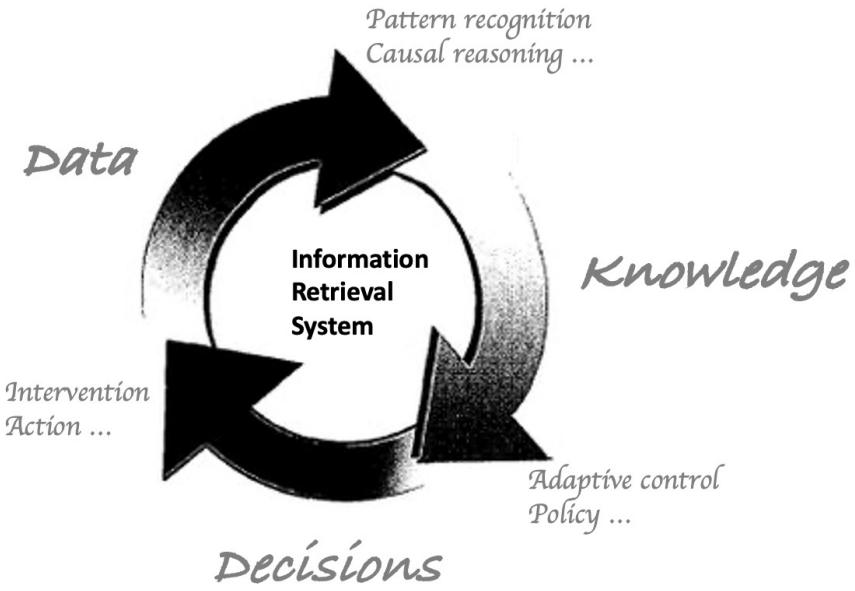
Theoretical Foundation for Modern Information Retrieval System

KDD 2022 Tutorial

Presenter: ***Da Xu***

Aug. 14, 2022



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

What this tutorial is about?

- Some observations and motivations:
  - Problem solving = f(data, deep learning, Hail Mary)?
  - Innovation in IR = rephrasing ideas/models developed elsewhere?
  - IR system = $\sum functions$?

- I know all about afternoon sessions and lunch coma, so:
  - Max intuition, minimum math!
  - A broad range of topics will be covered for the diversity of story telling.

- After 4 hours, hopefully, you will:
  - Gain a rough idea about: What makes an IR system system?
  - Capture the key insights and challenges behind IR's pattern recognition, causal inference, decision making, and system design.

Historical notes about IR

-- what makes IR the way it is today

- *Origins of IR*
  - **1920s**: "... Search for documents stored on microfilm ... ",
  - **1940s**: ".. US military indexing and retrieve documents captured from German ... "
- *Cranfield experiments, Lockheed system*
  - **1960s**: "exact retrieval", indexing, compressing, abstracting ...
  - **1970s**: first large-scale retrieval system
- *ACM SIGIR (1978 -)*
  - **1980s**: vector space model, clustering, naive Bayes, probabilistic model
  - **1990s**: search engines, graph link analysis, learning to rank, ...
  - 21st century: media search, recommender system, document classification, spam filtering, question answering, personalization

What makes IR a special discipline?

- **Intervenable** vs. **Observational**
  - Ability to take actions and interact with the world?

- **Generative** vs. **Discriminative**
  - Knowledge about data generation useful to the task?

- **Evaluative** vs. **Instructive**
  - Interpretate collected feedback as evaluation or instruction?

Introduction – Theoretical Foundation for IR Sys

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

Part 1: Pattern Recognition

- *ML Basics*:
  - Expressivity, Optimization, Generalization
- *Understanding Deep Learning*:
  - Double descent in bias-variance tradeoff
  - The kernel regimes
  - Implicit regularization
  - Generalization, memorization, and subpopulation
  - Role of model architecture
- *From classification to ranking*
  - Consistency / generalization of ranking
  - On smooth ranking loss
- *Domain challenges of IR data*:
  - Unlabellness: transductive (semi-supervised), domain adaptation
  - Sparsity: representation learning, graph neural network
  - Debias, extrapolation

Introduction – Theoretical Foundation for IR Sys

Part 2: Intervention and Causal Inference

- *The causal language*
  - Pearl and Rubin causal model
  - Invariant mechanism, confounding, randomization, counterfactual
- *Design and inference*
  - A/B testing, metric, and continuous monitoring
  - Best-arm identification and sequential testing
  - Interleaving and dueling
- *Observational studies and offline learning*
  - Is observational studies a missing data problem?
  - Double learning and targeted maximum likelihood learning
  - Propensity weighting method and counterfactual learning
  - Multiple causes, deconfounding, robust optimization
- *Conection to IR pattern recognition*
  - Causality and learning
  - Conformal prediction, learn-then-test

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

Introduction – Theoretical Foundation for IR Sys

Part 3: Action and sequential decision making

- *Online learning with evaluate feedback*:
  - Multi-armed bandit and exploration-exploitation
  - Dynamic system, planning, Markov decision process
- *The power of structure*:
  - Benefiting from structured policy & reward
- *Policy learning*
  - The hateful horizen & distribution shift
  - Supervised / representation learning challenged
  - Gradient or random search -- zeroth-order optimization
  - Model-based vs. model-free

Part 4: System design

- *Interaction between system components*
- *Pretrained embedding, negative sampling, calibration*
- *Decentralized system*

Da Xu

ML Manager, Staff ML Eng

Search & Recommendation

Walmart Labs

Bo Yang

ML Engineer

LinkedIn Ads AI

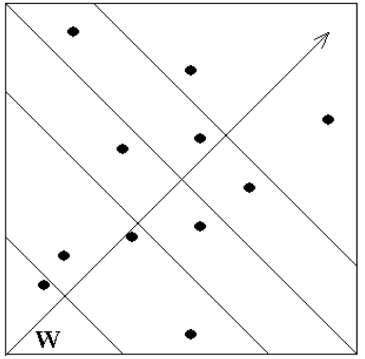
Just a few things before rock-n-roll …

- Three sessions + two breaks
  - 1st break at 2:00 pm.
  - 2nd break at 3:00 pm.
- Q&A and discussion
  - During the break
  - Approaching the end
- Very important: your feedback and comments!

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

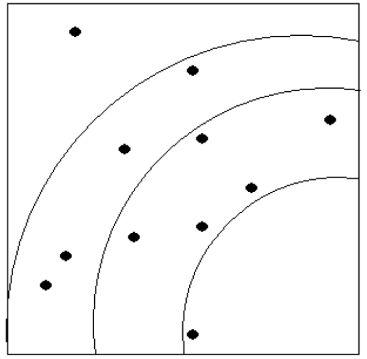
Part 1 – Pattern Recognition with Feedback Data

- Given a user query (or hidden intention) represented by X_u, the system reveals documents (items) each represented by X_i, and the user provide feedback $Y_{u,i}$



- Does it make a difference $Y_{u,i} \in \{0, 1\}$ or $Y_{u,i} \in \{0, 1, 2, \ldots, 5\}$?
  - Discretized response surface vs. partition boundary?
  - Partitioning boundary vs. decision boundary?
  - Will come back to this later ...

- Collected feedback data $\{(X_u, X_i, Y_{u,i}) \mid (u, i) \in D\}$



- Employ $f \in \mathcal{F}$ to learn the patterns s.t. $f_{u,i} := f(X_u, X_i)$ predicts preference / how likely to click.

A 10-min crash course on ML basics

- *Estimation* and *prediction* are central to statistics and ML.
- We start with the former which does not involve relationship among variables:
  - Independent random variable $X_1, \ldots, X_n$ sampled from the same distribution, assuming standard Gaussian or *1-subgaussian*
  - Often interested in the mean $\mu := \mathbb{E}[X_i]$ -- when is the sample average a good estimation?
  - Let $\hat{\mu} = 1/n \sum_{i=1}^{n} X_i$, then according to *Chebyshev's inequality*:

$$p(\hat{\mu} \geq \mu + \epsilon) \leq \exp(-n\epsilon^2/2), \forall \epsilon > 0$$

  - The above *concentration* result holds similarly in the other direction. They imply:

$$\mu \in \left[\hat{\mu} - \sqrt{\frac{2\log(1/\delta)}{n}}, \hat{\mu} + \sqrt{\frac{2\log(1/\delta)}{n}}\right], \quad \text{with probability at least } 1-\delta$$

  - The same $1/\sqrt{n}$ rate as *Law of Large Numbers*.

A 10-min crash course on ML basics (classification)

- Prediction with a *full knowledge* about the population distribution $p(X, Y)$
  - Via optimization using loss function and risk $R(f) := \mathbb{E}[\ell(f(X), Y)]$
  - The optimal predictor follows the likelihood ratio test, $f^*(x) = \mathbb{1}\left[\frac{p(X = x|Y = 1)}{p(X = x|Y = 0)} \geq \gamma\right]$ where the threshold depends on the 0-1 loss
  - Using Bayes rule, it is equivalent to $f^*_{Bayes}(x) = \arg\max_{y \in \{0,1\}} p(Y = y|X = x)$
  - Its optimality <- *Neyman-Pearson lemma*.
- Prediction with *i.i.d random samples* $(x_1, y_1), \ldots, (x_n, y_n) \sim p(X, Y)$
  - "Supervision" – availability of the label
  - Proper loss function -- Bayes consistent in the sense that: $\mathbb{1}[f^*_\ell(x) > 0] = f^*_{Bayes}(x)$ (exponential loss, logistic loss, ...)
  - Optimizing empirical risk: $R_n(f) := 1/n \sum \ell(f(x_i), y_i)$
  - Why does it work? $R(f) = R_n(f) + (R(f) - R_n(f))$

A 10-min crash course on ML Bascis

- Take a look at $R(f) = R_n(f) + (R(f) - R_n(f))$:
  - Since we are searching within $f \in \mathcal{F}$, is the *hypothesis class* expressive enough to represent the inductive bias of the data and task? (**Expressivity**)
  - How to find the hypothesis with: $\arg\min R_n(f)$? (**Optimization**)
  - How large could $R(f) - R_n(f)$ possibly be? (**Generalization**)

- Understanding the *expressivity* of $f_\theta(x), \theta \in \Theta, x \in \mathcal{X}$:
  - Structure of the *feature space* – vector space, string, graph, kernel, basis functions, etc
  - Structure of the *model space* – linear, sequential, modular, convolution, etc
  - Parameterization and smoothness

- Optimization for $R(\theta) := R(f_\theta)$, assuming convexity:
  - "Descent direction" – $R(\theta + \alpha v) < R(\theta), \alpha > 0$
  - Characterize descent direction – *negative gradient* (under good condition), because

$$v = \nabla R(\theta) \Rightarrow R(\theta + \alpha v) < R(\theta)$$

A 10-min crash course on ML Basics

- Gradient descent optimization:
  - Iterative update $\theta_{t+1} = \theta_t - \alpha_t \nabla R(\theta_t)$
  - Usually ends at *stationary point* where gradient vanishes $\theta^* : \nabla R(\theta^*) = 0$
  - Stochastic GD – evaluate gradient at a random sample, gradient is unbiased

    $$g(\theta; i) = \nabla \ell(f_\theta(x_i), y_i);\ \mathbb{E}[g(\theta; i)] = \nabla R(\theta)$$

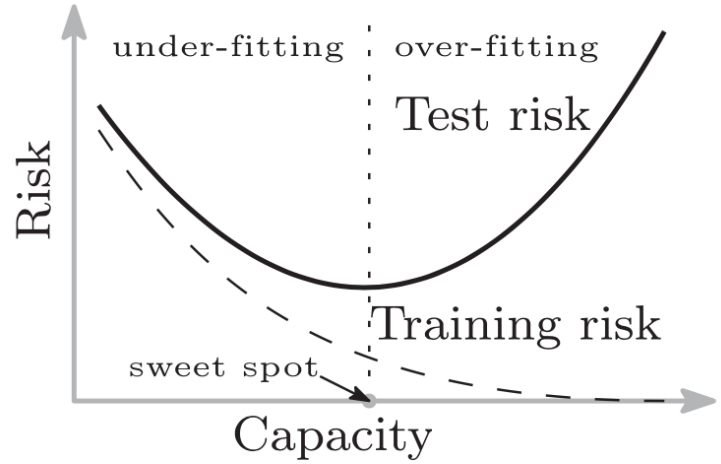
  - When and Why does SGD work?

    If the gradient's variance $\mathbb{E}[\|g(\theta; i)\|^2]$ is bounded, then the accumulated error due to the randominess still converges at a $\sqrt{T}$ rate, i.e. $\mathbb{E}[R(\bar{\theta}) - R(\theta^*)] = \mathcal{O}(1/\sqrt{T})$

- Generalization of empirical risk minimization:
  - Recall $R(f) = R_n(f) + (R(f) - R_n(f))$ -- optimization makes the 1st term small, the 2nd term is *generalization gap*
  - Generalization gap has been shown to related with *model complexity*, *stability*, and *margin*

A 10-min crash course on ML Basics

- The most well-known criteria for *generalization gap* is *model complexity*
  - *VC dim:* abiliy to conform to an arbitary labeling of data points
  - *Rademarcher complexity:* ability to interpolrate random sign pattern
  - *"Uniform convergence":* applies to all functions in $\mathcal{F}$



For *L-Lipschitz loss*: $R(f) - R_n(f) \leq 2L\mathcal{R}_n(\mathcal{F}) + 3\sqrt{\frac{\log 1/\delta}{n}}, \forall f \in \mathcal{F}$

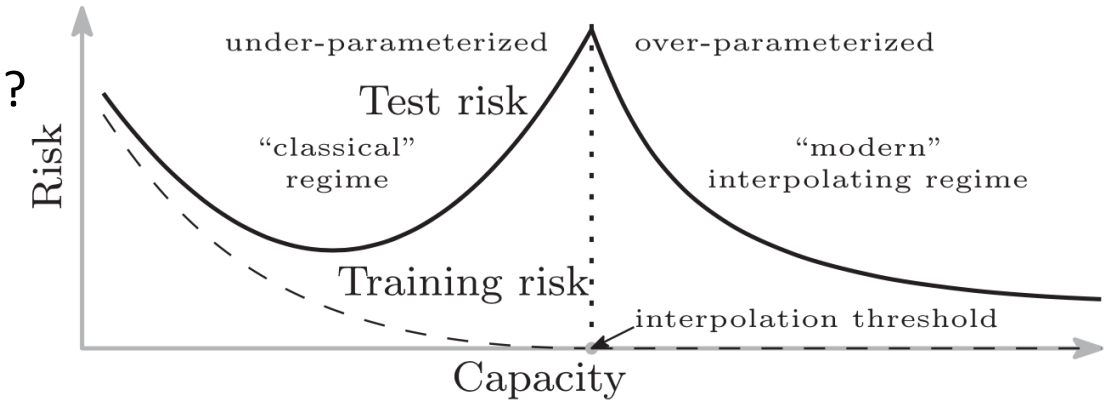
- *Margin-based* generalization bound
  - *Margin* is also a form of regularization (why?) that improves generalization
  - Let $R_n^\gamma(f)$ be the empirical risk wrt. the margin loss $\mathbb{1}[yf(x) \leq \gamma]$: $R(f) - R_n^\gamma(f) \lesssim \frac{\mathcal{R}(\mathcal{F})}{\gamma} + \sqrt{\frac{\log 1/\delta}{n}}$

- *Algorithmic-stability* bound
  - *Sensitivity* to the change of a single training sample
  - If *insensitive* to such perturbation, then generalization gap should be small. (Will revisit this in more detail.)

Understanding deep learning

- When classical bias-variance phenomenon fail
  - The *double descent* phenomenon
  - Finding the best inductive bias and global optimum?
  - Rethinking uniform generalization.
- The *limiting kernel* regimes
  - On the expressivity of overparameterized NN
  - The *optimization path* under GD
  - *Neural tangent kernel*
  - Adaptive feature learning
  - On the complexity of NN *

[BHM+19]



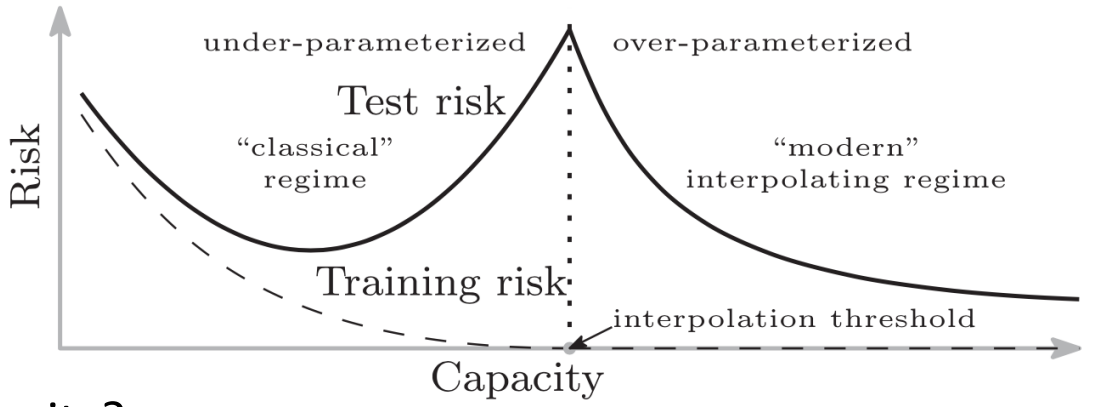
* we will use $\mathcal{C}(\mathcal{F})$ as a general complexity measure, which could be the previous Rademarcher complexity, naive count, VC dimension, covering number, etc

Understanding deep learning

- *Implicit regularization* of overparameterized models
  - Silver lining for non-convex ERM
  - Does SGD randomly pick a local optimum?
  - The role of *minimum-norm solution*
- Current understanding of how NN generalize:
  - A combined story of overparameterization, implicit regularization, and kernel
  - Through the lens of *memorization*
  - The under-investigated *fairness for subpopulation*
- The role of *structure* for IR models
  - Efficient optimization -- rethinking *auto-differentiation and backpropagation*
  - Inductive bias and the *smoothness of interpolating hypothesis*

Understanding deep learning

- Before the double descent phenomenon:
  - Overparameterized DL models routinely achieve almost-zero training risk
  - Desipte the near-perfect fitting and very high model class complexity, they still generalize well, even under high label noise
  - The sweet spot between underfitting and overfitting is gone?
- Beyond the *interpolation point*
  - Modern practice (d>n) is shown to the right of interpolation point
  - Training and testing risk decrease simultaneously
  - What is the inductive bias that perfectly describes this phenomenon?
  - How does SGD find a good solution depite non-convexity?
  - Can uniform convergence still explain NN's generalization?



Let's start with some intuition

- Guessing the general inductive bias for overparameterized NN models
  - Uniform approximation theorem?

    -- Capacity of functional class does not necessarily reflect how well the predictor matches the problem at hand ...
  - Represent many decision boundaries that perfectly separate data + SGD find the smoothest one?

    -- Seems like a form of *Occam's razor*: the simplest explanation compatible with observations
- But still huge gaps with current understanding, e.g.
  - 1). why can we expect the bahavior of SGD for non-convex function;
  - 2). interpolating the data means complexity grows with data (especially for *noisy data* cases), why can we expect any generation from the solution?
- Optimization-wise, could it be over-parameterization give rise to some *tractability*?
- Generalization-wise, could it be over-parameterization and SGD optimization to find a *small-norm* solution (corresponding to smoothness) + high-capacity component?

Gradient decent in the over-parameterized regime

- A change of direction – from *convergence of parameter* to *convergence of prediction*
  - Regardless of the parameter space, empirical evidence suggests the predictions can constantly reach zero training risk
  - Consider *square loss* for simplicity: $(f_\theta(x_i) - y_i)^2$, and $f_\theta(x) = \theta^T x$
  - First note that under GD update, it holds:

$$\hat{y}_{t+1} - y = (I - \alpha X X^T)(\hat{y}_t - y)$$

  - So, if α is small and XX^T is strictly positive definite, then predictions converge to training labels
  - A necessary condition for that is $d > n$.
  - Let's move on to non-convex models. Using first-order Taylor expansion, under GD update:

$$\hat{y}_{t+1} - y = (I - \alpha \mathcal{J} f_\theta(x)^T \mathcal{J} f_\theta(x))(\hat{y}_t - y) + \alpha \epsilon_t$$

    where $\mathcal{J} f$ is the Jacobian (because x, y are now the vectorized collections), and ϵ_t is second-order residue (usually small).
  - So non-convexity not really matter here if step size is small – but the Jacobian is again likely to be positive definite if $d > n$.

Gradient decent in the over-parameterized regime

- Just as it seems non-convexity doesn't really get in the way for *prediction's convergence under over-parameterization*, reaching *zero-loss* leads to something even better
  - because of zero-loss: $y = \theta^T x$
  - because GD/SGD: $\theta_{t+1} = \theta_t - \alpha e_t x_t$, e_t : gradient of the loss as current prediction
    the algoirthm searches over a space with dimension equal the number of data, and
    $\hat{\theta} = X^T \beta$ because we are in the span of the data (assume initialization at zero)
  - what is special about solutions of this form?
  - assume another solution, $\hat{\theta}' = X^T \beta + v,\ v \perp x_i$
  - easy to verify: $\|\hat{\theta}'\|_2 = \|\hat{\theta}\|_2 + \|v\|_2$ -- so $\hat{\theta} = X^T \beta$ is also the minimum-norm solution.
- Recall *kernel methods* and the *representer theorem*
  - consider L2 regularized ERM, then: $\hat{\theta}' = X^T \beta + v,\ v \perp x_i \rightarrow \frac{1}{n} \sum_i \ell(\beta^T X x_i, y_i) + \lambda \|X^T \beta\|^2 + \lambda \|v\|^2$
  - risk minimized at $v = 0$ -- optimial lies in the span of the data
  - can instead search for solution in the kernel expansion: $K = XX^T$

Gradient decent in the over-parameterized regime

- Minimum-norm solution and margin
  - Recall from SVM that if we correctly classify all data, then margin satisfies $\gamma(\theta) = \min_i \frac{y_i \theta^T x_i}{\|\theta\|}$
  - Suppose we interpolate the data: $y_i = \theta^T x_i$, then $\gamma(\theta) = \|\theta\|^{-1}$

    solution from SGD achieves interpolatiion and max-margin solution
- Moving to non-linear models
  - Similar convergence of prediction behavior:

$$\hat{y}_{t+1} - y = (I - \alpha \mathcal{J} f_{\theta^{(t)}}(x)^T \mathcal{J} f_{\theta^{(t)}}(x))(\hat{y}_t - y) + \alpha \epsilon_t,\ \epsilon_t = o(\alpha\kappa\|\hat{y}_t - y\|^2)$$

    where κ bounds the curvature.
  - Nonconvexity doesn't really stand in the way to follow the previous argument
  - Let's make it rigorous for classification methods, with
    - loss function has *exponential-tail* behavior
    - the predictor is *homogeneous* – $f(c \cdot \theta, x) = c^q f(\theta, x)$
    - some *smoothness and Lipschitzness* of the predictor

Gradient decent in the over-parameterized regime

- Implicit regularization of gradient descent
  - A difference with the squre loss is that gradient descent now tends to increase the norm of the parameters, i.e. $\lim_{t\to\infty} \|\theta^{(t)}\| = \infty$
  - Why? Intuitively, both the risk and gradient has the form: $\sum_i C_i \exp(-y_i f_\theta(x_i))$

    and since $f_\theta(x) = \|\theta\|^q f_{\theta/\|\theta\|}(x)$, it holds that: $\lim_{t\to\infty} \exp(-y_i f_{\theta^{(t)}}(x_i)) \to 0, \forall i$

    so we reach both zero-loss (*interpolation*) and zero-gradient (*stationarity*)
  - One further implictation: due to the exponential tail behavior, during optimization, the risk will be *dominated* by the sample with ***largest margin***: $\arg\max_i y_i f_{\theta^{(t)}}(x_i)$
  - They are like *support vectors* in SVM!
  - So the optimization path is similar to that of the hard-margin SVM:

$$\min \|\theta\|_2 \ \text{ s.t. } \ y_i f_\theta(x_i) \geq 1, \forall i$$

  - Under appropriate step size and separability, let $\hat{\theta}$ be the solution to the above *L2 max-margin problem*, then the optimzation path of GD follows: $\lim_{t\to\infty} \frac{\theta^{(t)}}{\|\theta^{(t)}\|} = \frac{\hat{\theta}}{\|\hat{\theta}\|}$

[SHN+18]

Margin, scale, and generalization of NN

- We just saw GD/SGD optimization carries the implict bias of favoring minimum-norm large-margin solution, which suggest for generalization:
  - The *critical role of* ***margin*** – where margin can often be treated as a form of ***generalization***
  - The ***size*** (scale) of the parameters matters perhaps more than the ***number of parameter***

    -- can this be a new direction for explaining the complexity of NN and grant compatible with uniform generalization?
  - Recall that for linear classifier, $\gamma(\theta) = \|\theta\|^{-1}$
- A margin bound for uniform generalization of NN
  - Margin is sensitive to scale, which depends on the norm of NN
  - With parameters layered up, defining a norm for NN is non-trivial
  - *Spectral norm / Forbenius norm* of each layer -> ***peeling*** -> *size independent complexity* of NN

    -- together with the *margin bound*, we finally reach:

[GRS19]

$$p_{\text{test}}(yf_\theta(x) \le 0) \le \frac{1}{n}\sum_i \mathbb{1}[y_i f_\theta(x_i) \le \gamma] + \frac{\sqrt{q}\sum_{j=1}^{q} M(q)}{\theta\sqrt{n}}, \; M(q) \text{ is the norm bound of layer q}$$

Implicit regularization and the kernel regime

- The previous result captures the essence of margin, but that's not the whole story
  - *Margin* might just be one form of implict regularization
  - The *norm bounds* can be loose in general
  - Equivalent notion exists for such as *square loss*?
  - remember in the linear regime, we connect the implicit bias of GD with the *representer theorem and kernel*
  - This connection holds for NN as well, under certain *limiting approximation*
  - an observation: many parameters (under random initialization) change very little during training, so –

$$\hat{y}_{t+1} - y = (I - \alpha \mathcal{J} f_{\theta^{(t)}}(x)^T \mathcal{J} f_{\theta^{(t)}}(x))(\hat{y}_t - y) + \alpha \epsilon_t$$

$$\Downarrow$$

$$\hat{y}_{t+1} - y = (I - \alpha \mathcal{J} f_{\theta^{(0)}}(x)^T \mathcal{J} f_{\theta^{(0)}}(x))(\hat{y}_t - y) + \alpha \epsilon'_t,$$

  - Define the ***Neural Tangent Kernel*** $K(x, x') = \mathbb{E}_{\theta^{(0)}}\left[\langle \nabla f_{\theta^{(0)}}(x), \nabla f_{\theta^{(0)}}(x') \rangle\right]$
    which depends only on the input, initialization and model structure

[JGH18]

The kernel regime of NN

- An alternative way to understand the connection between GD implicit bias and neural tangent kernel is as follows:
  - Directly conider the Taylor expansion at initialization:

$$f_\theta(x) = f_{\theta^{(0)}}(x) + \langle \nabla f_{\theta^{(0)}}(x), \theta - \theta^{(0)} \rangle + \epsilon$$

  - We can always use reparameterization to get rid of the intercept – $f_\theta(x) \approx \langle \nabla f_{\theta^{(0)}}(x), \theta - \theta^{(0)} \rangle$
  - In this way, $\nabla f_{\theta^{(0)}}(x)$ acts like a *feature lift ,* and the correspond RKHS is induced by the *neural tangent kernel*: $K(x, x') = \langle \nabla f_{\theta^{(0)}}(x), \nabla f_{\theta^{(0)}}(x') \rangle$
  - According to the previous derivation for *kernelized ERM*, it is easy to see the problem becomes:

$$\text{minimize}_\beta \frac{1}{n} \sum_i \ell(e_i^T K\beta, y_i) + \lambda \|\beta\|_K, \; e_i \text{ is the Euclidean basis vector}$$

  - Therefore, we will reach a *minimum RKHS-norm solution* defined by the *neural tangent kernel*, and small norm in RKHS generally suggests *smoothness*.
  - Note that the above results are *qualitative* and often don't reflect what happens in practice.

The kernel regime of NN

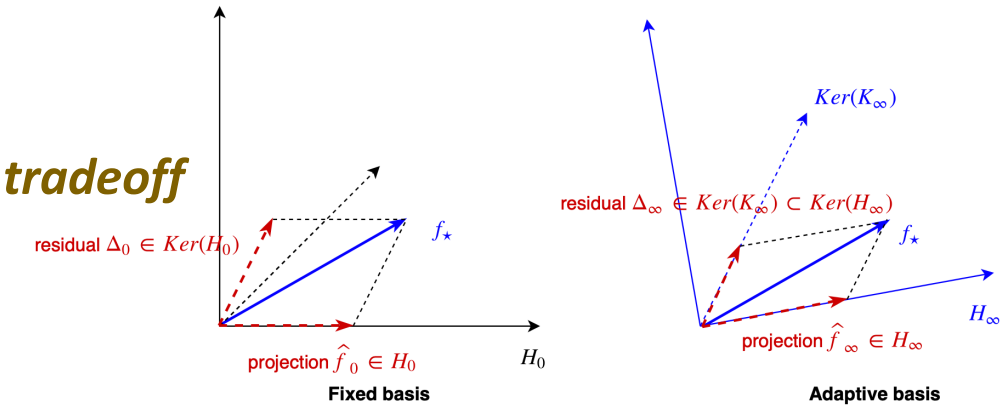
- What is the major gap between the NTK arugments and practice?
  - The NTK argument holds under *infinite-wide* setting and trained with *infitesemal learning rate* (second-order bias diminishes)
  - It resembles a *random-feature model* (e.g. the representations follows particular random initialization) with one trainable linear layer
  - In practice, (hidden) representations prior to the last linear layer can be updated during training
- Adaptive kernel (feature) learning of NN
  - The extra expressive capacity of NN creats a *decomposition* of the kernel space
  - One corresponds to the *NTK component*, the other is a *data-adaptive component*
  - Corresponds to decomsing the predictor into a *regular*

    prediction component (that generalizes well) and a

    interpolating component, they give ***different bias-variance tradeoff***

[GJK21]

$$\hat{f} = \hat{f}_{\text{pred}} + \hat{f}_{\text{interp}}$$



The role of model structure

- Containing more smooth decision boundaries that interpolates the data
  - IR data are extremely noisy
  - Interpolating IR dat often require both appropriate inductive bias and larger capacity
  - Smoother interpolation -> less sensitive to individual samples (thus better stability and generalization)
- Making GD suffering less from vanishing/exploding gradient (Jacobian more well-conditioned)
  - *Residual connection* – adding identity matrix
  - *Batch normalization* – similar scale across layers
  - *Dot product fusion* – spectral norm as regularization

Stability and memorization

- Recall that the *stability* of a learning algorithm also suggests *generalization*
  - NN can almost achieve interpolation even if we assign random label to a sample
  - Although directly deriving the stability of NN is challenging, empirically we know they are stable
  - Even with seemingly outlier samples from under-represented group, NN can fit those perfectly – is it a result of $\hat{f}_{\text{pred}}$ or $\hat{f}_{\text{interp}}$?
  - Why do we care? Data in IR is often a *mixture of individual distributions*, some users are well-represented, some are under-represented.
  - If samples from under-represented users are fitted because $\hat{f}_{\text{interp}}$ exploits ***supurious relationship*** (e.g. ***memorizing*** them using the model's extra capacity), then the testing performance will be bad on those users (***unfairness***)

[FZ20]

Memorization of NN and under-investigated fairness

- We've seen before that *over-parameterization* and ***interpolation*** are crucial to the success of NN – *defined by generalizing to the same population (distribution) seen by the training data*
  - ***Memorizing*** the under-represented subpopulations seems inevitable, judging both by the means and the goal
  - Does importance weighting address this issue?
- Understanding ***importance weighting*** for deep learning
  - When data is inseparable, importance weighting helps the training risk emphasizing the different sub-populations, e.g. with training data collected from P and we target at Q, the generalization bound is dominated by

[XYR21]

$$\mathcal{O}(\frac{D(P,Q)\mathcal{C}(\mathcal{F})}{\sqrt{n}}), \text{ where } D(P,Q) \text{ is the discrepancy between } P \text{ and } Q$$

  - Upon achieving interpolation, however, the convergence to the minimum-norm max-margin solution is *not altered by reweighting*!

$$\| \frac{\theta^{(t)}}{\|\theta^{(t)}\|} - \frac{\hat{\theta}}{\|\hat{\theta}\|} \| = o(\frac{\log n}{\log t})$$

From classification to learning-to-rank

- What we learned so far also applies to ranking problem?
  - Prediction with full knowledge about the distribution?
  - Generalization wrt. the decision rule? The rate?
- The ***ranking problem***
  - Given two pairs of $(X, Y), (X', Y')$, define $Z = \frac{Y - Y'}{2}$ -- in practice, we only care about its sign
  - A ***ranking rule*** is: $r : \mathcal{X} \times \mathcal{X} \to \{-1, 1\}$, and its performance is measured by the ***ranking risk***

$$R(r) = p(Z \cdot r(X, X') < 0)$$

  - If the whole distribution is known, easy to verify that the optimal rule also follows the max-a-poteriori principle

$$r^*(x, x') = 2\mathbb{1}\left[p(Z > 0 \,|\, x, x') \geq p(Z < 0 \,|\, x, x')\right] - 1$$

  - Now we move on to the ***empirical ranking risk***

$$R_n(r) = \frac{1}{n(n-1)} \sum_{i \neq j} \mathbb{1}[Z_{i,j} \cdot r(X_i, X_j) < 0]$$

From classification to learning-to-rank

- Ranking, scoring, and the bipartite problem
  - Would like to reduce ranking to scoring so we can unleash our zoo of ML methods
  - It is possible when the joint distribution is such that there exists f^* such that:

$$r^*(x, x') = 1 \text{ if and only if } f^*(x) \geq f^*(x')$$

  - So the risk becomes: $R(f) := 2\mathbb{1}[f(x) \geq f(x')] - 1$
  - A particular interesting case is $Y\ in\ \{-1, 1\}$, because we can show that

$$\mathrm{AUC}(f) = p(s(X) \geq s(X') \,|\, Y = 1, Y' = -1) = 1 - \frac{1}{2p(1-p)} R(f),\ p = p(Y = 1)$$

  - Hence minimizing ranking loss is ***optimizing the AUC***
- ***U-statistics*** and *improved generalization bound*
  - The ranking loss is a ***degree-2 U-statistics*** -- *minimum variance* among all unbiased estimators.
  - What we encountered so far are all based on *Hoeffding and Mcdiarmid's* inequality based on first-order statistics. There is also the ***Bernstein-type*** bound

$$R(f) - R_n(f) \leq \mathcal{O}\left(\sqrt{\frac{V_n(f)\log(\mathcal{C}(\mathcal{F})/\delta)}{n}}\right), \text{ with probability at leat } 1 - \delta$$

From classification to learning-to-rank

- The ***generalization bound*** for ranking loss is generally: $\mathcal{O}\left(\left(\frac{\mathcal{C}(\mathcal{F})\log(n/\delta)}{n}\right)^{1/(2-\alpha)}\right)$ [CLV08]
  - Seems like an improvement to standard supervised learning,
    if finding *appropriate formulation* that satisfies $\alpha > 0$
  - But it still comes to finding a surrogate loss and do optimization
- RANKBOOST – using convex surrogate loss function
  - Use such as *exponential function*, *logistic cost*, or *hinge loss*, e.g.

$$R(f) = \mathbb{E}\exp(-\mathrm{sgn}(Z)\cdot f(X, X'))$$

  - The structure of the risk is now more *supervised-learning alike*
  - Indeed, the generalization error bound now becomes a standard one

$$R(f) - R_n(f) \leq \mathcal{O}\left(\frac{\mathcal{C}_n(\mathcal{F})}{\sqrt{n}} + \sqrt{\frac{\log 1/\delta}{n}}\right) \qquad \text{[RCM05]}$$

From classification to learning-to-rank

- When using metrics other than *AUC*, the previous approach do not directly optimize the ranking metrics (e.g. *AP, NDCG*)
  - The main difficulty for directly optimizing ranking metrics is that they are *piecewise constant*
  - Use some smooth approximation to approximate ranking?
  - Given K numbers $f_1, \ldots, f_K$, assign a probability for each of them to being the largest (we will see this *softmax trick* again): $p_i := \frac{\exp(f_i/\eta)}{\sum_{j=1}^{K} \exp(f_i/\eta)}$ [CW10]
  - The risk function under ranking metrics becomes smooth under this formulation, and η affect the Lipschitzness -- standard generalization bound apply : $\mathcal{O}\left(K^2/(\eta\sqrt{n})\right)$
  - The error due to the approximation is: $\mathcal{O}\left(\exp\left(-\min_{i\neq j}(f(x_i)-f(x_j))^2/\eta\right)\right)$

  tradeoff for smoothness and generalization

- The unexamined ***ranking consistency***
  - Being consistent in the risk-minimization sense ≠ giving optimal ranking under infinite data!
  - NP-hard for ranking consistency? Another tradeoff ... [DMJ10]

Domain challenges using IR feedback data

- Perhaps the biggest domain challenge of using IR feedback data for learning is the ***missingness / sparsity*** of feedback
  - Limited slots to show recommendation or respond to queries
  - If we view the feedback as ***supervised signals*** (other views will be discussed later), then handling the missingness is critical
  - If we use feedback for ***data exploratory / analysis tasks***, sparsity will be a major blocker
- Revisiting some of the classical ideas in IR
  - ***Transductive SVM***: low-density separation
  - ***Collaborative filtering***: clustered user interest
  - ***Latent semantic indexing / matrix factorization***: distributional hypothesis (entity with similar distribution should have similar representation), shared low-rank hidden structure
  - ***PageRank / TF-IDF***: numerical statistics on the importance of an entity
  - …

Handling unlabeledness

- *Transduction* and *semi-supervised* learning
  - Both aims to achieve good performance on $(x_{n+1}, \ldots, x_{n+m})$ based on $\{(x_i, y_i)\}_{i=1}^n$
  - Supervised learning aim at: $x_{n+1} \sim P_X$, P_X unknown
  - Does knowing the testing samples in advance make a difference?

    -- if P_X, $P_{Y|X}$ are related, and $\{x_i\}_{i=1}^{n+m}$ provides a better estimation of P_X than $\{x_i\}_{i=1}^n$
  - Semi-supervised learning focus on finding good decision rule $f : \mathcal{X} \to \{-1, 1\}$

    Transductive learning focus on finding a set of good predictions: $\{\hat{y}_i\}_{i=n+1}^{n+m}$
- Provable guarantee? Hardly any without assumptions ...
  - Just say what if X is cause of Y, and P_X is generated independently from $P_{Y|X}$
- What realistic assumptions lead to the success of some classical IR methods?
  - P_X has a ***smooth density***, and decision boundary lies in the ***low-density region***
  - The optimal decision boundary on $P_{X_{\text{train}}}$ and $P_{X_{\text{test}}}$ does not shift much, so ***pseudo-labelling*** may work well

Handling unlabeledness

- The distributional or data geometry assumption by semi-supervised learning all imply some degree of similarity between $\{x_i\}_{i=1}^{n+m}$ and $\{x_i\}_{i=1}^{n}$
  - What if they are naturally dissimilar due to the selection process?

    -- samples with certain property are less likely to be labelled

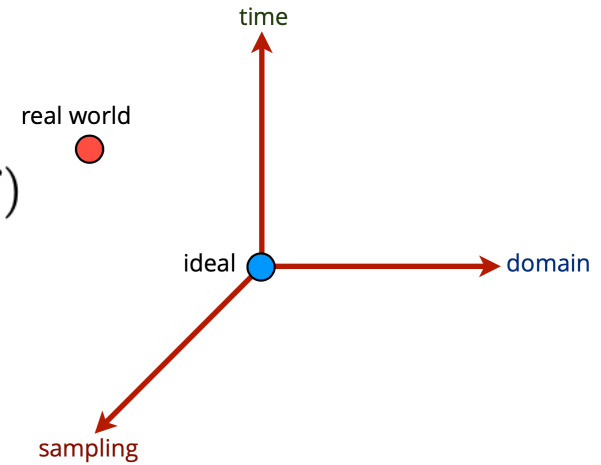
    -- the testing sample we are interested in are under the influence of new system

    -- ...
- Domain adaptation: train $\hat{f}_P$ on *source domain* P, and achieving small risk $R_Q(\hat{f}_P)$ on target domain Q
  - The generalization error bound usually depends on two terms:

    (a). *some (hypothesis-driven) domain discrepancy:* $D_{\mathcal{F}}(P, Q)$

    (b). *the smallest joint error that can be achieved*: $\min_{f \in \mathcal{F}} R_P(f) + R_Q(f)$
  - Can add *domain-discrepancy regularization* to make (a) small,

    but (b) will depend on the overlapping, i.e. $\text{supp}(Q) \subset \text{supp}(P)$ [JSR19]



Handling sparsity*

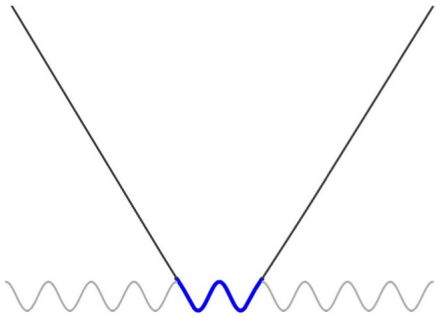
- Intuition behind using ***representation learning*** or ***graph neural network*** (GNN) to address *sparsity*:
  - The collected feedback themselves reveal some *structural similarity* among samples
  - The loss function + model structure of representation learning and GNN *implicitly characterizes those similarity* (e.g. pointwise mutual information, Weisfeiler-Lehman Distance, shortest-path metrics)
  - When over-parameterized (e.g. by layering up just like standard NN) and optimized by GD, we may again enter the *implicit regularization* regime
  - The prediction of the models are ***interpolation results***
  - The learnt representation correspond to the mapping of the ***feature-learning kernel***

*: The background and set up are somewhat tedious, so we defer the detailed discussions to our online material

Debias, extrapolation

- Relative feedback are less prone to bias than point-wise feedback, but:
  - Immune to all of *selection bias, exposure bias, popularity bias, position bias*, ... ?
  - Can possibly address all the bias in learnt patterns?
  - Does it provide a scientific way to quantify the effect of bias?
  - What about cases not represented in the collected data?

- Beyond the *support* of collected data:
  - We studied so far i.i.d generalization, which is intrinsically *interpolation*
  - *"... It takes all sorts to make a world ..."* -- interpolation doesn't satisfy all
  - Do not wish overparameterized models to *extrapolate* smartly – they behave like *linear functions* outside the support (recall from *NTK theory*).



[XZL+20]

- It's time to consider how to unleash the power of *intervention:*
  - A way to exchange plausible assumption for plausible conclusion.

Summary of Part 1

- Understanding what makes overparameterized NN work:
  - **Expressivity**: Generating smoother decision boundaries that *interpolate the data*
  - **Optimization**: Allowing GD to go beyond interpolation point -> bringing *implicit regularization*
  - **Generalization**: implicit regularization + interpolation -> optimum solution with *smaller capacity*
- Making NN work ≠ good performance for IR tasks
  - *Memorization* of NN -> less *fairness* for *under-represented group*
  - *Importance weighting* may not achieve the expected effect
- Some extra complications for ranking problems
  - What we learnt in classification mostly carry to ranking problems
  - For effective optimization, must suffer the tradeoff between *smoother loss* and *bias*
  - Consistency of risk minimization ≠ consistency of ranking
- IR domain challenges deserve more attention
  - Modern ML solutions often suffer from them, not magically solving them
  - The merits of introducing domain knowledge to solve the impossible