attribute prediction using nonlinear model

Figure 4: Accuracy results from blog political affiliation classifiers. Lower is better (embeddings more debiased). MONET$_G$ achieves this perfectly with a linear classifier, due to the MONET unit's novel orthogonalization technique. Interestingly, on this task, MONET$_G$ is dominant even under a nonlinear classifier.

- Surprisingly, here we find that MONET$_G$ still produces the least biased representations, even beating an adversarial approach, even though the MONET unit only performs linear debiasing
- Showing that MONET can reduce bias even when its representations are used as features inside of a non-linear model.

Experiment 1: Debiasing a Political Blogs Graph

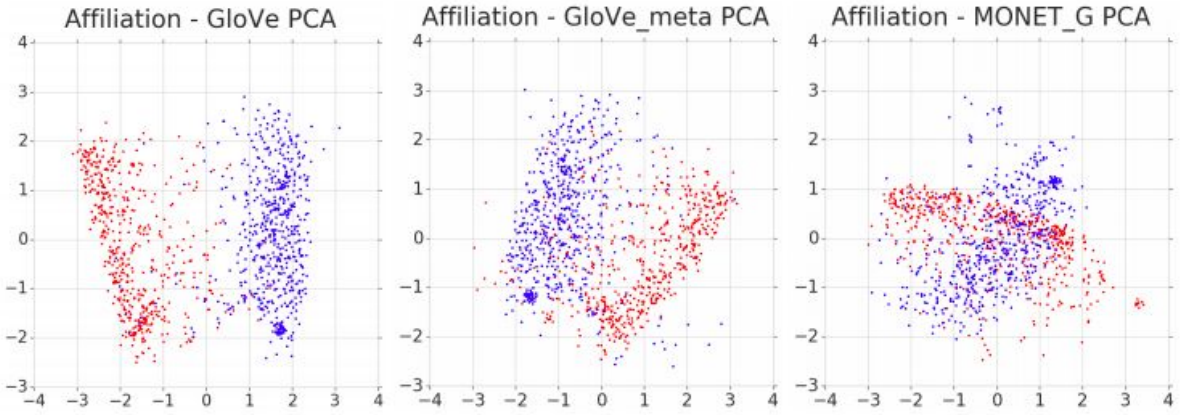


Figure 5: PCA of political blog graph embeddings. (a): affiliation separation clearly visible on standard GloVe embeddings. (b): affiliation separation reduces when GloVe$_{meta}$ captures some metadata information. (c): affiliation separation disappears with MONET$_G$ orthogonalized training.

- Fig 5 shows the 2-D PCA of each models' 16-dimensional topology embeddings, colored by political affiliation. Clear separation by affiliation is visible with the GloVe model's PCA. The PCA of GloVemeta also shows separation, but less so.
- This shows that having a metadata partition in embedding space – desiderata D1 – does some work toward debiasing the topology dimensions.

Experiment 2: Debiasing a Shilling Attack

- A shilling attack is where a number of users act together to artificially increase the likelihood that a particular influenced item will be recommended for a particular target item.
- In this experiment, the authors explore the context of using debiasing methods to defend against a shilling attack on graph-embedding based recommender systems
- **Setup:** In a single repetition of this experiment, the authors inject an artificial shilling attack into the MovieLens 100k dataset .
- The raw data is represented as a bipartite graph with 943 users, 1682 movies (“items"), and a total of 100,000 ratings (edges). Each user has rated at least 20 items.

Method	W_{GloVe} Distances r	Wall Time (sec)
DeepWalk	0.347 ± 0.003	45 ± 4
GloVe	1.000 ± 0.000	357 ± 54
GloVe$_{meta}$	0.790 ± 0.002	359 ± 49
MONET$_G$	0.758 ± 0.004	442 ± 58
Random	0.031 ± 0.001	N/A
NLP	0.770 ± 0.008	N/A

Table 2: Shilling experiment performance metrics: “W_{GloVe} Distances r” for a method is the Pearson correlation between (1) the cosine embedding distances produced by GloVe topology embeddings and (2) those same distances produced by the method. These figures show that MONET$_G$ does not overly corrupt the GloVe embedding signal, and does not add prohibitive computation time.

- The topology embeddings from MONET$_G$($\lambda = 1.0$) prevent the most attacker bias by a large margin, letting less than 1 influenced item into the top-20 neighbors of ti
- MONET$_G$ outperforms the random baseline by at least 8.5x, and is comparable to one baseline (DeepWalk).
- NLP debiasing method outperformed MONET$_G$ in terms of ranking accuracy, but allowed a surprising number of attacked items (~7.5) in top-20 lists
- relaxing MONET’s orthogonality constraint (λ, the level of orthogonal projection), decreases the effective debiasing, but improves embedding performance (as might be expected)

Experiment 2: Debiasing a Shilling Attack

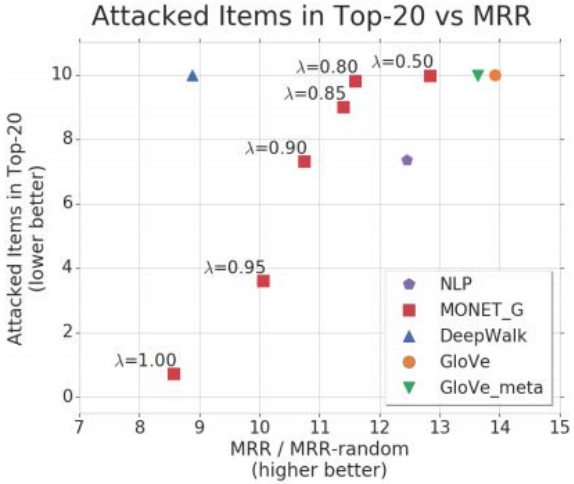


Figure 6: Shilling experiment accuracy-bias trade-off: the x-axis is the lift-above-random of the mean reciprocal rank (MRR) of embedding-based item retrieval against item random-walk co-occurrence ordering. The y-axis is the number of attacked items in the top-20 embedding-distance nearest-neighbors of the target item t_i. All points are averages over ten runs of the shilling experiment. Only MONET$_G$ completely prevents attack bias, and maintains comparable performance to DeepWalk and greater than 8x better than a random baseline. Confidence intervals along both axes are provided in a supplemental table.

“Adversarial Learning for Debiasing Knowledge Graph Embeddings”

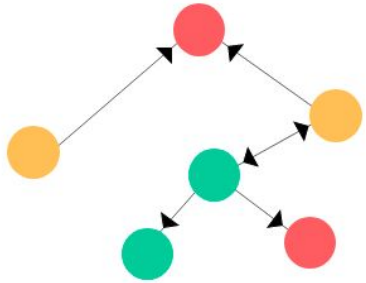
Mario Arduini, Lorenzo Noci, Federico Pirovano, Ce Zhang, Yash Raj Shrestha, and Bibek Paudel

Outline

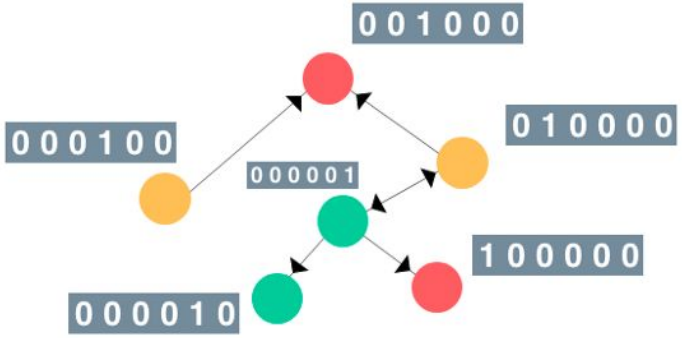
1. Understanding the problem space
2. Understanding the why
3. Getting to what
4. The novel model - FAN
5. Data
6. Analysis and Experiment Results

Understanding the problem space

Knowledge Graph Multi-relational graphs, composed of entities (nodes) and edges representing semantic meaning. Example: Google Knowledge Graph for Search



Adversarial Learning Adversarial machine learning is a machine learning technique that attempts to fool models by supplying deceptive input.



Understanding the why

- Recent research have identified social and cultural biases embedded in the representations learned from KGs
- Such biases can have detrimental consequences on different population and minority groups as applications of KG begin to intersect and interact with social spheres
- Besides a few exceptions, research work on identification and mitigation of such social bias in KGs remains absent.
- Absence of coherent and useful debiasing framework for KG is problematic and could lead to detrimental societal consequences particularly with respect to protected class of individuals.

Getting to what

- Explored **popularity bias** — the relationship between node popularity and link prediction accuracy
- Found that in node2vec graph embeddings prediction accuracy of the embedding is negatively correlated with the degree of the node
- knowledge-graph embeddings (KGE), were observed to have an opposite trend
- Explored **gender bias** in KGE, and a careful examination of popular KGE algorithms suggest that sensitive attribute like the gender of a person can be predicted from the embedding
- **This implies that such biases in popular KGs is captured by the structural properties of the embedding.**
- Introduced a **novel framework to filter out the sensitive attribute** information from the KG embeddings, which we call FAN (Filtering Adversarial Network)
- Findings suggest that **sensitive attributes in KGs** are not only represented by explicit relations having the name of such attributes, but rather they **are expressed by the whole graph structure**
- An important implication of this finding is the necessity of **fine-grained debiasing algorithm** operating on the embeddings, **instead of just removing the sensitive relations** from the KG.

Popularity Bias

- Popularity bias is the bias resulting from correlation between the degree of a node in a graph and the accuracy of link prediction of the embeddings of the nodes
- Since network embeddings are also increasingly used in search and recommendations, such biases could affect these downstream tasks and lead to the lack of diversity and filter-bubbles in users' online experiences
- To investigate whether network embeddings exhibit popularity bias, the authors examined the popular node2vec method on the benchmark AstroPh dataset
- The network consists of 187, 22 nodes and 198, 110 edges

Gender Bias

- KGEs might exhibit several societal biases, like race, gender, religion, etc
- Explore how different attributes interact in KGEs and how to remove sensitive attributes (e.g., gender, race) from the embedding while preserving all the other information
- The authors treat gender as a sensitive attribute and perform occupation prediction (which in this case is posed as an unbalanced multiclass classification problem) training a simple neural network operating in the embedding space
- Measure the interaction between the gender sensitive attribute and the occupation nonsensitive attribute, and use this information as evidence for the existence of bias

The novel model - FAN

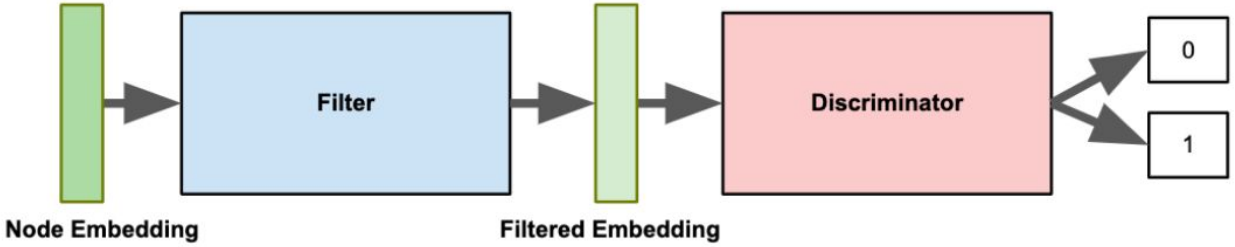


Figure 1: FAN model: the filter takes in input a vector and output its filtered version (ideally without the sensitive attribute). The discriminator tries to predict the sensitive attribute (in the figure it is assumed to be binary) from the filtered embedding, and ideally will reach an accuracy of 50% (random prediction).

- The model is an adversarial network composed of two players, **a filter module $F_{\theta f} : R^d \rightarrow R^d$** that aims at filtering the information about the sensitive attribute from the input, and **a discriminator module $D_{\theta d} : R^d \rightarrow [0, 1]$** that aims to predict the sensitive attribute from the output of the filter.
- The filter aims at minimizing the reconstruction loss.
- The discriminator aims at minimizing the cross-entropy loss
- The optimum saddle point is reached when the discriminator cannot predict the sensitive attribute from the filtered input

Data

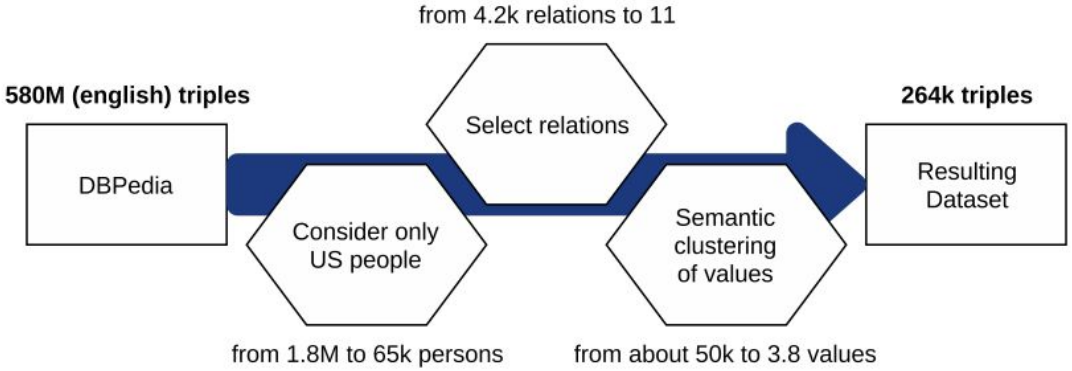


Figure 4: DBpedia dataset preprocessing.

- DBpedia is a crowd-sourced community effort to extract structured content from the information created in various Wikimedia projects — provides a unique research context to examine the questions.
- Structured information curated in DBpedia is available for everyone on the web and resembles an Open Knowledge Graph (OKG).
- This last version of DBpedia had about 200k triples and 44 occupations, leading to fast training and meaningful embeddings.

Analysis and Experiment Results

- Observe that the model is able to remove the gender information from the embeddings, making it impossible for the classifier to predict the gender, while at the same time keeping constant performance for occupation prediction
- For the unfiltered embeddings, observe that the classifier can predict the gender and the performance improves for high-degree nodes, indicating that the gender information is not only contained in the gender relation
- These shows that the filtering network FAN is able to remove gender biases from the KGEs from both high degree and low-degree entities.

	Prediction Task	*TransH* (834 epochs)	*TransH* (999 epochs)	*TransD* (834 epochs)	*TransD* (912 epochs)
Unfiltered	Gender	0.67	0.67	0.68	0.68
	Occupation	0.49	0.49	0.49	0.49
$\lambda = 0.5$	Gender	0.50	0.52	0.50	0.51
	Occupation	0.48	0.49	0.44	0.43
$\lambda = 0.05$	Gender	0.44	0.54	0.51	0.51
	Occupation	0.49	0.48	0.41	0.43

Table 1: Results of our debiasing algorithm. Columns represent different embedding algorithm and the number of training epochs. Rows denote three debiasing approaches: unfiltered embeddings and two applications of FAN with different λ values. For each embedding algorithm, the top value shows gender prediction accuracy and the bottom value shows the occupation prediction accuracy. Our debiasing algorithm is able to reduce the accuracy of gender prediction to a random event, without hurting the occupation prediction accuracy.

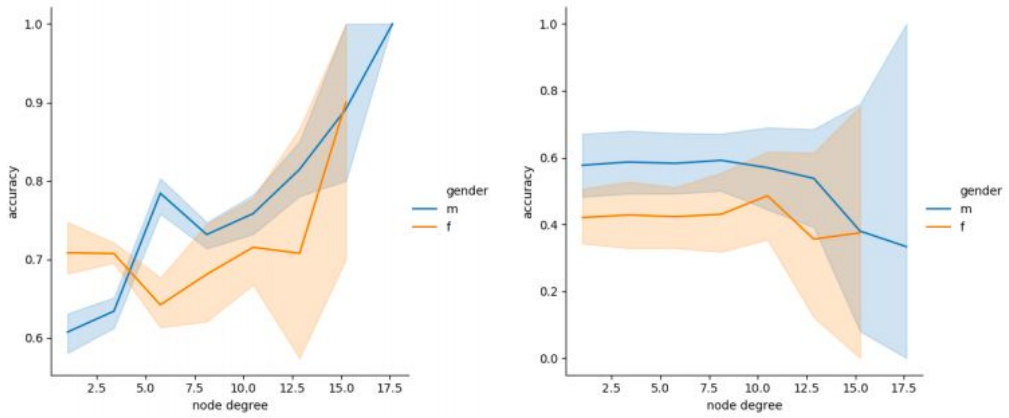


Figure 2: Left: gender prediction accuracy against node degree for *unfiltered* embeddings. Right: gender prediction accuracy against node degree for *filtered* embeddings generated by our FAN algorithm. Colored bands represent confidence intervals. Results are obtained starting from embeddings trained with TransH for 999 epochs. We can see that FAN is able to remove gender bias from both high- and low-degree entities.

Questions?