Information Retrieval GAN for RS:

Jun Wang*, Lanton Yu*, Yu Gong**

University College London*, Alibaba Group**

Anup Deshmukh

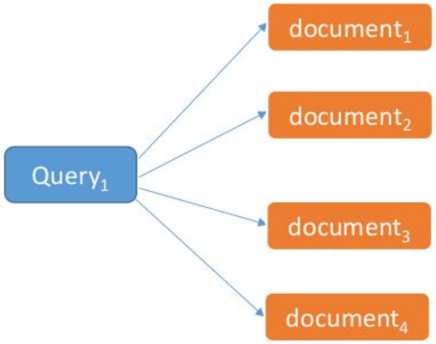
Sections

- Information Retrieval (IR) models
- **Formalizing the minimax game for IR!**
    - **Difficulties and Solutions**
- Exploiting IRGAN's in Recommender systems

1.1

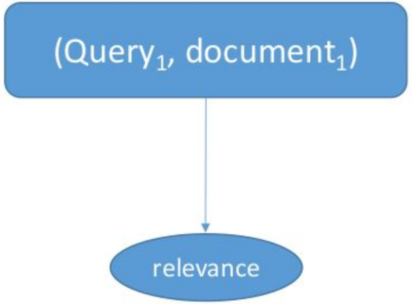
Two schools of thinking in IR modeling

Generative Retrieval



- Assume there is an underlying stochastic generative process between documents and queries
- Generate/Select relevant documents given a query

Discriminative Retrieval



- Learns from labeled relevant judgments
- Predict the relevance given a query-document pair

1.2 Borrowing the idea from GAN's

- Take advantage of both schools of thinking
  - The **generative model** learns to fit the relevance distribution over documents via the signal from the discriminative model.
  - The **discriminative model** should be able to exploit the unlabeled data selected by the generative model to achieve a better estimation for document ranking.

2.1 IRGAN: formulation

- r: relevance of the document, q: queries, d: corresponding document
- $\mathbf{p}_{\mathbf{true}}(\mathbf{d}\,|\,\mathbf{q}, \mathbf{r})$: relevance preferance distribution
- Generative retrieval model $\mathbf{p}_{\theta}(\mathbf{d}\,|\,\mathbf{q}, \mathbf{r})$, which tries to generate **(or select)** relevant documents, from the candidate pool for the given query q.
- Discriminative retrieval model $\mathbf{f}_{\phi}(\mathbf{q}, \mathbf{d})$: goal is to distinguish between relevant documents and non-relevant documents for the query q as accurately as possible.

2.2 IRGAN: overall objective

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

$$\mathbb{E}_{d \sim p_{\theta}(d|q_n,r)} \left[\log(1 - D(d|q_n)) \right] \Big)$$

where $D(d|q) = \sigma(f_{\phi}(d,q)) = \dfrac{\exp(f_{\phi}(d,q))}{1 + \exp(f_{\phi}(d,q))}$

2.3 IRGAN: optimizing discriminative retrieval

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

2.4 IRGAN: optimizing generative retrieval

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

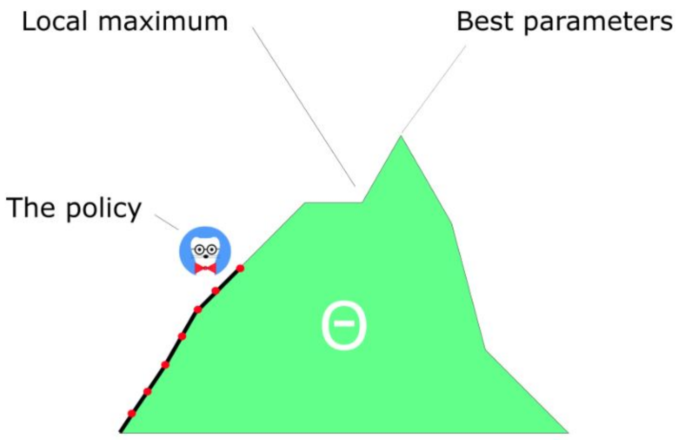
2.5 IRGAN: moving away from GAN's

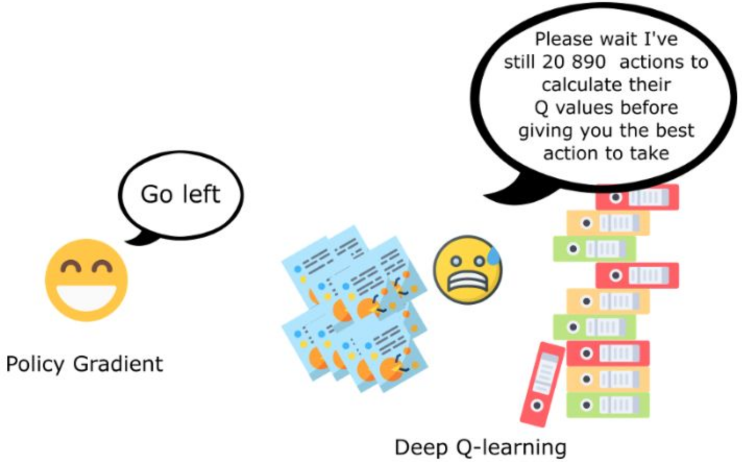
- First, the generative retrieval process is **stochastic sampling over discrete data**, i.e., the candidate documents, which is different from the deterministic generation based on the sampled noise signal in the original GAN.
- As the sampling of d is **discrete**, it cannot be directly optimised by gradient descent as in the original GAN formulation.

2.6 Q-learning in RL

1. Initialize Q-values ($Q(s,a)$) arbitrarily for all state-action pairs.
2. For life or until learning is stopped...
3. Choose an action (a) in the current world state (s) based on current Q-value estimates ($Q(s,\cdot)$).
4. Take the action (a) and observe the the outcome state (s') and reward (r).
5. Update $Q(s,a) := Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$

2.7 Policy gradients over Q-learning





2.8 Policy gradients

$$\pi_\theta(a|s) = P[a|s]$$

Probability of taking action a given state s with parameters theta.

$$J_{avgv}(\theta) = E_\pi(V(s)) = \sum d(s)V(s)$$

$$where\ \ d(s) = \frac{\boxed{N(s)}}{\boxed{\sum_{s'} N(s')}}$$

Number of occurrences of the state

Total nb occurences of all states

$$J_{avR}(\theta) = E_\pi(r) = \underbrace{\sum_s d(s)}_{\text{Probability that I'm in state s}} \underbrace{\sum_a \pi\theta(s,a)}_{\text{Probability that I take this action a from that state under this policy}} R_s^a$$

Immediate rewar I'll get

$$Policy:\ \pi_\theta$$

$$Objective\ function\ :J(\theta)$$

$$Gradient:\ \nabla_\theta J(\theta)$$

$$Update:\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta)$$

2.9 IRGAN: optimizing generative retrieval

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

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

3.1 IRGAN

Algorithm 1 Minimax Game for IR (a.k.a IRGAN)

Input: generator $p_\theta(d|q, r)$; discriminator $f_\phi(\mathbf{x}_i^q)$;
training dataset $\mathcal{S} = \{\mathbf{x}\}$

Initialise $p_\theta(d|q, r)$, $f_\phi(q, d)$ with random weights θ, ϕ.
Pre-train $p_\theta(d|q, r)$, $f_\phi(q, d)$ using $\mathcal{S}$
repeat
    for g-steps **do**
        $p_\theta(d|q, r)$ generates K documents for each query q
        Update generator parameters via policy gradient Eq. (5)
    end for
    for d-steps **do**
        Use current $p_\theta(d|q, r)$ to generate negative examples and combine with given positive examples $\mathcal{S}$
        Train discriminator $f_\phi(q, d)$ by Eq. (3)
    end for
until IRGAN converges

3.1 Recommender Systems

$$\text{recommender system precision:} \quad P = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of items we recommended}}$$

$$\text{recommender system recall:} \quad r = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of all the possible relevant items}}$$

Table 3: Item recommendation results (Movielens).

	P@3	P@5	P@10	MAP
MLE	0.3369	0.3013	0.2559	0.2005
BPR [35]	0.3289	0.3044	0.2656	0.2009
LambdaFM [45]	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 [35]	0.3410	0.3245	0.3076	0.5290
LambdaFM [45]	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 [35]	0.3040	0.2933	0.2774	0.0935
LambdaFM [45]	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 [35]	0.3077	0.2993	0.2866	0.5040
LambdaFM [45]	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%*

3.1 Recommender Systems

- How?
  - Maybe in next presentation


Thank
you.