Recommender Systems
Learning to Rank

Professor Robin Burke
Spring 2019

Thanks to Alexandros Karatzoglou, Linas Baltrunas, Yue Shi for some materials

Outline

- Learning to rank
- TFMAP algorithm
- BPR
- Maybe non-accuracy metrics

+ Review: Learning to Rank

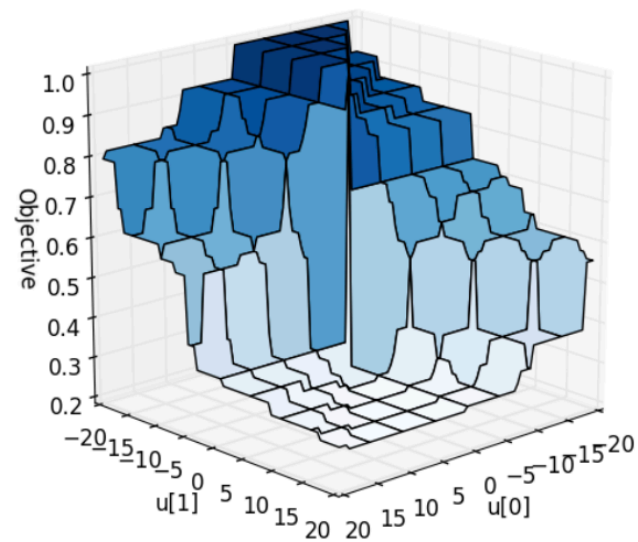
- Fundamental question for factorization models
  - Why are we optimizing $||R-\check{R}||^2$
  - If what we are really trying to do is get good lists
- Maybe we should optimize ranking directly

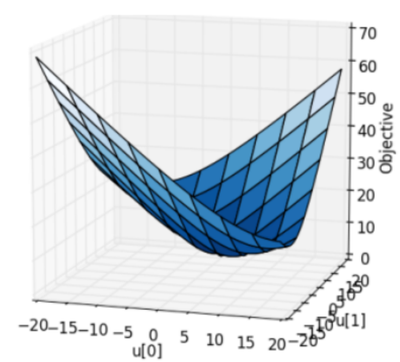
Learning to rank is harder than learning to optimize rating error because:

- A. Minimizing rating error produces great results and it is hard to do better
- B. All we get from users is ratings not rankings
- C. Ranking metrics are not continuous functions so gradient-based optimization cannot be used on them

Example: 1 latent factor dim 2

- Change the values of the factor
- Get different MAP values
- Compare with equivalent for RMSE
- Cannot apply gradient descent





Decompose the objective

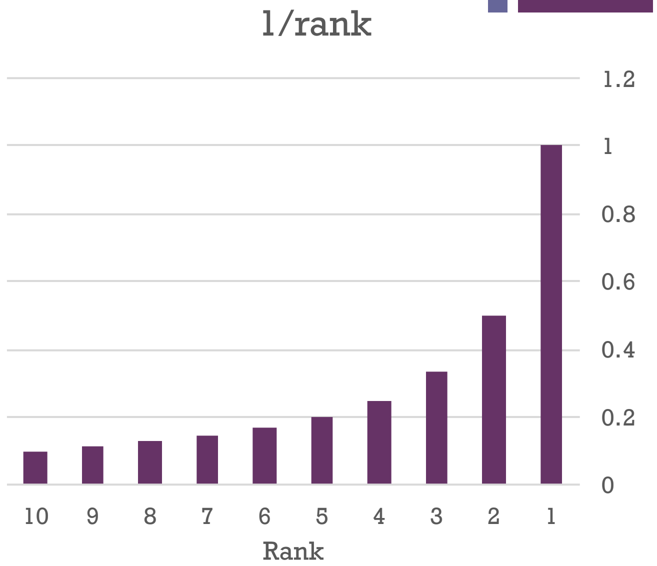
- Assume unary data
  - y_{ui} = 1 if the item i is relevant to user u
  - s_{ui} = the rank of item i
  - $\mathcal{J}$ is the indicator function 1 if s_{uj} <= s_{ui}

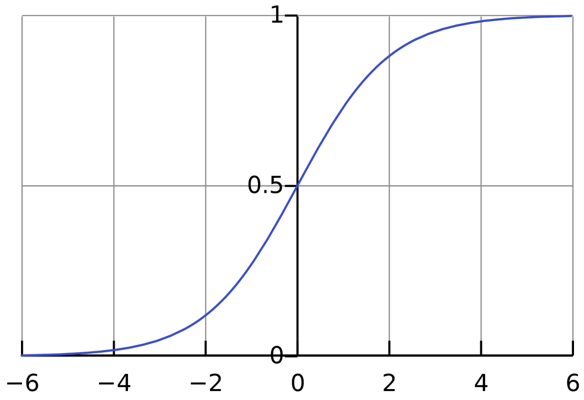
- $$MAP_u = \frac{1}{\sum_{i=1}^{N} y_{ui}} \sum_{i=1}^{N} \frac{y_{ui}}{s_{ui}} \sum_{j=1}^{N} y_{uj} \mathcal{J}(s_{uj} \leq s_{ui})$$

- Discontinuous because of the indicator function and $1/s_{ui}$

Getting continuity

- Goal: a continuous function of the predicted rating p
- Continuous version of $1/s_{ui}$
- As the score of an item goes up
  - Its 1/rank will also go up
  - But the score has no inherent limits and the 1/rank has a minimum and maximum
- Good candidate = logistic function
  - $g_1 = 1/(1+e^{-p})$





Getting continuity, part 2

- Continuous version of $I(s_{mj} <= s_{mi})$
- s_j will have lower rank than s_i
  - if s_j has a higher predicted rating p_j, than s_i (p_i)
- This is like a step function that depends on the difference between p_j and p_i
- Again logistic function is useful
  - $g_2 = 1/(1 + e^{-(pj-pi)})$

Intuition

- $1/(1 + e^{-(pj-pi)})$
- Item i scores higher than item j
  - $p_i > p_j$ mean $p_j - p_i < 0$
  - $-(p_j - p_i) > 0$
  - $1/(1+e^5)$ close to 0
    - "Down" side of the step function
- Item i scores lower than item j
  - $-(p_j - p_i) < 0$
  - $1/(1+e^{-5})$ close to 1
    - "Up" side of the step function
- Logistic approximates the step function

What is the logistic of the difference between two scores $1/(1 + e^{-(pj-pi)})$ if the scores are the same?

- A. ½
- B. 1
- C. 0
- D. undefined
- E. 1/e

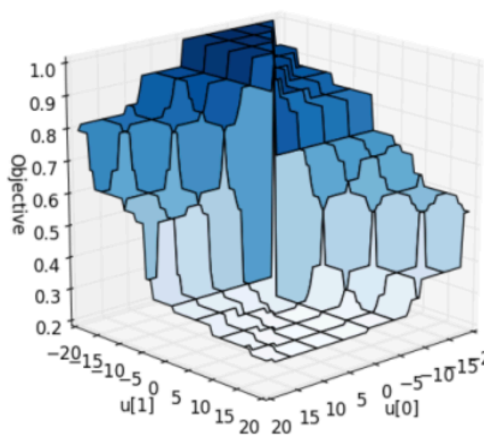
New objective

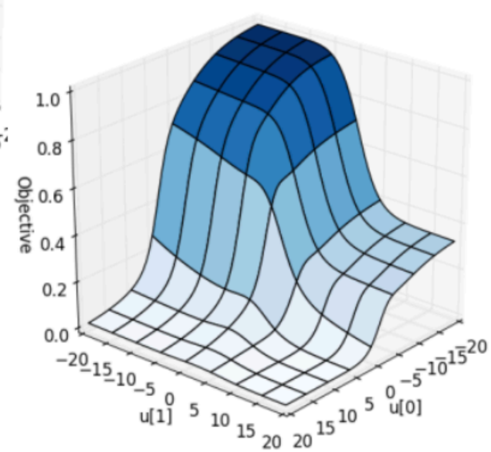
Logistic of i's rating:
How near the top?

- $MAP_u = \frac{1}{\sum_{i=1}^{N} y_{ui}} \sum_{i=1}^{N} y_{ui} g_1(p(i)) \sum_{j=1}^{N} y_{uj} g_2(p(i), p(j))$

Logistic of the score difference:
Is i ranked above j?

- Where p(i) is the predicted rating of item i





Now what?

- Let $p(i) = \check{r}_{ui} = uv^T$
  - For some UV^T factorization of this loss
- Now I can calculate the gradient of this loss function
  - for any given set of factors
  - for each known rating
- Use gradient descent as before

Caution

- Approximation is not always good
- Reason 1
  - Factorization is low-rank
  - May fail to capture important distinctions if the number of factors is too small
- Reason 2
  - Possible for the function approximations to give results not consistent with the approximated expressions
  - A > B and B > C but C > A

General strategy

- Take the ranking-oriented objective O
  - Inherently discontinuous because ranking creates discrete answer sets
- Find some continuous approximation C that is a function of the predicted rating
- Use C to create a continuous loss function over R
- Use factorization to create a prediction function that optimizes for C

Basic approaches

- *Point-wise approach*
  - *f(user,item) ⇒ score*
  - *Convert ranking to regression, classification, ordinal regression, etc.*
- Pair-wise approach
  - f(user, item1, item2) ⇒ score
  - Convert ranking to pair-wise classification
    - Is item1 ranked higher than item2?
- List-wise approach
  - f(user, item1, item2, …, itemn) ⇒ score
  - Direct optimization of the ranking metric over the list

Examples

- MAP optimization (originally proposed for context-aware recommendation as TFMAP)
  - List-wise approach – optimizing MAP
- BPR and RankALS
  - Pair-wise approaches
  - Different optimization criteria