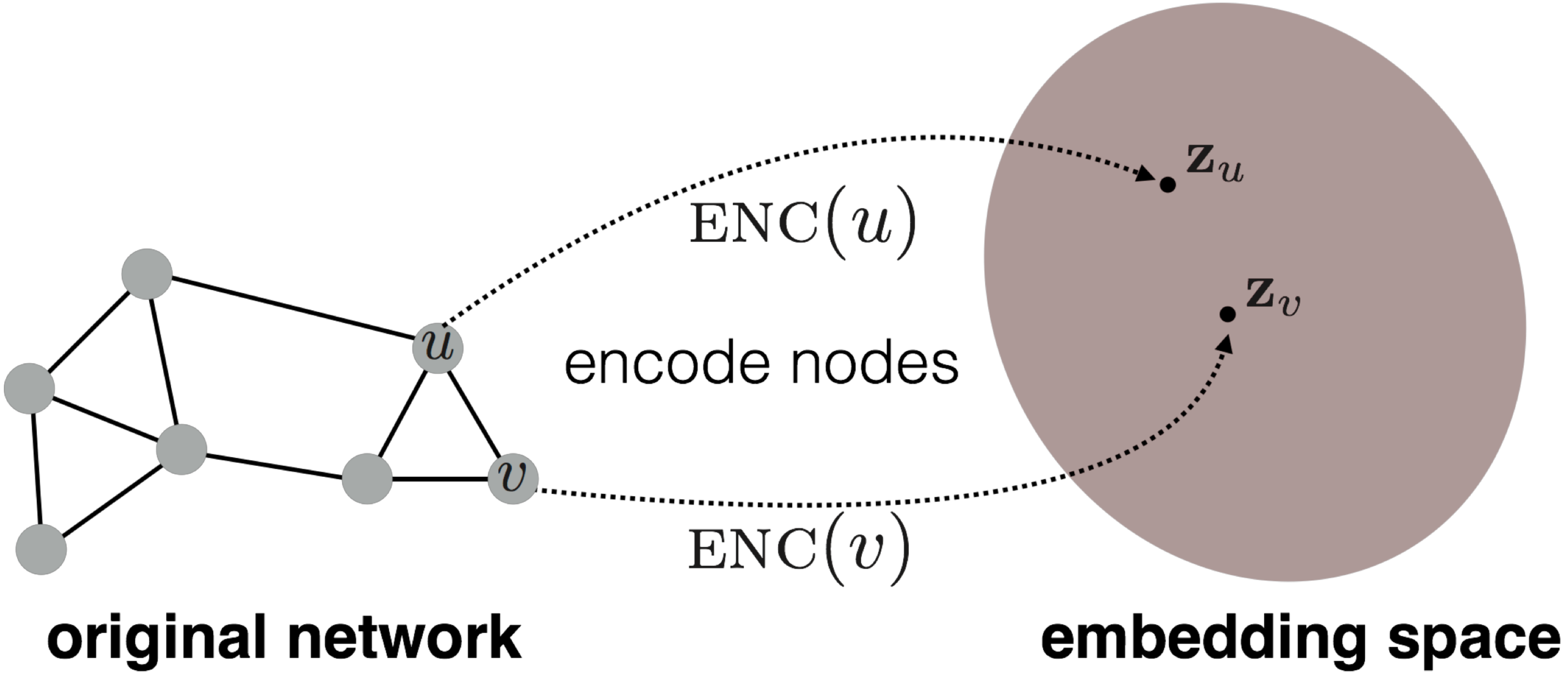
Fairness Constraints for Graph Embeddings*

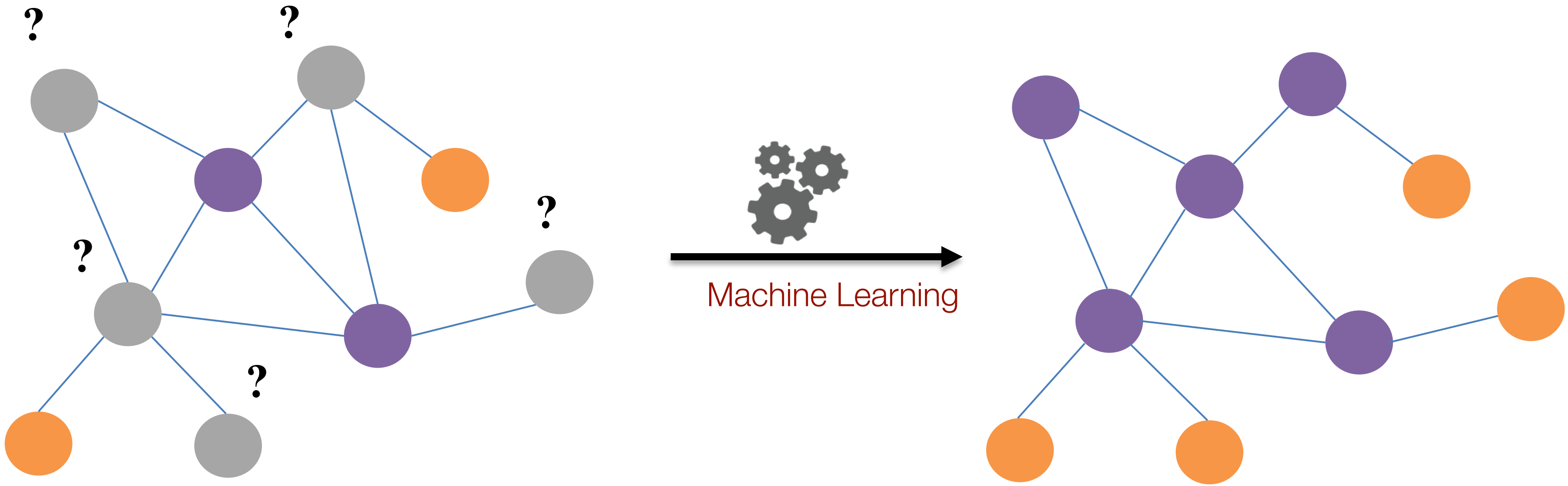
William L. Hamilton
Assistant Professor at McGill University and Mila
Canada CIFAR Chair in AI
Visiting Researcher at Facebook AI Research

*Joint work with my PhD student Joey Bose, to appear in ICML 2019 (pdf)

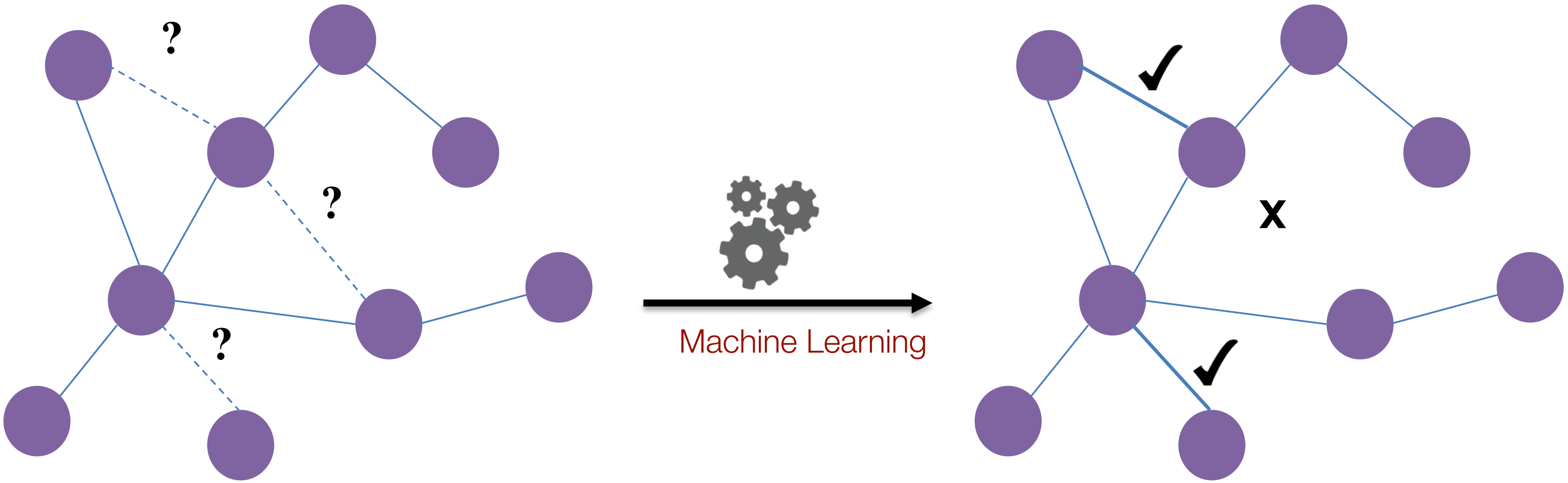
Graph embeddings



Application: Node classification



Application: Link prediction



Becoming ubiquitous in social applications

- Graph embedding techniques are a powerful approach for social recommendations, bot detection, content screening, behavior prediction, geo-localization,
  - E.g., Facebook, Huawei, Uber Eats, Pinterest, LinkedIn, WeChat

- Classic collaborative filtering approaches can be re-interpreted in a more general graph embedding framework.

But what about fairness and privacy?

- Graph embeddings designed to capture everything that might be useful for the objective.
- Even if we don’t provide the model information about sensitive attributes (e.g., gender or age), the model will use this information.
- What if a user doesn’t want this information used?

Fairness from a pragmatic perspective

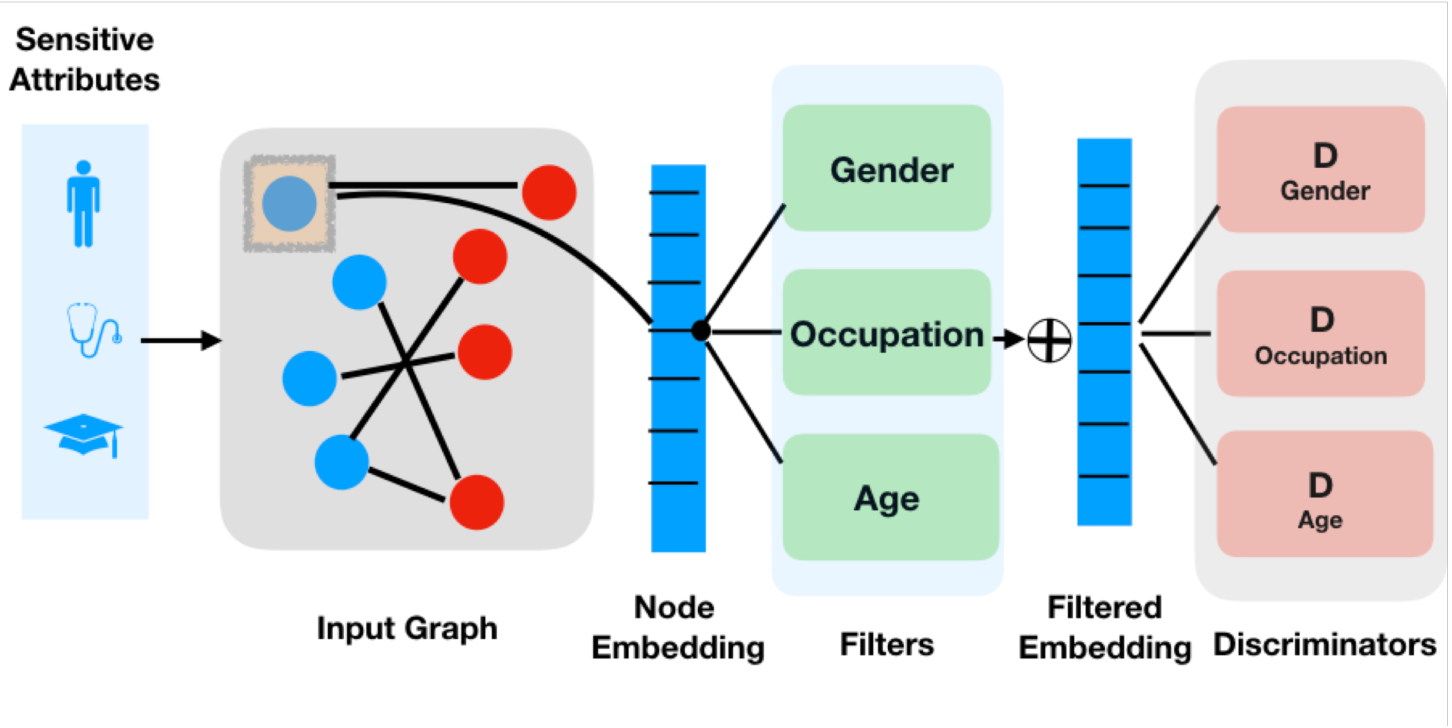
- Strict privacy and discrimination concerns are one motivation.
- But what if users just don't want their recommendations do depend on certain attributes?
- What if users want the system to "ignore" parts of their demographics or past behavior?

Fairness in graph embeddings

- Basic idea: How can we learn node embeddings that are invariant to particular sensitive attributes?

- Challenges:
  - Graph data is not i.i.d.
  - There is not just one classification task that we are trying to enforce fairness on.
  - There are often many *possible* sensitive attributes.

Our work: Fairness in graph embeddings



Preliminaries and set-up

- Learning an encoder function to map nodes to embeddings:

$$\mathbf{z}_v = \mathrm{ENC}(v)$$

- Using these embeddings to “score” the likelihood of a relationship between nodes:

$$s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) \qquad s(e) > s(e'), \forall e \in \mathcal{E}, e' \in \bar{\mathcal{E}}.$$

Preliminaries and set-up

- Learning an encoder function to map nodes to embeddings:

$$\mathbf{z}_v = \text{ENC}(v)$$

- Using these embeddings to "score" the likelihood of a relationship between nodes:

$$s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) \qquad s(e) > s(e'), \forall e \in \mathcal{E}, e' \in \bar{\mathcal{E}}.$$

Score of a (possible) edge is a function of the two node embeddings and the relation type.

Preliminaries and set-up

- Learning an encoder function to map nodes to embeddings:

$$\mathbf{z}_v = \mathrm{ENC}(v)$$

- Using these embeddings to "score" the likelihood of a relationship between nodes:

$$s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) \qquad s(e) > s(e'), \forall e \in \mathcal{E}, e' \in \bar{\mathcal{E}}.$$

Goal: Train the embeddings (with a subset of the true edges) so that the score for all real edges is larger than all non-edges.

Preliminaries and set-up

- Generic loss function:

$$\sum_{e\in\mathcal{E}_{\mathtt{train}}} L_{\mathrm{edge}}\left(s(e), s(e_1^-), ..., s(e_m^-)\right)$$

Sum over (batch of) training edges.

Task-specific loss function

Score assigned to positive/real edge.

Scores assigned to random negative sample edges.

Preliminaries and set-up: Concrete examples

- Score functions:

- Loss-functions:

Preliminaries and set-up: Concrete examples

- Score functions:
  - Dot-product: $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) = \mathbf{z}_u^\top \mathbf{z}_v$
- Loss-functions:

Preliminaries and set-up: Concrete examples

- Score functions:
  - Dot-product: $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) = \mathbf{z}_u^\top \mathbf{z}_v$
  - TransE: $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) = -\|\mathbf{z}_u + \mathbf{r} - \mathbf{z}_v\|_2^2$
- Loss-functions:

Preliminaries and set-up: Concrete examples

- Score functions:
  - Dot-product: $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) = \mathbf{z}_u^\top \mathbf{z}_v$
  - TransE: $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) = -\|\mathbf{z}_u + \mathbf{r} - \mathbf{z}_v\|_2^2$
- Loss-functions:
  - Max-margin: $L_{\text{edge}}(s(e), s(e_1^-), ..., s(e_m^-)) = \sum_{i=1}^{m} \max(1 - s(e) + s(e_i^-), 0)$

Preliminaries and set-up: Concrete examples

- Score functions:
  - Dot-product: $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) = \mathbf{z}_u^\top \mathbf{z}_v$
  - TransE: $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle) = -\|\mathbf{z}_u + \mathbf{r} - \mathbf{z}_v\|_2^2$
- Loss-functions:
  - Max-margin: $L_{\text{edge}}(s(e), s(e_1^-), ..., s(e_m^-)) = \sum_{i=1}^{m} \max(1 - s(e) + s(e_i^-), 0)$
  - Cross-entropy: $L_{\text{edge}}(s(e), s(e_1^-), ..., s(e_m^-)) = -\log(\sigma(s(e)) - \sum_{i=1}^{m} \log(1 - \sigma(s(e_i^-))$

Formalizing fairness

- How do we ensure fairness in this context?

Formalizing fairness

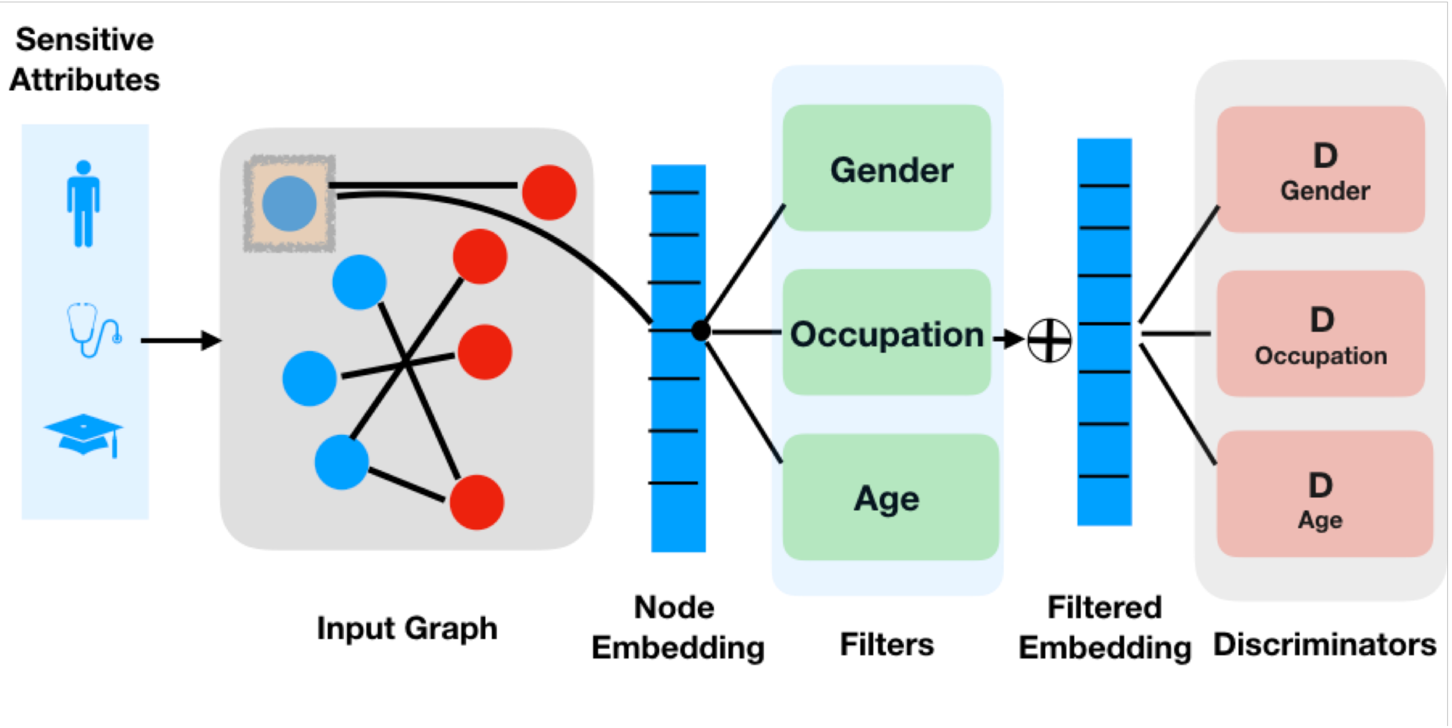
- How do we ensure fairness in this context?
- Solution: representational invariance
  - Want embeddings to be independent from the attributes:

$$\mathbf{z}_u \perp a_u, \quad \forall u \in \mathcal{V}$$

  - Which is equivalent to minimizing the mutual information to between the embeddings and the attributes:

$$I(\mathbf{z}_u, a_u^k) = 0, k \in S, \forall u \in \mathcal{V}$$

Enforcing fairness through an adversary



Enforcing fairness through an adversary

- Key component 1: Compositional encoder.
- Given a set of attributes, it outputs “filtered” embeddings that should be invariant to those attributes.

$$\text{C-ENC}(u, S) = \frac{1}{|S|}\sum_{k\in S} f_k(\text{ENC}(u))$$

Input: node ID and set of sensitive attributes

Sum over all sensitive attributes

Trainable filter function (neural network) outputs embedding that is invariant to attribute *k*.

Enforcing fairness through an adversary

- Key component 2: Adversarial discriminators
- For each sensitive attribute, train an adversarial discriminator that tries to predict that sensitive attribute from the filtered embeddings:

$$D_k(\text{C-ENC}(u, S), a^k)$$

Output: Likelihood that node *u* has that attribute value.

Discriminator for sensitive attribute *k*.

Input: Filtered embeddding for node *u* and attribute value.

Enforcing fairness through an adversary

- Putting it all together in an adversarial loss:

Original loss function for the edge prediction task

$$L(e) = L_{\text{edge}}(s(e), s(e_1^-), ..., s(e_m^-))$$
$$+ \lambda \sum_{k \in S} \sum_{a^k \in \mathcal{A}_k} \log(D_k(\text{C-ENC}(u, S), a^k))$$

Constant that determines the strength of the fairness constraints

Likelihood of discriminator predicting the sensitive attributes.

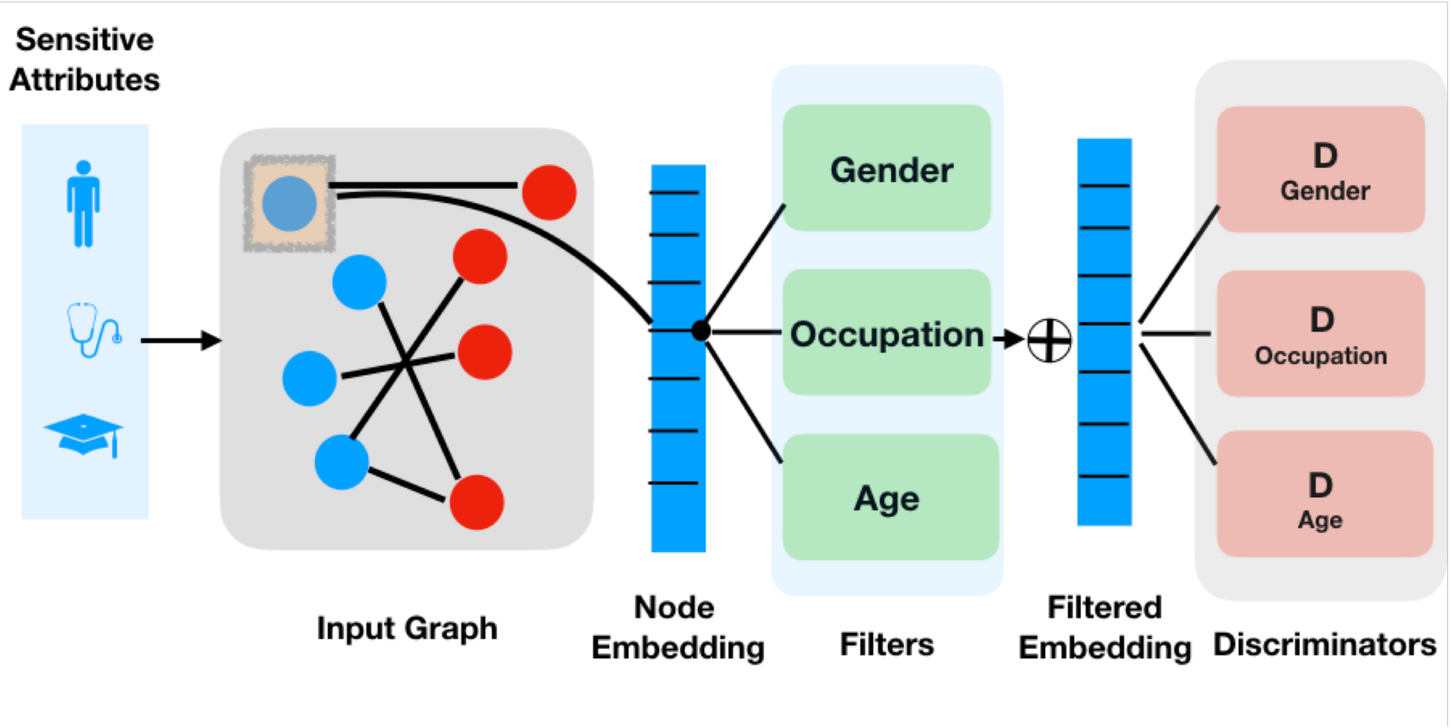
Enforcing fairness through an adversary

- Putting it all together in an adversarial loss:

$$L(e) = L_{\text{edge}}(s(e), s(e_1^-), ..., s(e_m^-)) + \lambda \sum_{k \in S} \sum_{a^k \in \mathcal{A}_k} \log(D_k(\text{C-ENC}(u, S), a^k))$$

- During training the encoder tries to minimize this loss and the adversarial discriminators are trained to maximize it.

Enforcing fairness through an adversary



Dataset 1: MovieLens-1M

- Classic recommender system benchmark.
- Bipartite graph between users and movies.

- Nodes (~10,000): Users and movies
- Edges (~1,000,000): Rating a user gives a movie
- Sensitive attributes:
  - Gender
  - Age (binned to become a categorical attribute)
  - Occupation

Dataset 2: Reddit

- Derived from public Reddit comments.
- Bipartite graph between users and communities.

- Nodes (~300,000): Users and communities
- Edges (~7,000,000): Whether a user commented on that community
- Sensitive attributes: Randomly select 50 communities to be "sensitive" communities

Dataset 3: Freebase 15k-237

- Derived from classic knowledge base completion benchmark.
- Knowledge graph between set of typed entities.

- Nodes (~15,000): Users and communities
- Edges (~150,000): 237 different relation types (e.g., married_to, born_in, capital_of, director_of)
- Sensitive attributes: Randomly selected 3 entity type annotations (e.g., is_actor) to be “sensitive attributes”

Experiments: Three questions

1. What is the cost of invariance?

2. What is the impact of compositionality?

3. Can we generalize to unseen combinations of attributes?

MovieLens: Fairness results

- How strongly can we enforce fairness?
- Compare three approaches to enforcing fairness:
  - No adversary (i.e., just train on the recommendation task)
  - Independent adversarial model for each attribute
  - Full compositional model

MovieLens1M	Baseline No Adversary	Gender Adversary	Age Adversary	Occupation Adversary	Comp. Adversary	Majority Classifier	Random Classifier
Gender	0.712	0.532	0.541	0.551	0.511	0.5	0.5
Age	0.412	0.341	0.333	0.321	0.313	0.367	0.141
Occupation	0.146	0.141	0.108	0.131	0.121	0.126	0.05

MovieLens: Fairness results

- How strongly can we enforce fairness?
- Evaluate how well a two-layer MLP can classify the sensitive attributes from the learned node embeddings.
  - AUC for the binary gender attribute
  - Micro-averaged F1-score for the age and occupation attributes.

MovieLens1M	Baseline No Adversary	Gender Adversary	Age Adversary	Occupation Adversary	Comp. Adversary	Majority Classifier	Random Classifier
Gender	0.712	0.532	0.541	0.551	0.511	0.5	0.5
Age	0.412	0.341	0.333	0.321	0.313	0.367	0.141
Occupation	0.146	0.141	0.108	0.131	0.121	0.126	0.05

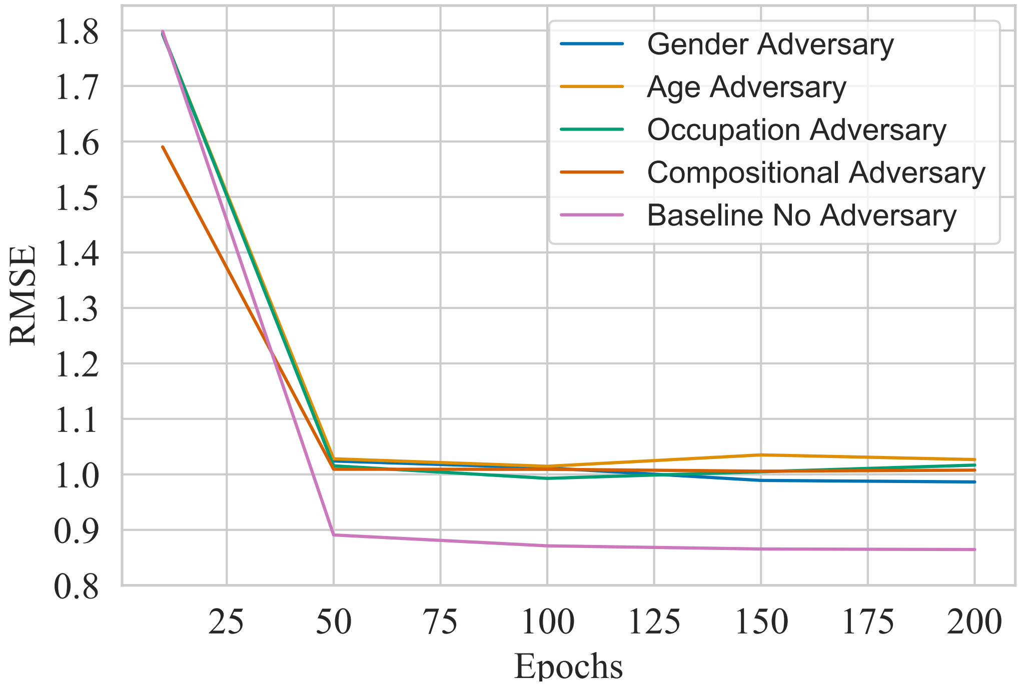
MovieLens: Fairness results

- Key takeaways:
  - After applying the compositional adversary, accuracy is no better than majority classifier!
  - Performance of compositional adversary on par with independent adversaries!

MovieLens1M	Baseline No Adversary	Gender Adversary	Age Adversary	Occupation Adversary	Comp. Adversary	Majority Classifier	Random Classifier
Gender	0.712	0.532	0.541	0.551	0.511	0.5	0.5
Age	0.412	0.341	0.333	0.321	0.313	0.367	0.141
Occupation	0.146	0.141	0.108	0.131	0.121	0.126	0.05

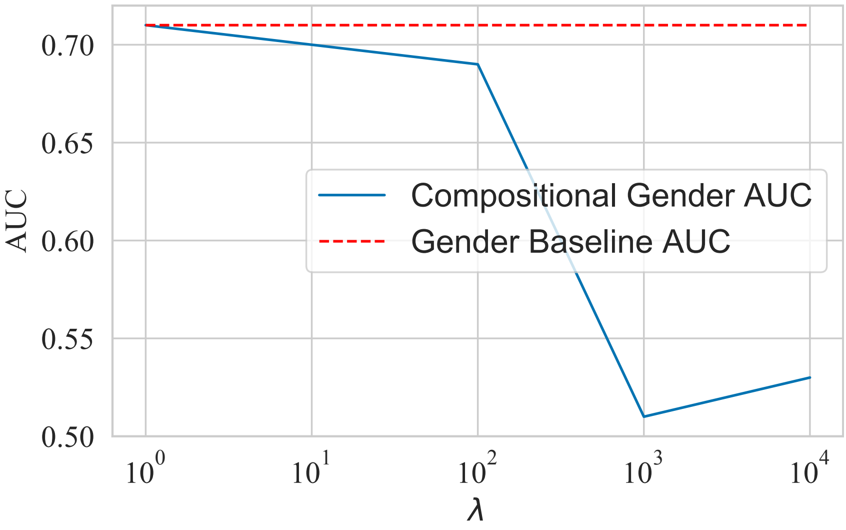
MovieLens: Impact on recommendations

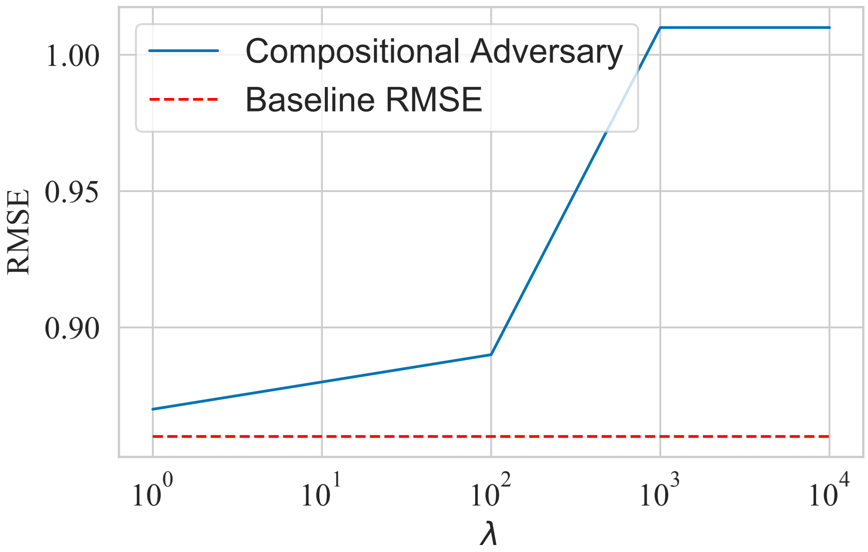
- Evaluate recommendation performance (RMSE) with and without enforcing fairness.
- There is a drop in accuracy, but not catastrophic.



MovieLens: Trade-off

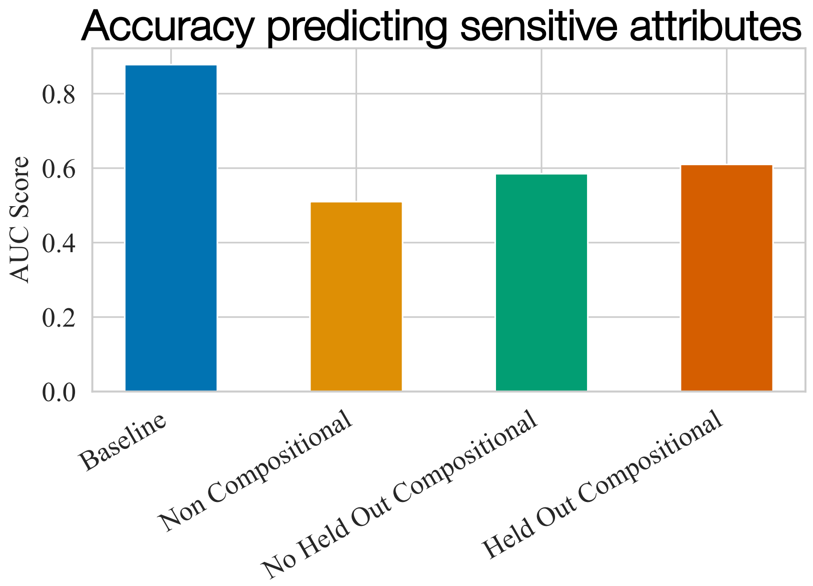
- λ allows trade-off between fairness and recommendation performance.

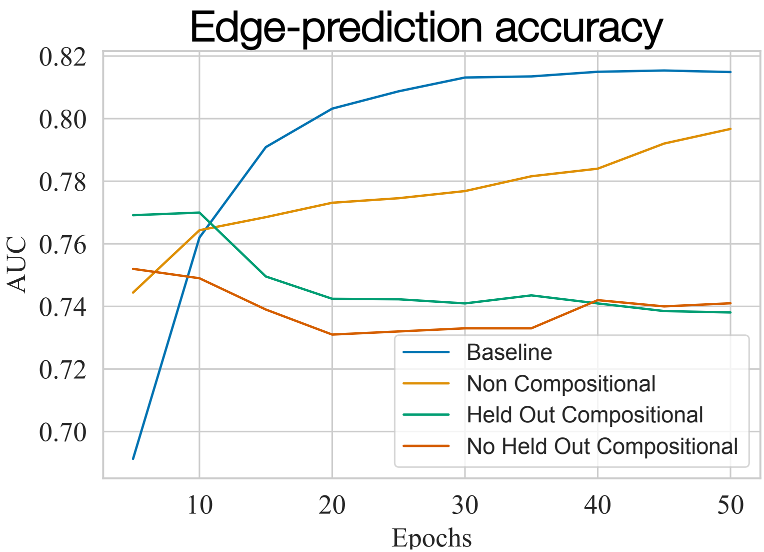




Reddit results: Fairness

- Same set-up as MovieLens, but here we have 10 sensitive attributes.
- Again, able to strongly enforce fairness, but at a non-trivial cost.





Freebase results

- On the synthetic Freebase data we see that enforcing fairness leads to a significant drop in task performance.

Ability to predict sensitive attributes (measured in AUC) and the impact on task-performance (mean rank)

FB15K-237	Baseline No Adversary	Non Comp. Adversary	Comp. Adversary
Attribute 0	0.97	0.82	0.77
Attribute 1	0.99	0.81	0.79
Attribute 2	0.98	0.81	0.81
Mean Rank	285	320	542

Conclusions and outlook

- Fairness in network representation learning is an understudied issue.
- We can enforce fairness in a flexible way, but at a cost.
- There is no perfect notion of fairness.