

ERGAN: Generative Adversarial Networks for Entity Resolution

Jingyu Shao ¹, Qing Wang ¹, Asiri Wijesinghe ¹, and Erhard Rahm ²

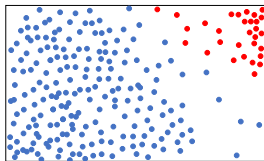
¹Research School of Computer Science
Australian National University

²Database Group, University of Leipzig

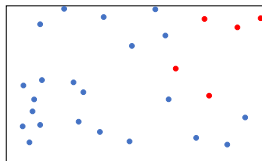
November 20, 2020

Acknowledgement: This work was partially funded by the Australian Research Council (ARC) under Discovery Project DP160101934

Two main challenges in solving Entity Resolution (ER) tasks:

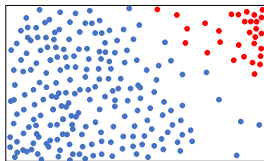


(a) Entire Data Distribution

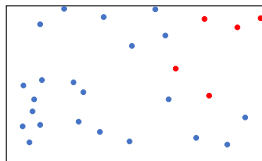


(b) Samples with limited labels

Two main challenges in solving Entity Resolution (ER) tasks:



(a) Entire Data Distribution

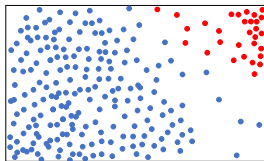


(b) Samples with limited labels

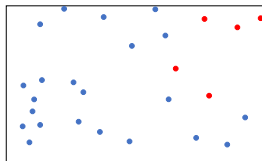
- The imbalanced class problem:

The number of matches (record pairs referring to the same entity) is far less than the number of non-matches.

Two main challenges in solving Entity Resolution (ER) tasks:



(a) Entire Data Distribution



(b) Samples with limited labels

- The imbalanced class problem:

The number of matches (record pairs referring to the same entity) is far less than the number of non-matches.

- The overfitting problem:

The number of labeled instances is limited and a learning model is powerful enough to remember all the features of training instances.

Generative adversarial network (GAN) is a powerful technique for image generation and natural language processing (NLP).

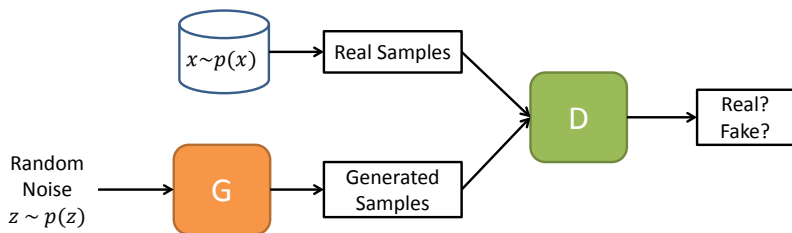
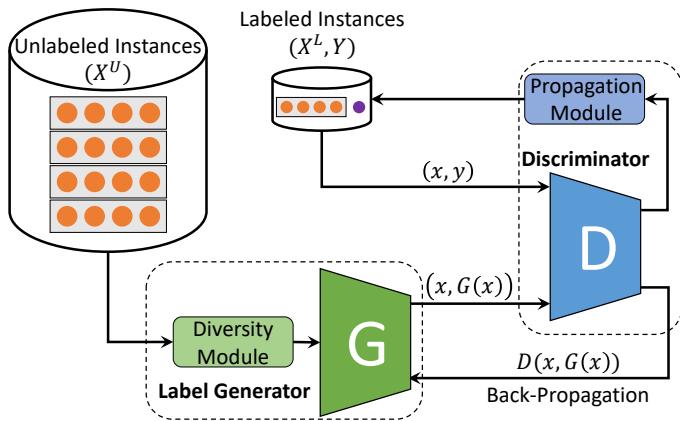
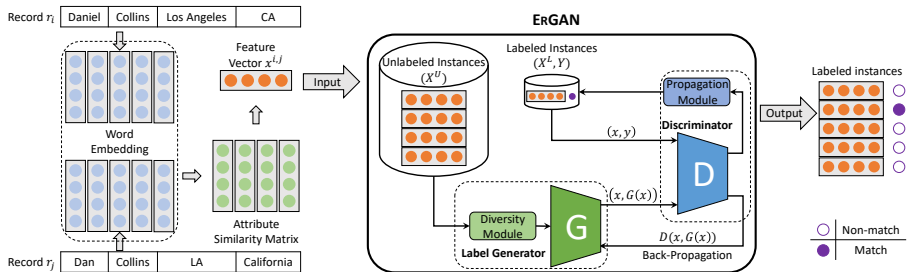
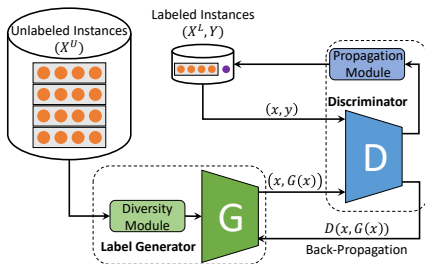


Figure: An overview of GANs



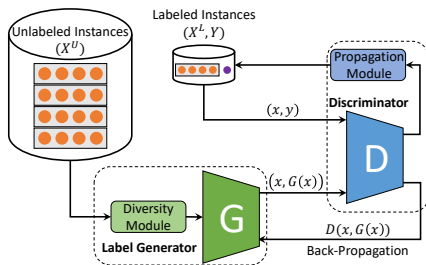
Framework Overview





The *label generator* G - aims to generate pseudo labels for unlabeled instances.

- The goal of the label generator G is to learn a conditional distribution $p_g(Y|X^U) \approx p(Y|X^U)$.
where X^U refers to all unlabeled instances and Y refers to their labels.
- To simulate the conditional distribution $p(Y|X^U)$, the label generator G receives feedback (i.e. gradients) from the discriminator D and is trained iteratively through backpropagation.



The *diversity module* enriches the diversity of both labeled and unlabeled instances during the sampling process.

- Different from GANs, we consider the diversity of instances in the minibatch sampling process.
- For all instances in X , we have $X = \bigcup_{i=1}^b X_i$ and $\bigwedge_{1 \leq i \neq j \leq b} X_i \cap X_j = \emptyset$.
where X_i refers to subspaces.

- A minibatch of m instances is selected from X^U according to the following objective function:

$$\text{maximize} \quad \|\mathbf{v}\|_{2,1} \quad \text{s.t.} \quad \sum_{i,j} v_i^j = m$$

- Let $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_b)$ be a vector corresponding to b subspaces, and $\|\mathbf{v}\|_{2,1}$ is a $l_{2,1}$ -norm function, i.e.

$$\|\mathbf{v}\|_{2,1} = \sum_{i=1}^b \|\mathbf{v}_i\|_2 = \sum_{i=1}^b \sqrt{\sum_{j=1}^{n_i} v_i^j{}^2}$$

where:

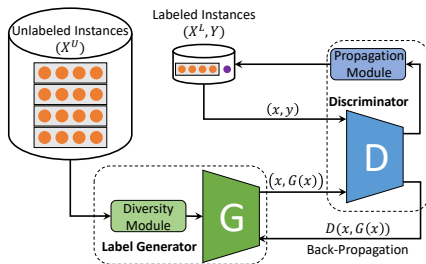
- $\mathbf{v}_i = (v_i^1, \dots, v_i^{n_i})^T$;
- $\mathbf{v}_i \in [0, 1]^{n_i}$;
- $n_i = |X_i^U|$.

- G updates its parameters according to the following objective function:

$$\mathcal{L}_G = \min_G \mathbb{E}_{x \sim p(x_i^U)} [\log(1 - D(x, G(x)))] \quad (1)$$

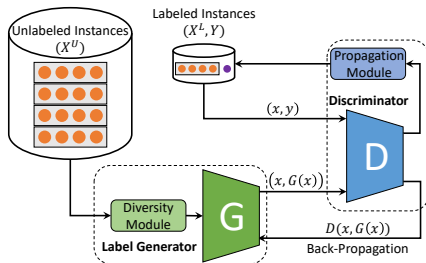
where:

- $G(x_i)$ is the pseudo label of x_i generated by G ;
- $x_i, G(x_i)$ is a pseudo labeled instance sent to the discriminator D ;
- $D(x, G(x))$ is the feedback from the discriminator D .



The *discriminator* D - aims to distinguish instances with pseudo labels from instances with real labels.

- The goal of D is to distinguish whether a labeled instance $(x, G(x))$ is from the real distribution $p(X, Y)$
- Given a pair $(x, G(x))$ as input, D generates a scalar value in $[0, 1]$ to indicate the probability that $G(x)$ is the same as the real label y of x .



The *propagation module* guarantees the selection of high-quality unlabeled instances for training the discriminator D when the labeled instances are not sufficient.

- Different from GANs, the discriminator D in ERGAN is designed to approximate the true joint distribution $p(X, Y)$ progressively through a propagation module.
- The higher score of $D(x_i, G(x_i))$ indicates the higher correctness of $G(x_i)$ to the real label y_i .
- A minibatch of γ pseudo labeled instances are propagated according to the following objective function:

$$\operatorname{argmax}_{\Delta X^t \subseteq X^t} \sum_{x \in \Delta X^t} D(x, G(x))$$

where:

- $(X^t, G(X^t))$ denote all pseudo labeled instances at the t -th iteration;
- $|\Delta X^t| = \gamma$;

- Then, this subset of pseudo labeled instances $(\Delta X^t, \hat{Y})$ is propagated into the set of labeled instances $(X^*, Y)^t$ to train D .

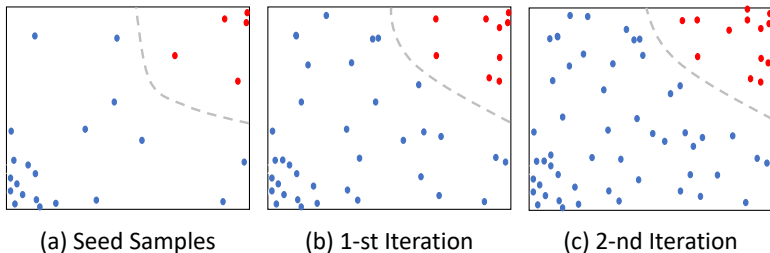


Figure: An overview of label propagation

- The objective function of D at the t -th iteration of propagation is defined as:

$$\mathcal{L}_D = \max_D \mathbb{E}_{x \sim p(X_i^U)} \log[(1 - D(x, G(x)))] + \lambda \mathbb{E}_{(x,y) \sim (X^*, Y)^t} \log[D(x, y)] \quad (2)$$

where:

- λ refers to a weighted term.
- $(X^*, Y)^t$ refers to the labeled instances in t -th iteration.

- The number of subspaces b is decided based on the number of attributes in each dataset.
- n is a hyper-parameter referring to the number of iterations for converging G and D .
- Propagation iterations t is decided by the total number X^U of unlabeled instances and the number γ of instances being propagated in each iteration, i.e. $t = \lceil \frac{|X^U|}{\gamma} \rceil$.

Dataset	#Attributes ($ A $)	#Instances ($ X $)	Imbalance Rate	#Subspaces (b)
Cora	4	837,865	1:49	16
DBLP- ACM	4/4	6,001,104	1:2,698	16
DBLP- Scholar	4/4	168,112,008	1:71,233	16
NCVoter	18/18	1,000,000	1:4,202	64

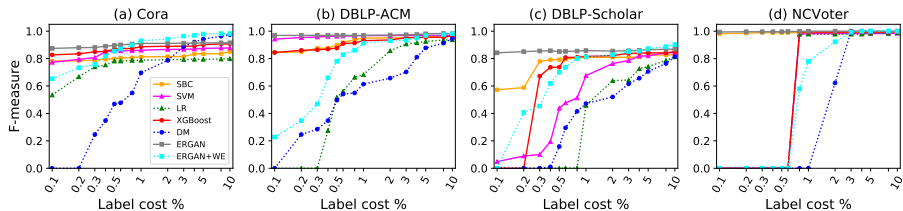
- *Unsupervised methods*: **Two-Steps** and **Iterative Term-Entity Ranking and CliqueRank (ITER-CR)**
- *Semi-supervised methods*: **Semi-supervised Boosted Classifier (SBC)**
- *Fully supervised methods*: **Magellan** and **eXtreme Gradient boosting (XGboost)**
- *Deep Learning based methods*: **DeepMatcher (DM)** and **Deep Transfer active learning (DTAL)**.

Some variants of **ErGAN** used in the ablation study:

- **ErGAN+WE** refers to the model of ERGAN augmented with word embeddings for attribute values.
- **ErGAN-D** refers to a model being obtained by removing the diversity module from ERGAN
- **ErGAN-P** refers to a model being obtained by removing the propagation module from ERGAN
- **ErNN** refers to a model whose GANs architecture is replaced by a single multi-layer perceptron.

Method	Datasets			
	Cora	DBLP- ACM	DBLP- Scholar	NCVoter
2S	62.69	91.43	68.78	98.96
ITER-CR*	89.00	–	–	–
SBC	85.71	97.09	85.47	99.78
SVM	88.95	97.19	85.71	98.48
LR	80.25	95.56	83.84	99.37
XGBoost	91.34	97.20	86.63	100
ERGAN	93.03	98.23	88.32	100
DM	98.58	98.29	94.68	100
DTAL*	98.68 \pm 0.26	98.45 \pm 0.22	92.94 \pm 0.47	–
ERGAN+WE	98.72 \pm 0.15	98.51 \pm 0.23	94.73 \pm 0.35	100

Results: 0.1% - 10% Training



Datasets	Cora				DBLP-ACM			
	0.1%	1%	20%	60%	0.1%	1%	20%	60%
ERNN	84.46	90.67	91.43	92.78	88.05	95.68	98.20	98.22
ERGAN-D	79.87	85.14	91.27	92.97	0	93.30	97.16	98.21
ERGAN-P	85.18	90.76	91.42	93.03	92.67	95.96	98.21	98.23
ERGAN	87.45	91.07	91.54	93.03	96.89	96.93	98.22	98.23
Datasets	DBLP-Scholar				NCVoter			
	0.1%	1%	20%	60%	0.1%	1%	20%	60%
ERNN	82.76	83.17	86.71	87.73	99.39	100	100	100
ERGAN-D	0	78.85	83.43	88.29	0	99.58	100	100
ERGAN-P	83.43	85.34	86.55	88.32	99.39	99.79	100	100
ERGAN	84.23	85.85	86.86	88.32	99.45	100	100	100

- We have proposed a novel method, called ERGAN, to solve the ER classification problem with very limited labeled instances.
- ERGAN incorporates the diversity of instances into sampling, prior to training the models. ERGAN consists of a label generator G to generate pseudo labels for unlabeled instances, and a discriminator D to distinguish instances with pseudo labels from instances with real labels.
- This method can be extended with word embedding for handling attribute values, leading to an enhanced method, called ERGAN+WE.
- Our experimental results show that the performance of our methods beats all the baselines.

Thank You!

Q & A

Email: Jingyu.shao@anu.edu.au