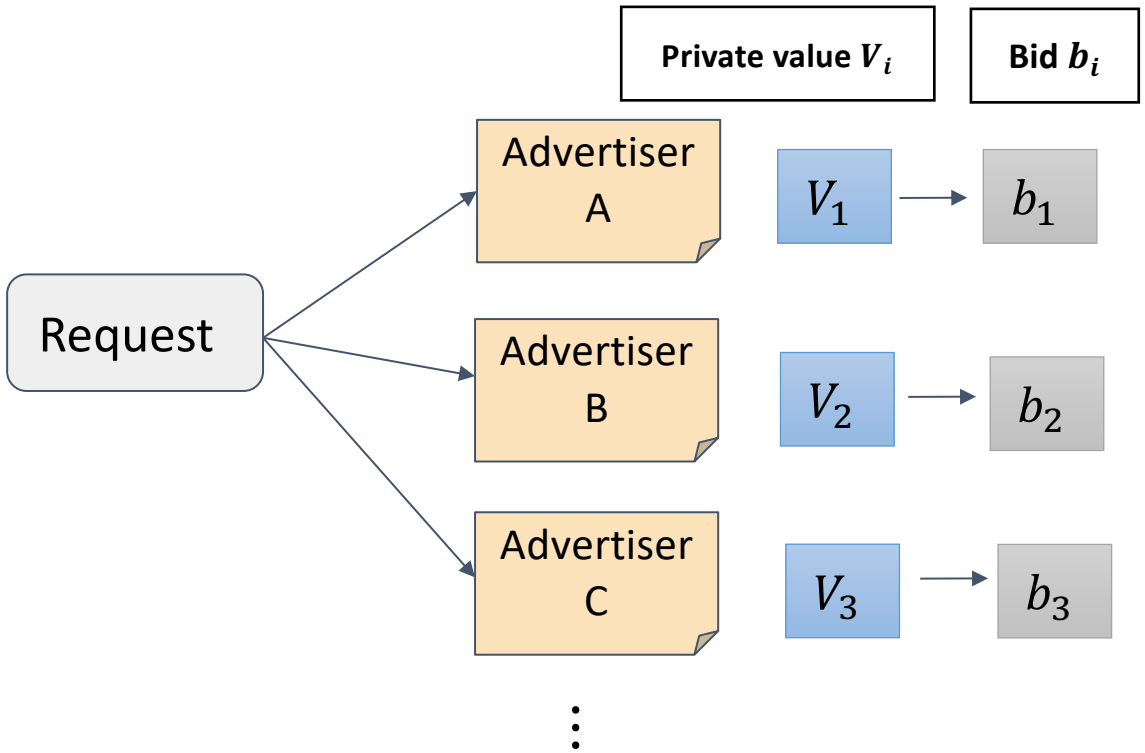
Part 4 – System Design

- Interaction between system components
  - Example – auction system with second-order pricing (presented by *Bo Yang*)
  - **Two-stage** *Exploration and offline learning*?
- Using pre-trained embedding (model)
  - The hidden debt of *negative sampling*
  - Mismatch between *model architecture*
  - Calibrating model output
- Some topics on decentralized system

Interaction between system components
-- Auction Example (Generalized second-price)



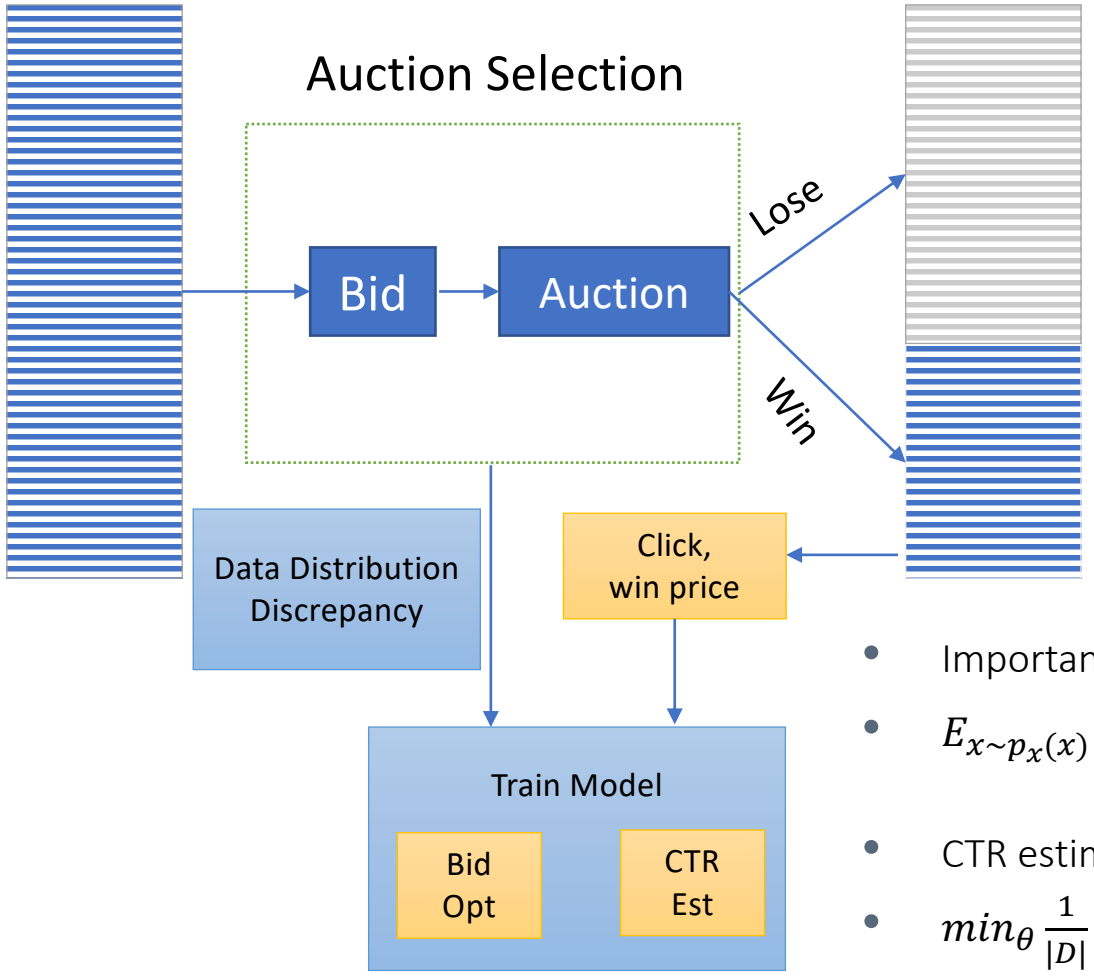
- K ad slots (K unknown in advance)
  - Top K bidders shown ads in feed
  - Each bidder pays the second highest bid
  - Truthful bidding is a dominant strategy
  - Expected reward is maximized when $b_i = V_i$

Ranking

- Advertisers are ranked according to bid and pCTR via ecpi.
- ecpi = pCTR * bid
  - pCTR = predicted click through rate (prediction model)
  - bid = price to pay if impression is clicked
  - ecpi = expected cost per impression

$$P_1(v_1, b_i, b()) = \begin{cases} v_1 - b_1 & if\, b_1 > b_i > \max\{b(v_2), \dots, b(v_{i-1}), b(v_{i+1}), \dots, b(v_n)\} \\ 0 & if\, b_1 < \max\{b(v_2), \dots, b(v_n)\} \end{cases}$$

User Response Prediction



$$q_x(x) = P(win|x, b_x) * p_x(x)$$

$q_x(x)$: Winning impression

$P(win|x, b_x)$: Prob of winning the auction

$p_x(x)$: Bid request

- The training data for user response estimation is biased and contains missing data.
  - When bid wins the auction, user response y, market price z, is observed
- Unbiased supervised learning problem
  - Loss-minimization problem on prediction data distribution $p_x(x)$
  - $\min_\theta E_{x\sim p(x)}[L(y, f_\theta(x))] + \lambda\Phi(\theta)$

- Importance sampling to reduce the bias of training data:
- $E_{x\sim p_x(x)}[L(y, f_\theta(x))] = \int_x p_x(x)L(y, f_\theta(x))dx = \frac{1}{|D|}\sum_{(x,y,x)\in D} \frac{L(y,f_\theta(x))}{1-\prod_{b_j<b_x}\frac{n_j-d_j}{n_j}}$

- CTR estimation for logistic regression:
- $min_\theta \frac{1}{|D|}\sum_{(x,y,x)\in D} \frac{-ylogf_\theta(x)-(1-y)\log(1-f_\theta(x))}{1-\prod_{b_j<b_x}\frac{n_j-d_j}{n_j}} + \frac{\lambda}{2}||\theta||_2^2$

User Response Prediction

Linear Models

Logistic regression, Bayesian probit regression.

Build the model based on the feature independence assumption.

Non-linear Models

Factorisation machine, tree models, (deep) neural networks.

More capacity of automatically learning feature interaction patterns without designing combining features.

More computational resources.

Transfer learning

Implicitly and jointly learn user's profile on both web browsing $\prod_{(x^c,y^c)\in D^c} P(y^c|x^c;\Theta)$ and ad response behaviors $\prod_{(x^r,y^r)\in D^r} P(y^r|x^r;\Theta)$.

$$\widehat{\Theta} = \max_{\Theta} P(\Theta) \prod_{(x^c,y^c)\in D^c} P(y^c|x^c;\Theta) \prod_{(x^r,y^r)\in D^r} P(y^r|x^r;\Theta)$$

Exploration and offline learning in a two-stage system?

- Most large-scale IR systems rely on the ***two-stage retrieval-ranking*** framework
  - There are usually several retrieval systems (focusing on different sources/signals) that generating the pool of candidates
  - The retrieval systems are usually *lightweight* and *using fewer signals* to reduce latency
  - A ranking system then further exploits *more complete signals* to rank the candidates
- The distribution mismatch of retrieval and ranking
  - E.g. the ***candidate distribution*** presented to the ranker is conditioned on the retrieval systems
  - E.g. the ***feedback distribution*** presented to the retrieval system is confounded by the ranker
  - E.g. different retrieval systems may exploit different ***feature distributions*** that are emphasized differently by the ranker
- Some potential consequences

[HKJ+20]
  - Common exploration strategies may suffer a *linear regret* $\mathcal{O}(T)$ due to a lack of communications between the two stages

[HKJ21]
  - *Unknown optimality* of optimizing ranking and retrieval systems *jointly vs. separately*

Using a pre-training system

- *Noise-contrastive estimation (NCE)* is often used as a unsupervised learning technique for pre-training the embeddings
  - Use ***semantic similarity*** to create ***self-supervision signal*** for learning
  - With the input of $(x, x^+, x_1^-, \ldots, x_k^-)$, the following objective is being optimized

$$\mathbb{E}_{x,x^+,\{x_i^-\}_{i=1}^k}\left[-\log \frac{\exp(\phi(x)^T\phi(x^+))}{\sum_{i=1}^k \exp(\phi(x)^T\phi(x_i^-))}\right]$$

  - The positive sample x^+ is often constructed from data, and the negative samples are obtained from *negative sampling*
  - *Cross-entropy loss* for *(extreme) multi-label classification*
- The pre-trained embeddings are then used by downstream tasks, however:
  - Imagine the downstream task uses $\phi(x)$ for *downstream classification*
  - What if a particular (x, x_i^-) turns out belonging to the *same class* in the downstream task?

The hidden debt of negative sampling

- The ***collision-coverage tradeoff***
  - Increasing #neg samples naturally leads to a better *density estimation* of the pre-training data, since it better covers the semantic space
  - But it also increases the chance of *class collision* in downstream data
  - The *class collision* can cause an excessive bias term even for generalizing to the same class-condition distribution:

$$\mathcal{O}\left(\mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{\log 1/\delta}{n}} + \text{collision bias}\right)$$

- Many advanced negative sampling methods are providing *heuristic solutions* to avoid class collision:
  - Finding ***hard negatives*** to reduce the chance that (x, x_i^-) belong to the same class
  - Using ***self-training*** (self-labelling) techniques to select appropriate negative samples
- Many other confounding factors can affect the usefulness of pre-trained embedding...

Mismatch between model architectures

- In NCE, the scores are constructed from the *inner products* between embeddings
  - The *model architecture* takes a factorization format – $\langle \phi(x_i), \phi(x_j) \rangle$
  - Downstream task may consume the embedding by a different architecture, e.g. $f_\theta(x) = \theta_1 \sigma(\theta_2^T \phi(x)))$
  - What are the potential consequences of the mismatch?

- The dual kernel view of linear regression
  - Suppose $f_\theta(x) = \theta^T \phi(x)$, then the property of the solution is decided by the kernel:

$$K_\phi(x_i, x_j) := \phi(x_i)^T \phi(x_j)$$

  - A nice correspondence with the scores that are optimized during pre-training

- But for the more complicated downstream architectures
  - There could exist parameterizations where $f_\theta(x_i),\ f_\theta(x_j)$ correspond to $\langle \phi(x_i), \phi(x_j) \rangle$ in a meaningful way (recall the ***NTK theory***)
  - In general, the mismatch can cause *non-negligible bias* even when generalizing to the same class-conditional distribution

[XYK+22]

Calibrating model output

- We have seen that even if the loss function has a *probabilistic interpretation* (e.g. logistic loss), the *exponential-tail property* can lead to margin-maximization behaviors
  - The output of the pre-trained models are *margins* rather than *conditional probabilities*
  - The predicted scores thus do not reflect the ***uncertainty*** of the prediction, and will tend to ***polarize***
  - Would like $\hat{f}_\theta$ to be calibrated wrt. some reference distribution
  - If the goal is to reconstruct the uncertainty of a true probability distribution:

$$\text{find } \hat{P} := c(\hat{f}_\theta)(X) \text{ s.t. } p(\hat{Y} = y \,|\, \hat{P} = p) = p,\ \forall p \in [0, 1]$$

[GPS+17]

  - If the goal is to reconstruct the empirical user interest $\hat{Q}$:

$$\text{force } D(\hat{P} \,\|\, \hat{Q}) \text{ to be small}$$

[Ste18]

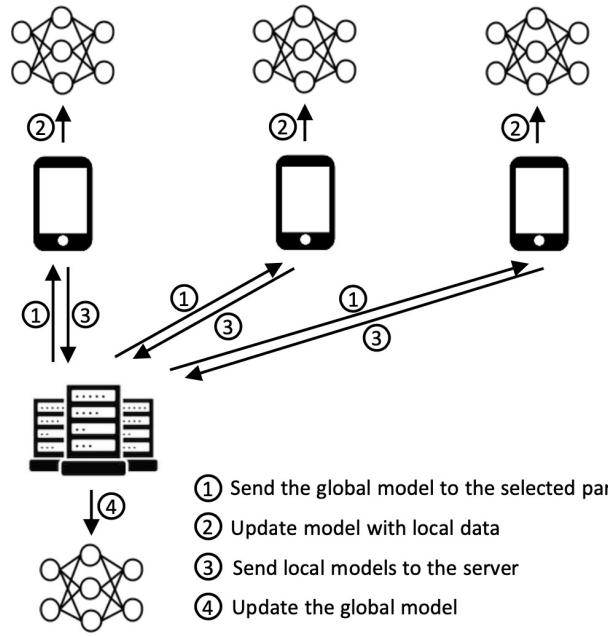
  - If the goal is to encourage invariance across environments $\{e_i\}_{i=1}^k$:

$$\text{constraint } p(\hat{Y} = y|\hat{P} = p, E = e_i) = p(\hat{Y} = y|\hat{P} = p, E = e_j),\ \forall i, j$$

[WFG+21]

Decentralized system

- *Collaborative training* of IR models across individual devices (clients) under *privacy restrictions*
  - We primarily discuss the framework of ***federated learning***
  - The key considerations aren't that different from *centralized learning*:
    - How to characterize the ***data distributions***?
    - What is the appropriate ***inductive bias*** for modelling?
    - Why does the various ***tradeoffs*** exist?
- About the source and target distributions
  - The source distribution is a *mixture* of client distribution with *known mixture probability* (e.g. given by the sample size), but it is *unknown* how the target distribution will be mixed



[MMR+17]

Decentralized system

- Following what we learnt in *Part 1*:
  - Using a large ***global model*** may cause poor performance on *under-represented clients*
  - The *unknown discrepancy* between the source and target distributions exacerbates this effect (imagine we care about generalizing uniformly across all clients)
  - *Importance weighting* may not resolve this challenge
  - A reasonable inductive bias is to employ ***personalized local models*** to complement the global model
  - But we are also subject to **privacy constraint** – even the local models are not supposed to memorize the clients' sampled data
  - Lacking ability to memorize -> less stable -> poorer overall generalization performance
- Some emerging tradeoffs for decentralized systems:
  - The traditional *bias-variance* tradeoff is now partly caused by the *relative strength* of **local vs. global** model

    [BWD+22]
  - The **privacy vs. accuracy** tradeoff now also affects the bias-variance tradeoff

Summary of Part 4

- System design has a high complexity and requires fundamental understanding of the problem and algorithm
  - *How does ML solution fit the business need*? (the auction example)
  - *What is a good inductive bias for the data distributions involved*? (the decentralized system example)
  - *How do the system components interact*? (the retrieval-ranking example)
  - *What is the gap between ML results and the desired goal?* (the calibration example)
  - ……
- The external constraints lead to a hierarchy of tradeoffs
  - The modelling and learning for pattern recognition is often the cornerstone
  - External constraints, e.g. service time, privacy, how ML outputs is used, can create several layers of algorithmic tradeoffs
- Good solutions often navigate through the problem, algorithm, and tradeoff

Some epilogue

- Embrace “universal model” vs. diving into domain knowledge
- System vs. $\sum functions$
- Mind (theory) vs. hand (engineering)

- Establish IR as a leading discipline in ML/AI
- Oceans of research topics, converting to $$
- Leading business in Web 2.0, even more so in Web 3.0?

Thank you for attending!

Da Xu
ML Manager, Staff ML Eng
Search & Recommendation
Walmart Labs

- Acknowledgement -- would like to thank:
  - Bo Yang (LinkedIn)
  - Chuanwei Ruan (Instacart)
  - Evren Korpeoglu, Sushant Kumar, Kannan Achan (Walmart Labs)
  - Many other peers, collaborators, annonymous reviewers ...
- DM for collab (research & job opportunity)
- Appreciate any feedback, comments, suggestions!

Contact: daxu5180@gmail.com (daxu5180.github.io)
Website: theoreticalfoundation4irsystem.github.io/Tutorial-KDD22

Discussions and Q&A

Part 1: Pattern Recognition

- *ML Basics*:
- *Understanding Deep Learning*
- *From classification to ranking*
- *Domain challenges of IR*

Part 2: Intervention and Causal Inference

- *The causal language*
- *Design and inference*
- *Observational studies and offline learning*
- *Connection to pattern recognition*

Part 3: Action and sequential decision making

- *Online learning with evaluate feedback*
- *The power of structure*:
- *Policy learning*

Part 4: System design

- *Interaction between system components*
- *Pre-training, negative sampling, calibration*
- *Decentralized system*

Contact: daxu5180@gmail.com (daxu5180.github.io)
Website: theoreticalfoundation4irsystem.github.io/Tutorial-KDD22

References

- [MP21] Mohan K, Pearl J. Graphical models for processing missing data[J]. Journal of the American Statistical Association, 2021, 116(534): 1023-1037.
- [AKJ14] Ailon, Nir, Zohar Karnin, and Thorsten Joachims. "Reducing dueling bandits to cardinal bandits." *International Conference on Machine Learning*. PMLR, 2014.
- [RR83] Rosenbaum P R, Rubin D B. The central role of the propensity score in observational studies for causal effects[J]. Biometrika, 1983, 70(1): 41-55.
- [PM18] Pearl J, Mackenzie D. The book of why: the new science of cause and effect[M]. Basic books, 2018.
- [Ken16] Kennedy E H. Semiparametric theory and empirical processes in causal inference[M]//Statistical causal inferences and their applications in public health research. Springer, Cham, 2016: 141-167.
- [VR06] Van Der Laan M J, Rubin D. Targeted maximum likelihood learning[J]. The international journal of biostatistics, 2006, 2(1).
- [CCD+18] Chernozhukov V, Chetverikov D, Demirer M, et al. Double/debiased machine learning for treatment and structural parameters[J]. 2018.
- [YJ09] Yue Y, Joachims T. Interactively optimizing information retrieval systems as a dueling bandits problem[C]//Proceedings of the 26th Annual International Conference on Machine Learning. 2009: 1201-1208.
- [WB19] Wang Y, Blei D M. The blessings of multiple causes[J]. Journal of the American Statistical Association, 2019, 114(528): 1574-1596.
- [WLC+18] Wang Y, Liang D, Charlin L, et al. The deconfounded recommender: A causal inference approach to recommendation[J]. arXiv preprint arXiv:1808.06581, 2018.
- [XRK+20] Xu D, Ruan C, Korpeoglu E, et al. Adversarial counterfactual learning and evaluation for recommender system[J]. Advances in Neural Information Processing Systems, 2020, 33: 13515-13526.

References

- [PJS17] Peters J, Janzing D, Schölkopf B. Elements of causal inference: foundations and learning algorithms[M]. The MIT Press, 2017.
- [AB21] Angelopoulos A N, Bates S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification[J]. arXiv preprint arXiv:2107.07511, 2021.
- [XY22] Xu D, Yang B. Rethinking learning with missing feedback for recommendation. Openreview, 2022.
- [TR19] Tu S, Recht B. The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint[C]//Conference on Learning Theory. PMLR, 2019: 3036-3083.
- [DLY+20] Dong K, Luo Y, Yu T, et al. On the expressivity of neural networks for deep reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2020: 2627-2637.
- [BHM+19] Belkin M, Hsu D, Ma S, et al. Reconciling modern machine-learning practice and the classical bias–variance trade-off[J]. Proceedings of the National Academy of Sciences, 2019, 116(32): 15849-15854.
- [SHN+18] Soudry D, Hoffer E, Nacson M S, et al. The implicit bias of gradient descent on separable data[J]. The Journal of Machine Learning Research, 2018, 19(1): 2822-2878.
- [GRS18] Golowich N, Rakhlin A, Shamir O. Size-independent sample complexity of neural networks[C]//Conference On Learning Theory. PMLR, 2018: 297-299.
- [XYR21] Xu D, Ye Y, Ruan C. Understanding the role of importance weighting for deep learning. ICLR, 2021.
- [JGH18] Jacot A, Gabriel F, Hongler C. Neural tangent kernel: Convergence and generalization in neural networks[J]. Advances in neural information processing systems, 2018, 31.

References

- [SJ15] Swaminathan A, Joachims T. Counterfactual risk minimization: Learning from logged bandit feedback[C]//International Conference on Machine Learning. PMLR, 2015: 814-823.
- [HKJ21] Hron J, Krauth K, Jordan M, et al. On component interactions in two-stage recommender systems[J]. Advances in Neural Information Processing Systems, 2021, 34: 2744-2757.
- [HKJ20] Hron J, Krauth K, Jordan M I, et al. Exploration in two-stage recommender systems[J]. arXiv preprint arXiv:2009.08956, 2020.
- [XZL+20] Xu K, Zhang M, Li J, et al. How neural networks extrapolate: From feedforward to graph neural networks[J]. arXiv preprint arXiv:2009.11848, 2020.
- [JGH18] Jacot A, Gabriel F, Hongler C. Neural tangent kernel: Convergence and generalization in neural networks[J]. Advances in neural information processing systems, 2018, 31.
- [GJK21] Gatmiry K, Jegelka S, Kelner J. Optimization and Adaptive Generalization of Three layer Neural Networks[C]//International Conference on Learning Representations. 2021.
- [FZ20] Feldman V, Zhang C. What neural networks memorize and why: Discovering the long tail via influence estimation[J]. Advances in Neural Information Processing Systems, 2020, 33: 2881-2891.
- [CLV08] Clémençon S, Lugosi G, Vayatis N. Ranking and empirical minimization of U-statistics[J]. The Annals of Statistics, 2008, 36(2): 844-874.
- [RCM+05] Rudin C, Cortes C, Mohri M, et al. Margin-based ranking meets boosting in the middle[C]//International Conference on Computational Learning Theory. Springer, Berlin, Heidelberg, 2005: 63-78.
- [CW10] Chapelle O, Wu M. Gradient descent optimization of smoothed information retrieval metrics[J]. Information retrieval, 2010, 13(3): 216-235.

References

- [MMR09] Mansour Y, Mohri M, Rostamizadeh A. Domain adaptation: Learning bounds and algorithms[J]. arXiv preprint arXiv:0902.3430, 2009.
- [ZZS+16] Zhao S, Zhou E, Sabharwal A, et al. Adaptive concentration inequalities for sequential decision problems[J]. Advances in Neural Information Processing Systems, 2016, 29.
- [FS08] Foster D P, Stine R A. α-investing: A procedure for sequential control of expected false discoveries[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2008, 70(2): 429-444.
- [JSR19] Johansson F D, Sontag D, Ranganath R. Support and invertibility in domain-invariant representations[C]//The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019: 527-536.
- [KSW+15] Kveton B, Szepesvari C, Wen Z, et al. Cascading bandits: Learning to rank in the cascade model[C]//International conference on machine learning. PMLR, 2015: 767-776.
- [CG17] Chowdhury S R, Gopalan A. On kernelized multi-armed bandits[C]//International Conference on Machine Learning. PMLR, 2017: 844-853.
- [GPS+17] Guo C, Pleiss G, Sun Y, et al. On calibration of modern neural networks[C]//International conference on machine learning. PMLR, 2017: 1321-1330.
- [NS17] Nesterov Y, Spokoiny V. Random gradient-free minimization of convex functions[J]. Foundations of Computational Mathematics, 2017, 17(2): 527-566.
- [AKL+21] Agarwal A, Kakade S M, Lee J D, et al. On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift[J]. J. Mach. Learn. Res., 2021, 22(98): 1-76.
- [DKW+20] Du S S, Kakade S M, Wang R, et al. Is a Good Representation Sufficient for Sample Efficient Reinforcement Learning?[C]//International Conference on Learning Representations. 2020.

References

- [ZLK+20] Zanette A, Lazaric A, Kochenderfer M, et al. Learning near optimal policies with low inherent bellman error[C]//International Conference on Machine Learning. PMLR, 2020: 10978-10989.
- [LS20] Lattimore T, Szepesvári C. Bandit algorithms[M]. Cambridge University Press, 2020.
- [Ste18] Steck H. Calibrated recommendations[C]//Proceedings of the 12th ACM conference on recommender systems. 2018: 154-162.
- [WFG+21] Wald Y, Feder A, Greenfeld D, et al. On calibration and out-of-domain generalization[J]. Advances in neural information processing systems, 2021, 34: 2215-2227.
- [XY22] Xu D, Yang B. Revisiting pre-trained embedding for e-commerce machine learning. Under review, 2022.
- [WAS21] Weisz G, Amortila P, Szepesvári C. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions[C]//Algorithmic Learning Theory. PMLR, 2021: 1237-1264.
- [MMR17] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.
- [BWD22] Bietti A, Wei C Y, Dudik M, et al. Personalization Improves Privacy-Accuracy Tradeoffs in Federated Learning[C]//International Conference on Machine Learning. PMLR, 2022: 1945-1962.
- [DMJ10] Duchi J C, Mackey L W, Jordan M I. On the consistency of ranking algorithms[C]//ICML. 2010.