Deep Reinforcement Learning in Intelligent Finance

Xiong Jun Wu@Ant Group

2018.08.28

ANT FINANCIAL

蚂蚁金融服务集团 版权所有

Outline

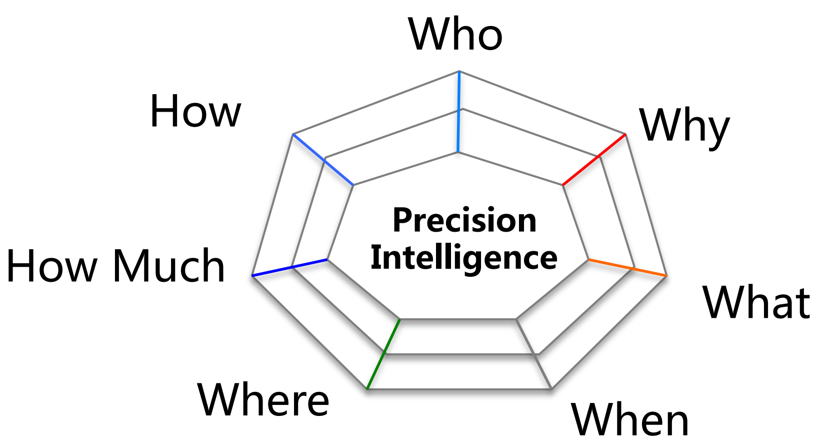
- Background
- DRL for intelligent finance decision making
  - Part 1: A practical example in credit consumer finance
    - DRL for ant credit intelligent finance marketing
  - Part 2: Recent work in DRL modeling
    - A policy gradient method for uplift modeling
- Ongoing and future work
- Q&A

Background

- Ant Financial’ s ecosystem
  - Provides various financial products
    - Ant Credit: credit pay / consumption credit
    - Cash Now/ Small and Micro Business Loan: credit loan for personal/small business
    - ……
  - Hundreds of millions users
    - Current users, potentials and inactive ones
  - How to target individual needs of users in the financial ecosystem?

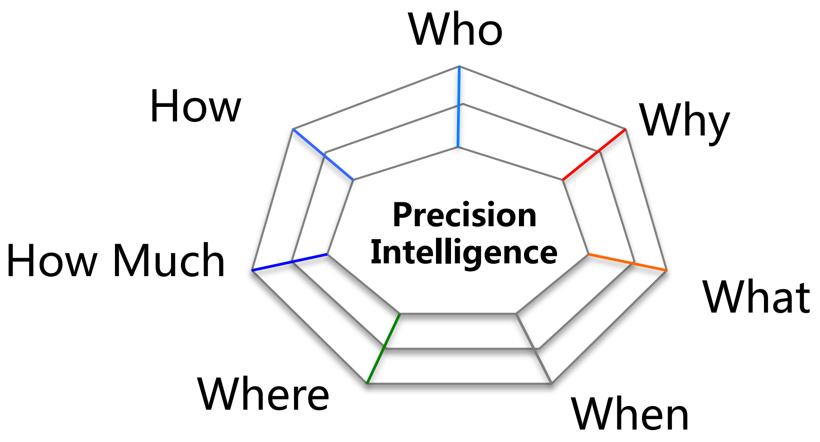
Main challenges for intelligent finance decision making

- Different customers with different needs(Who&Why)
  - Wealth status, demographics, behavioral economics
  - Different periods of their life
  - Aesthetic fatigue, behavioral psychology
- Financial products(What)
  - Simple function VS. business flow complexity
- User’ s environments(Where&When)
  - Partially observed or unknown
  - Random, multi-dimensional and dynamic
  - Immediate and intelligent decision making



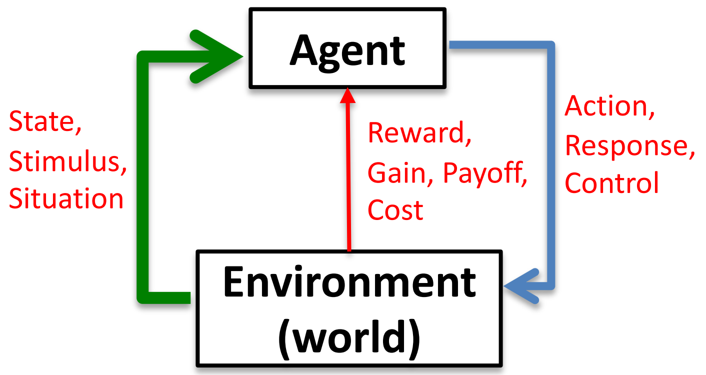
Main challenges for intelligent finance decision making

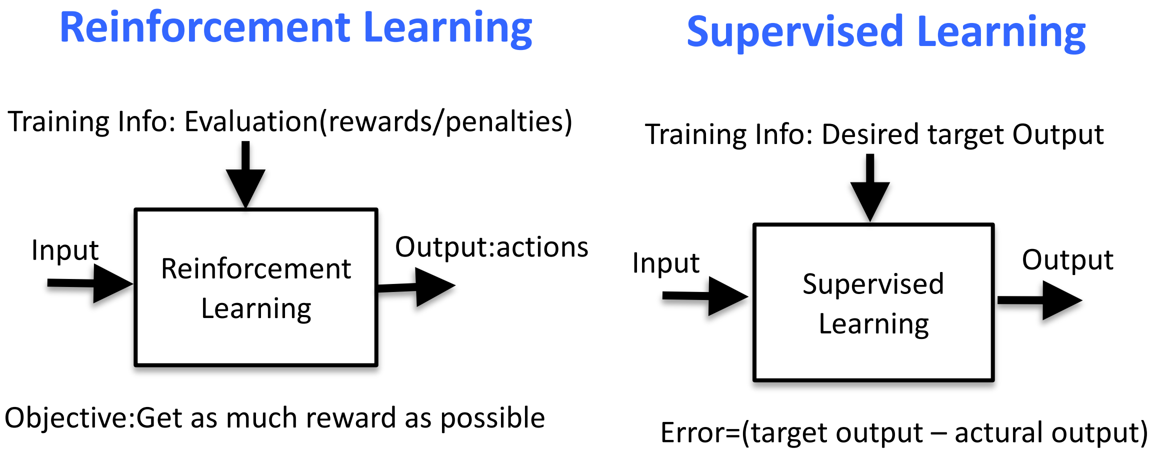
- Diversified forms of benefits for users(How much)
  - Discounted rate/price, red pocket, coupon, cash back
- Marketing budget(How much)
  - ROI: macro control and micro optimization
- Channel for different consumer finance scenarios(How)
  - Customer activity and scene targeting
  - Channel matching:Message, SMS, phone etc.
- Marketing cycle(How)
  - Frequency period, time decay, superposition, mutual exclusion



How to make intelligent decision in finance under complex and dynamic environment?

- Reinforcement Learning(RL) VS Supervised Learning(SL)
  - RL learning from interactions—Agent learns a policy mapping states to actions
    - Impractical to obtain examples of desired behavior that are both correct and representative of all the situations
    - Trade-off between exploration and exploitation
    - Delayed reward
    - Learn from its own experience
  - SL learning from examples
    - Provided by a knowledgeable external supervisor

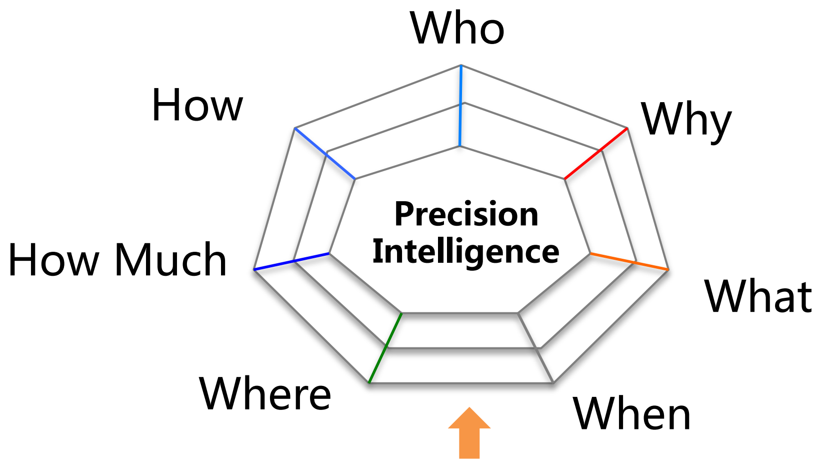


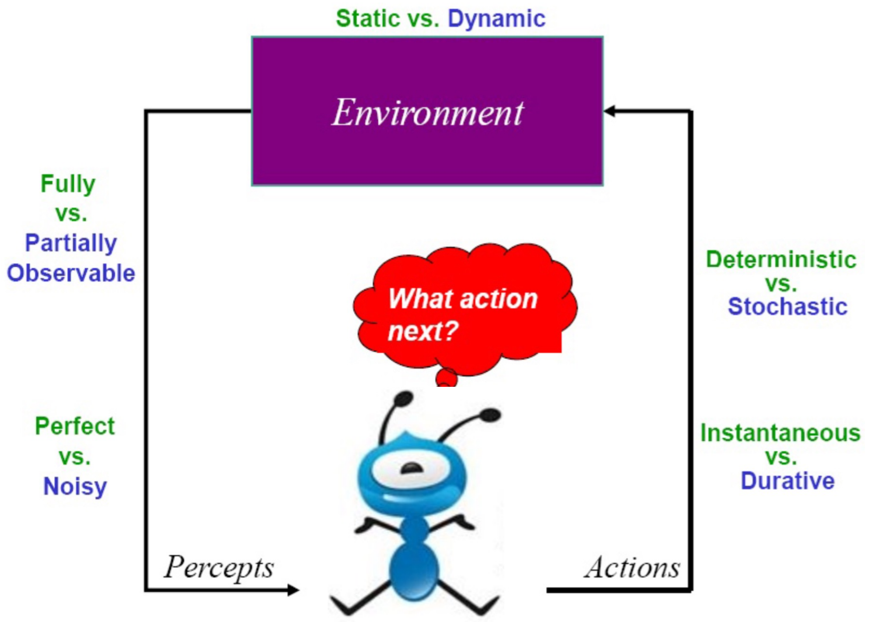


Reinforcement Learning VS Supervised Learning

How to make intelligent decision in finance under complex and dynamic environment?

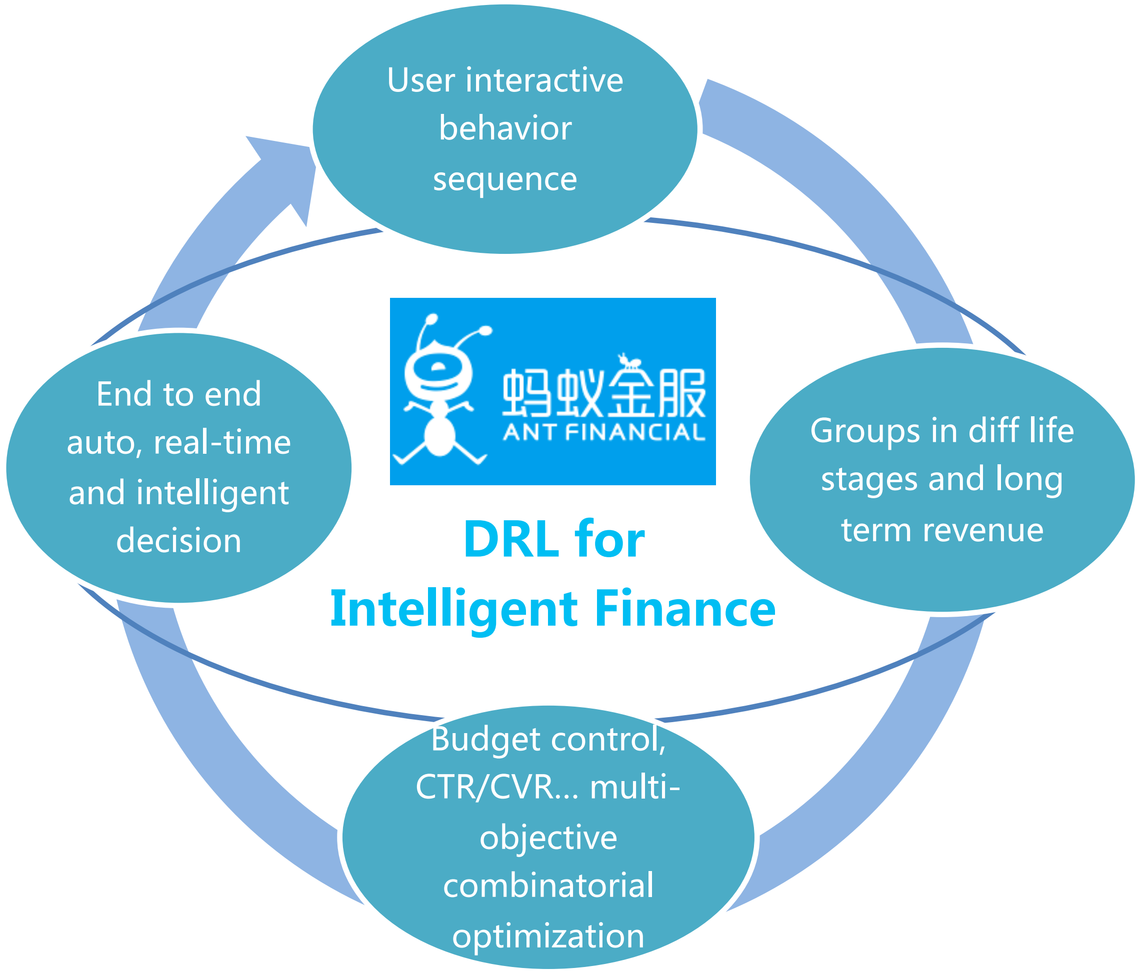
- RL seems to provide a very promising solution framework
  - A general purpose intelligent framework
  - Explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment
  - Seeking to maximize its cumulative reward in the long run
- RL with deep learning or DRL
  - Apply deep learning to RL
    - Use deep neural network approximation to opt value function/policy/model end-to-end





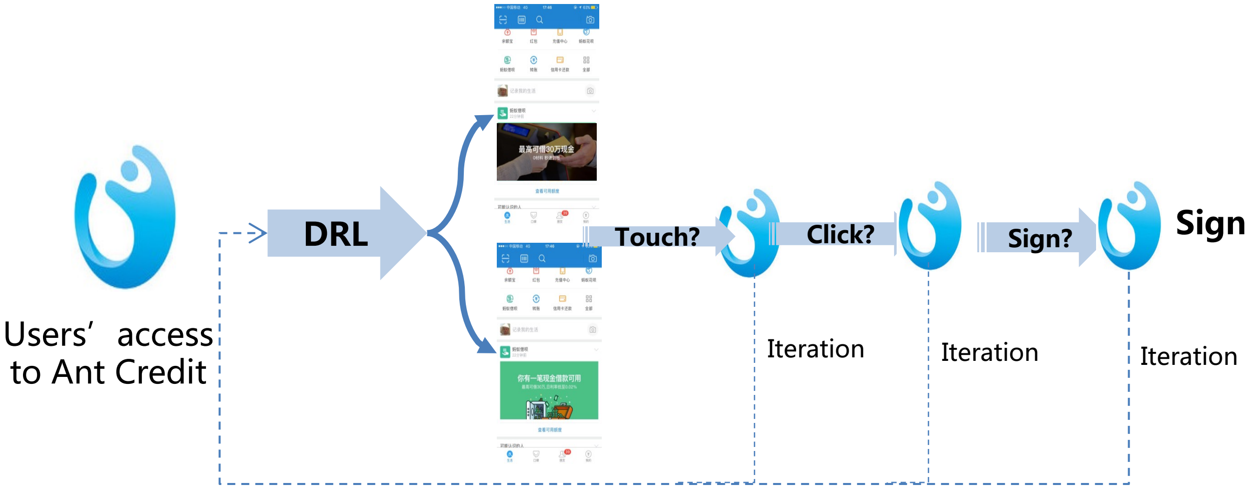
DRL for intelligent finance decision making

- Interactive and sequence decision learning
  - Interactive behavior sequences
- Long term revenue
  - Financial business often targets long-term revenues
  - Different groups in different life stages
- Multi-objective decision making
  - Precise timing, scene orientation
  - Channel matching, different ways of reach, various benefits
  - CTR/CVR, ROI budget constraints etc.
- End-to-end decision
  - A unified, automatic and real-time intelligent decision-making service

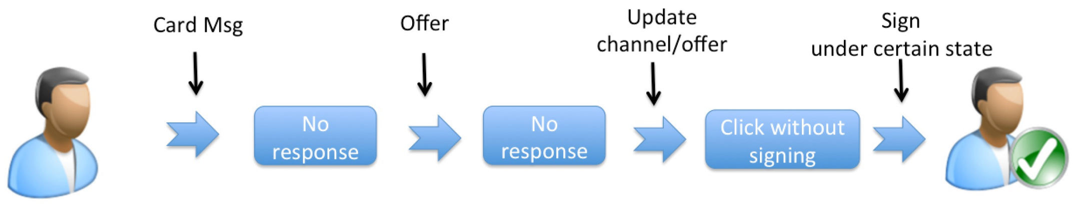


DRL for Ant Credit intelligent finance marketing

- **Context**
  - Start points of life cycle marketing
  - Key factors of GMV and profits
  - Most of the active users have converted, the others very difficult to convert
- **Goal**
  - Through different marketing activities repeatedly touch, change marketing strategy to reach sign target according to users' behavior feedback



- **DRL model design**
  - Repeated touch sequences for reinforcing decision, each marketing activity as a episode, N days for a delivery cycle



  - Actor-Critic Deep RL

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi\theta}[\nabla_\theta log \pi_\theta(s,a) A^{\pi\theta}(s,a)]$$

here $A^{\pi\theta}(s,a) = Q^{\pi\theta}(s,a) - V^{\pi\theta}(s)$

  - State: features from multiple business
  - Action: card x channel for compounded decisions
  - Reward: combined click with signs etc.

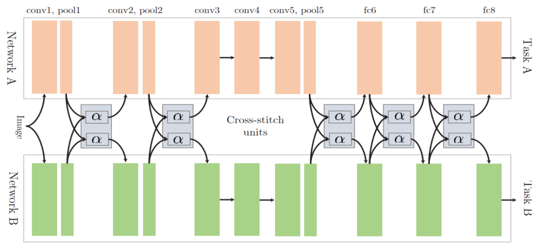
DRL ABTest experiments design

- The problem was formulated as a classification problem
  - Sign and click object and separately build two models
  - Given an user, the models predict the action that can make the user sign or click with max probability
- Performance among DRL， MTL methods and single DNN method were compared，especially for DRL with multi-task/multi-View/multi-Object supervised learning
  - Tensor Factorization for MTL through tensor trace norm[1] and Cross-Stitch MTL[2] methods were choosed
  - Tensor Trace Norm MTL
  - Cross Stich MTL

(Tensor Trace Norm) Tucker $||\mathcal{W}||_* = \sum_{i=1}^{N} \gamma_i ||\mathcal{W}_{(i)}||_*$

(Tensor Trace Norm) TT $||\mathcal{W}||_* = \sum_{i=1}^{N-1} \gamma_i ||\mathcal{W}_{[i]}||_*$

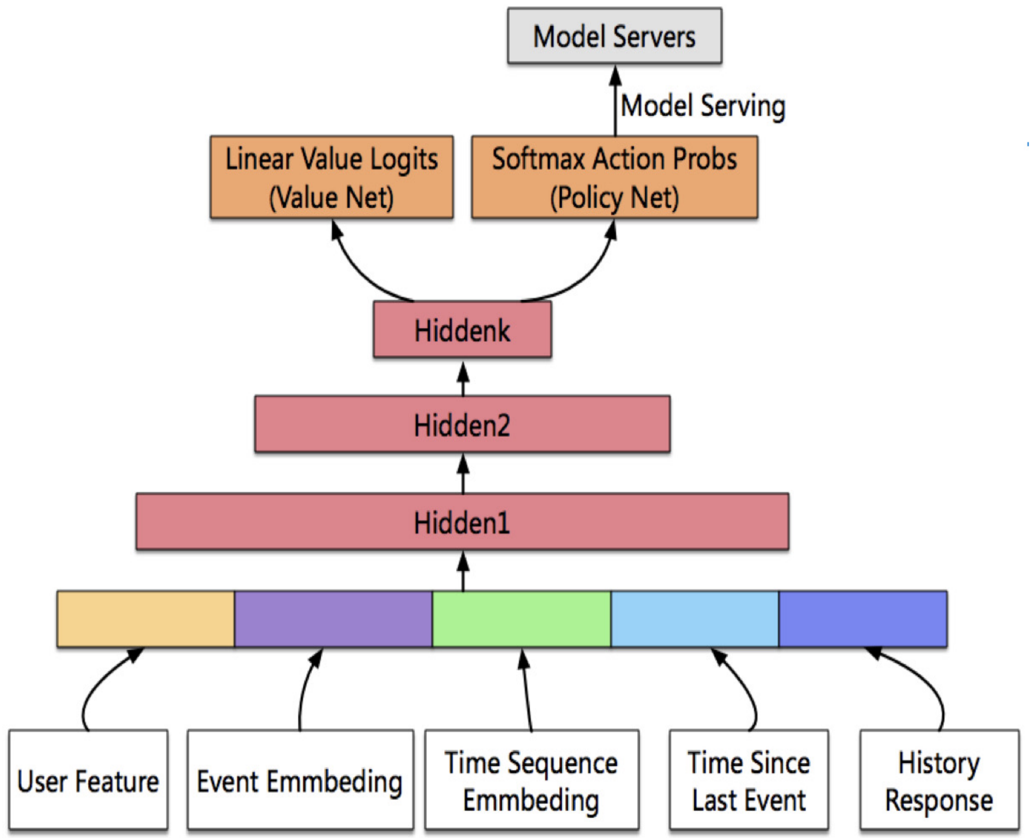
(Tensor Trace Norm) Last Axis Flattening $||\mathcal{W}||_* = \gamma ||\mathcal{W}_{(N)}||_*$



[1]Yang Y, Hospedales T M. Trace Norm Regularised Deep Multi-Task Learning[J]. 2017 ICLR
[2] Misra I, Shrivastava A, Gupta A, et al. Cross-stitch networks for multi-task learning[C], CVPR 2016

DRL ABTest experiments design

- DRL model settings
  - Discount factor = 0.99
  - The policy network is a classification network with 3 hidden layers:
    - The number of each layer: [256,256,256]
    - Activation function: tanh
    - Learning rate: 0.00025
    - Loss function: cross-entropy
  - The value networks is a regression network with 3 hidden layers:
    - The number of each layer: [256,256,256]
    - Activation function: tanh
    - Learning rate: 0.00025
    - Loss function: squared difference



DRL ABTest experiments design

- Trace norm MTL(Fig.1)
  - $Loss = L1(X1,Y1) + L2(X2,Y2) + Loss\_trace\_norm(W)$
  - $Loss\_trace\_norm$: The multitask regularization term with tensor trace norm constraint (LAF, Tucker, TT)
  - The weight of trace norm term: 0.0005
- Cross Stitch MTL(Fig.2)
  - $Loss = L1(X1, Y1) + L2(X2, Y2)$
  - The cross-stitch unit is used to learning task relationship
- Model setting
  - Left network learns the sign model and the right network learns the click model
  - X1, X2: User's feature (880).
  - Y1, Y2: The labels of different users (6).
  - W: The parameters of the two networks.
  - $L1$: The cross-entropy loss function of the sign model.
  - $L2$: The cross-entropy loss function of the click model.
  - The number of each layer: [125,125,125]
  - Activation function: sigmoid
  - Learning rate: 0.001
  - Batch size: 100

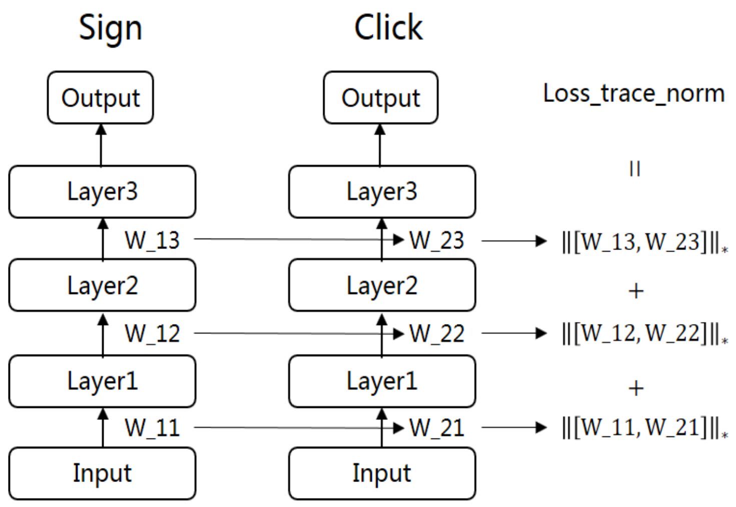


Fig. 1 Trace Norm MTL

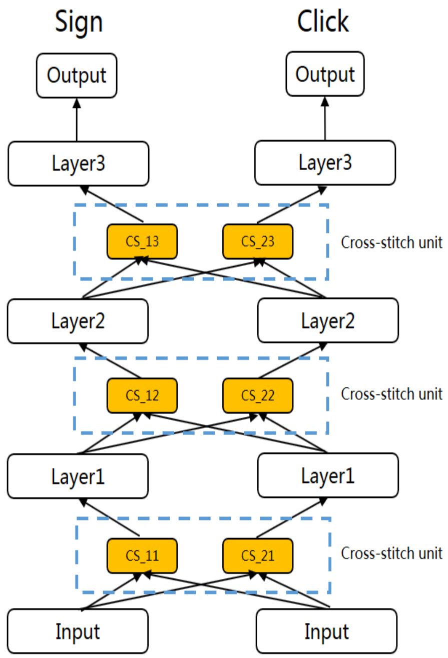


Fig. 2 Cross Stitch MTL

DRL performance evaluation

- Comparison DRL with MTL with BPI(Business Performance Index)

Methods	convRateLift	avgHitConvCost	avgAllConvCost
MTL-TN-TT	-10.53%	3.80	4.15
MTL-TN-Tucker	-15.84%	3.96	4.15
MTL-TN-LAF	-18.26%	3.92	4.15
MTL-CS-125	-18.34%	3.72	4.15
MTL-CS-256	-20.55%	3.92	4.15
MTL-CS-525	-19.10%	3.99	4.15

$$Lift_{bpi}(\pi) = \frac{ConvRate(C) - ConvRate(B)}{ConvRate(B)}$$

s.t.

$$A = \{s \in U \mid a = \pi_\theta(s)\}$$
$$B = \{s \in U \mid a = actual\_offer(s)\}$$
$$C = \{s \in U \mid a = \pi_\theta(s) \ \&\ \pi_\theta(s) = actual\_offer(s)\}$$
$$|C| \geqslant \gamma |B|, \qquad \gamma \leqslant 1$$

  - It shows that the performance DRL method better than this two type of MTL methods

A policy gradient method for uplift modeling

- Uplift problem
  - Directly model the incremental impact of a treatment on an individual response
  - Aims at maximizing the differences between offering awards to the customers or not
  - Extensively studied in traditional marketing, but received very little attention in internet financial marketing
    - Traditional classifiers predict the conditional probability

$$P^T(Y|X_1, \dots X_m)$$

    - Uplift models predict change in behavior resulting from the action

$$P^T(Y|X_1, \dots X_m) - P^C(Y|X_1, \dots X_m)$$

Uplift problem formulation

- Uplift Modeling

$Y(x,a) = B(x) + L(x,a)$

X: User' s features

a: The action provided, a = 0 means no action.

$Y(x,a)$: The observed action response when x receives action a

$B(x)$: The natural response of x when receiving no action

$L(x,a)$: The uplift response when x receives action a

Objective:

$$max_{\pi}\, E_{X,\pi}[L(X,\pi(X)]$$

The goal is to find a optimal policy π to maximize the expected uplift response.

Main Challenges for uplift modeling by reinforcement learning

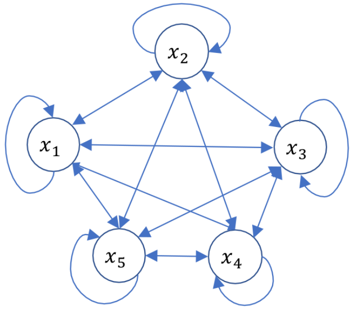
- The uplift value of a policy with the offline dataset hard to know because unobservable
  - Offline evaluation method provided
- The uplift value for each user hard to know
  - Policy gradient method dealing with delayed rewards
- Comparing with traditional direct modeling

$$Y(x,a) = B(x) + L(x,a) \quad \text{VS} \quad Y(x,a)$$

  - Algorithmic view
    - More information about the structure of data
  - Financial view
    - What truly matters is the difference between providing an action or not, especially when actions cost real money

A policy gradient method for uplift modeling

- The MDP model of uplift modeling and reward function



$$R_s^a = \begin{cases} Y(s,a), & \pi(s) = T(s) \text{ and } \pi(s) > 0 \\ -Y(s,0), & \pi(s) > 0 \text{ and } T(s) = 0 \\ 0, & others \end{cases}$$

- Q-value Estimation

$$Q^\pi(s,a) = \begin{cases} (Y(s,a) - \overline{Y^T}) + (V^\pi(s^*) - \overline{V^\pi}(s^*)), & \pi(s) = T(s) \text{ and } \pi(s) > 0 \\ (\overline{Y^C} - Y(s,0)) + (V^\pi(s^*) - \overline{V^\pi}(s^*)), & \pi(s) > 0 \text{ and } T(s) = 0 \\ 0, & others \end{cases}$$

$$\overline{Y^T} = \sum_{m=1}^M Y_m^T / M \text{ and } \overline{Y^C} = \sum_{m=1}^M Y_m^C / M$$

$\overline{V^\pi}(s^*) = \sum_{m=1}^M V_m^\pi(s^*)/M$: The average value of multiple batches

Y_m^T: The average response for actions group and Y_m^C for the control group

A policy gradient method for uplift modeling

Algorithm 1: Policy Graident Algorithm for Uplift Modeling

Input: Episode number $numEpoch$. Training data $Data$, batch size bs, learning rate α
Output: The policy network θ
for $epoch \leftarrow 1$ **to** $numEpoch$ **do**
    Sample M batches $\mathbf{\Gamma} = \{\Gamma_1, \ldots, \Gamma_M\}$ from $Data$, where each batch contains bs samples.
    foreach $\Gamma_m \in \mathbf{\Gamma}$ **do**
        $A_m = \{a_{m,1}, \ldots, a_{m,bs}\}$, where $a_{m,i} \sim \pi(s_{m,i}, \theta)$
        $V_m^\pi(s^*), \overline{Y^T}, \overline{Y^C} = UMG(\Gamma_m, A_m)$
    $\overline{V^\pi}(s^*) = \sum_{i=1}^{M} V_m^\pi(s^*)/M$
    for $m \leftarrow 1$ **to** M **do**
        Compute the $Q^\pi(s_{m,i}, a), \forall s_{m,i} \in \Gamma_m$, according to Equ.$\boxed{9}$
        $\theta \leftarrow \theta + \alpha \sum_{i=1}^{bs} \nabla_\theta \log\pi(s_{m,i}, a_{m,i}) Q^\pi(s_{m,i}, a_{m,i})$

A policy gradient method for uplift modeling

- Offline Evaluation Method-Uplift Modeling General Metric (UMG)

$$\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z^{(T,i)} - \frac{1}{N}\sum_{i=1}^{N} z^{(C,i)}$$

Where,

$$Z^T(\pi) = \textstyle\sum_{a=1}^{K} \frac{1}{p_a} Y(X,a)(\pi(X) == a)(T(X) == a)$$

$$Z^C(\pi) = \textstyle\sum_{a=1}^{K} \frac{1}{p_a} Y(X,0)(\pi(X) == a)(T(X) == 0)$$

  – An unbiased metric for accurate offline evaluation of uplift effects

RLift ABTest experiments design

- Compared Baselines
  - DRL-A3C
    - Same Markov Decision Process.
    - Reward is calculated for each sample, comparing with RLift using delayed rewards
  - DNN
    - Also known as Separate Model Approach in Uplift modeling literatures
    - Regressing the response for each couple of user's features and action first, and then choosing the action corresponding to the maximal response for each user
  - Contextual Bandit
    - The problem can be regarded as partial label problems in the field of contextual bandit
    - OffsetTree algorithm (Beygelzimer and Langford, 2009) claims a state-of-art performance
  - Random
    - All the results are compared with the one from random decision by improved percentage

RLift ABTest experiments design

- Parameter setting
  - Neural Network
    - {one, two, three} hidden layers with size of {256, 512, 1024, 2048} are considered
    - Activation function: tanh
    - Learning rate: 0.1
    - RLift Batch size: 10000
    - Maximal iterations: RLift: 200, DRL-A3C:20000000, DNN:20000000
  - Features
    - 250 related attributes, such as one’s resident, age, gender and so on
  - Samples
    - 20,000,000 samples are used for training, while 2,000,000 samples are used for evaluation

RLift performance evaluation

Model	RLift	DRL-A3C	DNN	Contextual Bandit	Random
Relative Lift	**9.0218%**	8.8134%	5.3585%	2.6724%	0

- RLift is slightly better than DRL-A3C, and it seems that they are both approaching the overall optimal policy
- The uplift signal is usually weak in real scenario, resulting a worse performance for directly modeling like DNN
- Contextual Bandit(OffsetTree) algorithm may be not suitable for big data scenario
- Besides, RLift can
  - Deal with **any number of actions (in comparison to traditional uplift modeling)**
  - Be applied to applications with **responses of general types**

Ongoing and future work

- **ROSA(Reinforcement Online Service of AI)**
  - Effective RL formulation, tuning and evaluation
    - General reward function design with reward learning
    - Industry RL Model Evaluation
      - General evaluation data set like ImageNet
      - Performance evaluation metrics
      - Virtual to actual simulation environment: feedback, interaction etc.
  - General DRL framework for intelligent finance decision making
    - To provide a unified, automatic and real-time intelligent decision-making service(driven by complex events)
- DRL with Lifelong Learning
- DRL with Constraints(budget/uplift/roi)
- DRL with Game theory and PGM, Multi-Agents System

Thanks!

- Q&A

DRL performance evaluation

- Single DNN
  - The classification accuracy of different activation functions with fixed network structure [1000, 1000, 800].

sigmoid	tanh	relu
0.647	0.644	0.563

  - The classification accuracy of different network structure with fixed activation function (sigmoid).

[1000]	[256]	[125]
0.647	0.643	0.654

- Trace norm MTL
  - The classification accuracy of different tensor decomposition methods (LAF, Tucker, TT) with sigmoid activation function.

Methods	[1000]	[256]	[125]
LAF	0.676	0.648	0.660
Tucker	0.686	0.672	0.699
TT	0.707	0.690	**0.709**

DRL performance evaluation

- Cross stitch MTL
  - The classification accuracy of different network structures.

[125]	[256]	[525]
0.670	0.662	0.659

- Experiment results with
  - Comparison the MTL methods on the random bucket data.

Methods	convRateLift	avgHitConvCost	avgAllConvCost
MTL-TN-TT	1.69%	4.13	3.57
MTL-TN-Tucker	3.67%	4.05	3.57
MTL-TN-LAF	1.64%	3.70	3.57
MTL-CS-125	-2.73%	3.55	3.57
MTL-CS-256	-9.16%	3.70	3.57
MTL-CS-525	-0.65%	3.46	3.57

  - Compared with random bucket data, the trace norm MTL have positive lift