IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models

Jae Yong Kim

Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017.(SIGIR '17)

Information Retrieval(IR): Generative vs Discriminative

q: query **d**: document **r**: relevance

Generative Models: q -> d

- describing how a document is generated from a given information needed
- predicting relevant documents given a query

Discriminative (classification) Models: q+ d -> r

- learned from labelled relevant judgements or their proxies such as clicks or ratings
- considers documents and queries jointly as features and predicts their relevance from a large amount of training data
- Pointwise: approximate relevance estimation of each document to the human rating
- Pairwise: most-relevant document from any document pair
- Listwise: optimize the loss function defined over the whole ranking list for each query

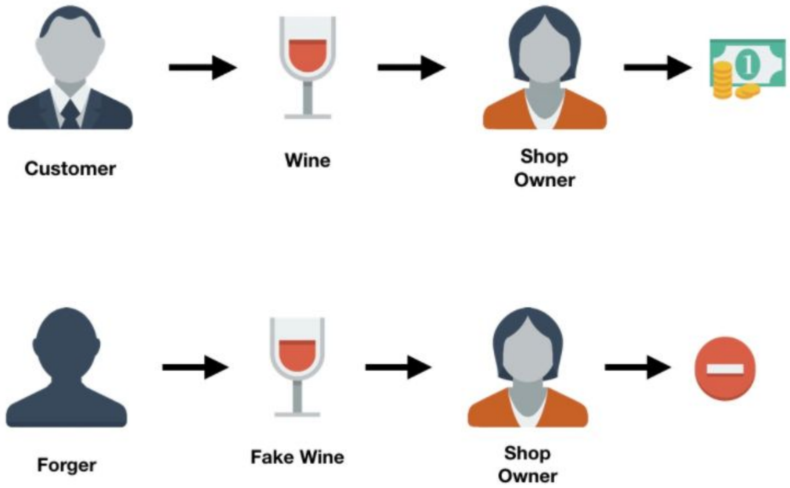
IR + GAN = IRGAN!

GAN(Generative Adversarial Networks)

- Customer sells shop owner wine, but may try to sell fake wine

- Forgers may make mistakes to forge fake wine and shop owners easily catch fake wine so they have to improve their forging skills

- Shop owners have to learn how to discriminate between fake and good wine

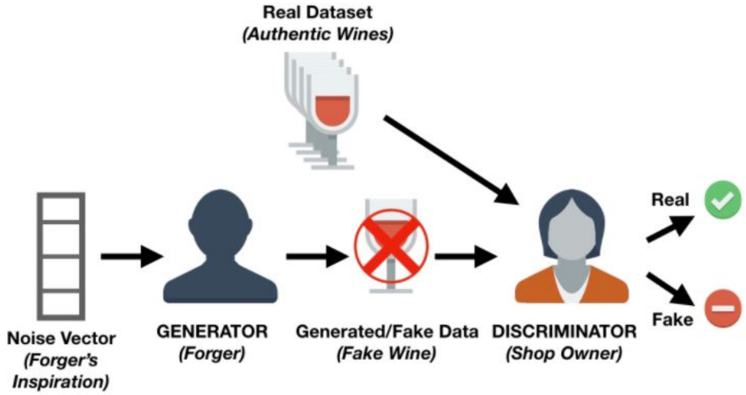


GAN(Generative Adversarial Networks)

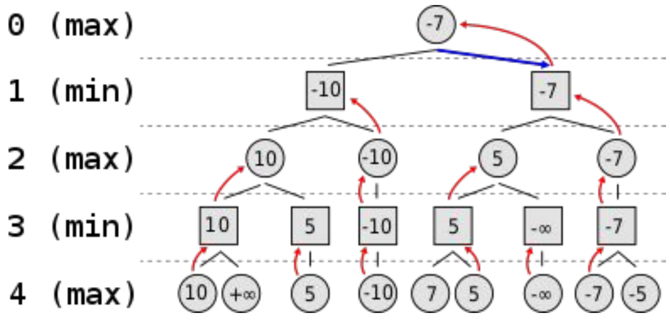
- **Generator(forger) vs Discriminator (shop owner)

- Generative network constantly creates data that resembles the ground truth

- Discriminative network tries to distinguish between fake data and real data from a real dataset



Minimax (Objective)



- Maximize benefits, minimize costs
- Objective Function:

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\text{true}}(d|q_n,r)} \left[\log D(d|q_n)\right] + \mathbb{E}_{d \sim p_{\theta}(d|q_n,r)} \left[\log(1 - D(d|q_n))\right] \Big)$$

- Discriminative Retrieval => maximize log-likelihood of correctly distinguishing the true and generated relevant documents

$$f_{\phi}(q, d) \qquad D(d|q) = \sigma(f_{\phi}(d,q)) = \frac{\exp(f_{\phi}(d,q))}{1 + \exp(f_{\phi}(d,q))}.$$

- Generative Retrieval => minimize the objective

$$p_{\theta}(d|q,r), \quad p_{\theta}(d_k|q,r) = \frac{\exp(g_{\theta}(q,d_k))}{\sum_d \exp(g_{\theta}(q,d))}$$

$g_{\theta}(q, d)$: chance of d being generated from q

Optimizing Retrievals

- Discriminative Retrieval => typically solved by stochastic gradient descent

$f_\phi(q, d)$

$$\phi^* = \arg\max_\phi \sum_{n=1}^{N} \Big(\mathbb{E}_{d\sim p_{\text{true}}(d|q_n,r)} \left[\log(\sigma(f_\phi(d, q_n))\right] + \mathbb{E}_{d\sim p_{\theta^*}(d|q_n,r)} \left[\log(1 - \sigma(f_\phi(d, q_n)))\right] \Big)$$

- Generative Retrieval => Use Policy gradient based reinforcement learning due to discrete sampling of documents *d*

$p_\theta(d|q, r)$,

$$\theta^* = \arg\min_\theta \sum_{n=1}^{N} \Big(\mathbb{E}_{d\sim p_{\text{true}}(d|q_n,r)} \left[\log \sigma(f_\phi(d, q_n))\right] + \mathbb{E}_{d\sim p_{\theta}(d|q_n,r)} \left[\log(1 - \sigma(f_\phi(d, q_n)))\right] \Big)$$
$$= \arg\max_\theta \sum_{n=1}^{N} \underbrace{\mathbb{E}_{d\sim p_{\theta}(d|q_n,r)} \left[\log(1 + \exp(f_\phi(d, q_n)))\right]}_{\text{denoted as } J^G(q_n)},$$

$$\begin{aligned} \nabla_\theta J^G(q_n) &= \nabla_\theta \mathbb{E}_{d\sim p_\theta(d|q_n,r)} \left[\log \sigma(f_\phi(d, q_n)) \right] \\ &= \sum_{i=1}^{M} \nabla_\theta p_\theta(d_i|q_n, r) \log \sigma(f_\phi(d, q_n)) \\ &= \sum_{i=1}^{M} p_\theta(d_i|q_n, r) \nabla_\theta \log p_\theta(d_i|q_n, r) \log \sigma(f_\phi(d, q_n)) \\ &= \mathbb{E}_{d\sim p_\theta(d|q_n,r)} \left[\nabla_\theta \log p_\theta(d|q_n, r) \log \sigma(f_\phi(d, q_n)) \right] \\ &\simeq \frac{1}{K} \sum_{k=1}^{K} \nabla_\theta \log p_\theta(d_k|q_n, r) \log \sigma(f_\phi(d, q_n)) . \end{aligned} \tag{22}$$

Extension to Pairwise

- Common that labelled training data may be a set of ordered document pairs for each query instead of a set of relevant documents

$$R_n = \{\langle d_i, d_j \rangle | d_i > d_j\}$$

- Discriminative Retrieval:

$$D(\langle d_u, d_v \rangle | q) = \sigma(f_\phi(d_u, q) - f_\phi(d_v, q))$$
$$= \frac{\exp(f_\phi(d_u, q) - f_\phi(d_v, q))}{1 + \exp(f_\phi(d_u, q) - f_\phi(d_v, q))} = \frac{1}{1 + \exp(-z)}$$

$$z = f_\phi(d_u, q) - f_\phi(d_v, q)$$

- Generative Retrieval:

$$G(\langle d_k, d_j \rangle | q) = p_\theta(\mathbf{o}' | q) = \frac{\exp\left(g_\theta(d_k, q) - g_\theta(d_j, q)\right)}{\sum_d \exp\left(g_\theta(d, q) - g_\theta(d_j, q)\right)}$$

** generative and discriminative retrieval models may reach different performances depending on the specific task

Applications

The implementation of two scoring function $f_\phi(q,d)$ $g_\theta(q,d)$ are both task-specific.

(i.e. discriminator scoring function could be implemented as a 3-layer neural work while generator scoring function could be a factorization machine)

$$g_\theta(q,d) = s_\theta(q,d) \quad \text{and} \quad f_\phi(q,d) = s_\phi(q,d)$$

1. Web Search $s(q,d) = \mathbf{w}_2^\top \tanh(\mathbf{W}_1 \mathbf{x}_{q,d} + \mathbf{b}_1) + w_0$
    - each query-document pair can be represented by a vector, where each dimension represents some statistical value of the pair
2. Item Recommendation $s(u,i) = b_i + \boldsymbol{v}_u^\top \boldsymbol{v}_i$
    - Matrix factorization
3. Question Answering
    - q: questions, a: answer (represented as a sequence of words)
    - relevance score can be defined as their cosine similarity

$$s(q,a) = \cos(\boldsymbol{v}_q, \boldsymbol{v}_a) = \frac{\boldsymbol{v}_q^\top \boldsymbol{v}_a}{|\boldsymbol{v}_q| \cdot |\boldsymbol{v}_a|}$$

Experiments: Web Search

- dataset: LETOR(LEarning TO Rank)
- Less labelled data, more unlabelled data
- relevance level: (-1, 0, 1, 2) but (-1 = “unknown)

Table 1: Webpage ranking performance comparison on the MQ2008-semi dataset, where ∗ means a significant improvement according to the Wilcoxon signed-rank test.

	P@3	P@5	P@10	MAP
MLE	0.1556	0.1295	0.1029	0.1604
RankNet [3]	0.1619	0.1219	0.1010	0.1517
LambdaRank [5]	0.1651	0.1352	0.1076	0.1658
LambdaMART [4]	0.1368	0.1026	0.0846	0.1288
IRGAN-pointwise	0.1714	0.1657	**0.1257**	**0.1915**
IRGAN-pairwise	**0.2000**	**0.1676**	0.1248	0.1816
Impv-pointwise	3.82%	22.56%*	16.82%*	15.50%*
Impv-pairwise	21.14%*	23.96%*	15.98%	9.53%
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.1893	0.1854	0.2054	0.3194
RankNet [3]	0.1801	0.1709	0.1943	0.3062
LambdaRank [5]	0.1926	0.1920	0.2093	0.3242
LambdaMART [4]	0.1573	0.1456	0.1627	0.2696
IRGAN-pointwise	0.2065	**0.2225**	**0.2483**	**0.3508**
IRGAN-pairwise	**0.2148**	0.2154	0.2380	0.3322
Impv-pointwise	7.22%	15.89%	18.63%	8.20%
Impv-pairwise	11.53%	12.19%	13.71%	2.47%

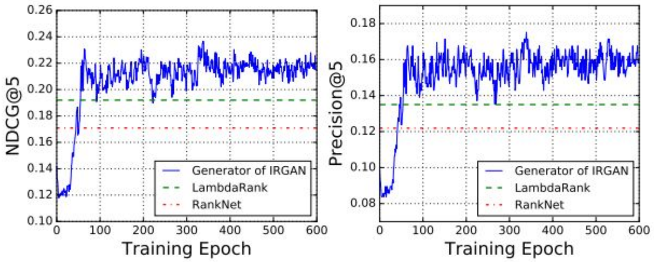


Figure 2: Learning curves of the pointwise IRGAN on the web search task.

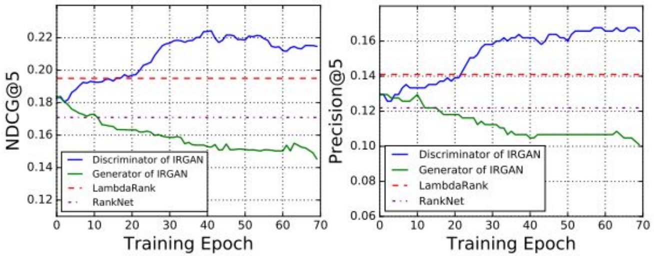


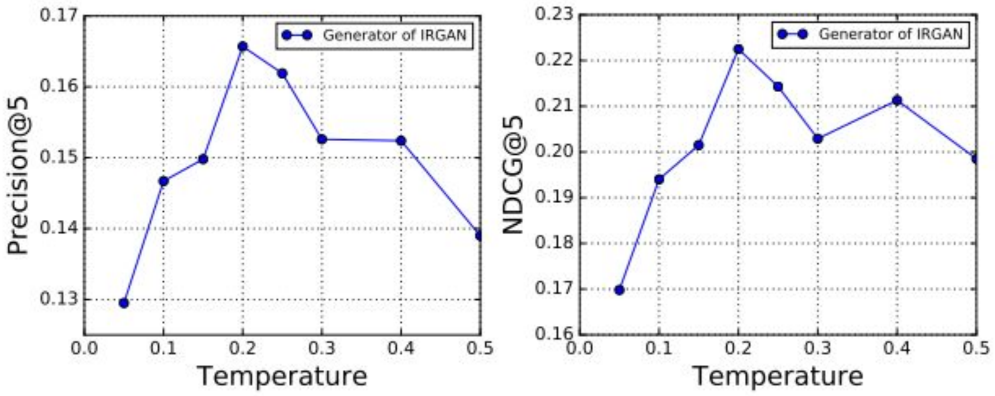
Figure 3: Learning curves of the pairwise IRGAN on the web search task.

Experiments: Web Search

- In the sampling stage, the temperature parameter is incorporated as ,

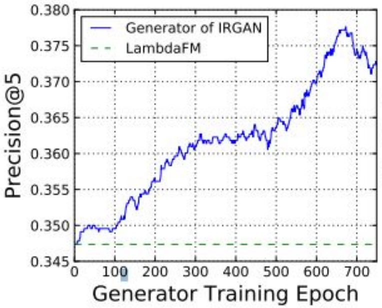
$$p_\theta(d|q,r) = \frac{\exp\left(g_\theta(q,d)/\tau\right)}{\sum_{j\in\mathcal{I}} \exp\left(g_\theta(q,d)/\tau\right)}$$

** low temperature => sampling focuses more on top-ranked documents



Experiments: Item Recommendations

- dataset: Movielens and Netflix
- 5 - star ratings as positive feedback and all others as unknown feedback
- matrix factorization



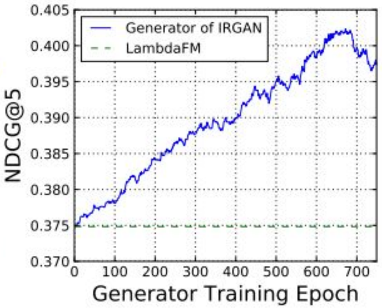


Table 3: Item recommendation results (Movielens).

	P@3	P@5	P@10	MAP
MLE	0.3369	0.3013	0.2559	0.2005
BPR [34]	0.3289	0.3044	0.2656	0.2009
LambdaFM [46]	0.3845	0.3474	0.2967	0.2222
IRGAN-pointwise	0.4072	0.3750	0.3140	0.2418
Impv-pointwise	5.90%*	7.94%*	5.83%*	8.82%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3461	0.3236	0.3017	0.5264
BPR [34]	0.3410	0.3245	0.3076	0.5290
LambdaFM [46]	0.3986	0.3749	0.3518	0.5797
IRGAN-pointwise	0.4222	0.4009	0.3723	0.6082
Impv-pointwise	5.92%*	6.94%*	5.83%*	4.92%*

Table 4: Item recommendation results (Netflix).

	P@3	P@5	P@10	MAP
MLE	0.2941	0.2945	0.2777	0.0957
BPR [34]	0.3040	0.2933	0.2774	0.0935
LambdaFM [46]	0.3901	0.3790	0.3489	0.1672
IRGAN-pointwise	0.4456	0.4335	0.3923	0.1720
Impv-pointwise	14.23%*	14.38%*	12.44%*	2.87%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3032	0.3011	0.2878	0.5085
BPR [34]	0.3077	0.2993	0.2866	0.5040
LambdaFM [46]	0.3942	0.3854	0.3624	0.5857
IRGAN-pointwise	0.4498	0.4404	0.4097	0.6371
Impv-pointwise	14.10%*	14.27%*	13.05%*	8.78%*

Experiments: Question Answering

- dataset: InsuranceQA
- 12,887 questions with correct answers
- 1000 unseen question-answer pairs in development set
- 1800 pairs for 2 sets

Table 5: The Precision@1 of InsuranceQA.

	test-1	test-2
QA-CNN [9]	0.6133	0.5689
LambdaCNN [9, 51]	0.6294	0.6006
IRGAN-pairwise	0.6444	0.6111
Impv-pairwise	2.38%*	1.75%

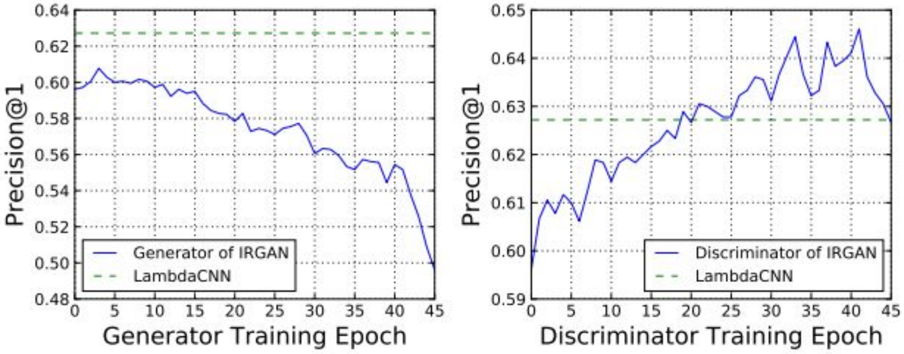


Figure 8: The experimental results in QA task.

Conclusion

1. Adversarial training of generative retrieval models and discriminative retrieval models show improvement in Information Retrieval in different applications such as web search, item recommendation, and question answering

2. Pairwise vs Pointwise shows different results
    a. Pointwise: generative retrieval > discriminative retrieval
    b. Pairwise: generative retrieval < discriminative retrieval