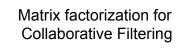


Matrix factorization motivation

- Discover/use latent factors – attributes, topics, features
- Factor matrices to uncover latent factors
- Don't know what latent factors represent – can conjecture

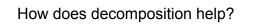


- Give ratings matrix R: M users X N items
 R has holes- R_{ij} with no value
- Want to fill in holes => predict ratings
- Idea: decompose R:

R=P*Q[⊤]

- P is M X f; Q is N X f
- f dimensions are latent factors
- no interpretation but can add one

must choose f



2

- estimate P and Q, leaving no holes
- get estimate of R as R_f = PQ^T
 R_f has holes of R filled in
- Several methods for estimation, e.g. – Gradient descent
 - Stochastic gradient descent
 - Koren et al. Matrix Factorization Techniques for Recommender Systems, IEEE Computer, Aug 2009
 - Least squares based calculations
 Bell et al Modeling Relat'ships at Multiple Scales to Improve Accuracy of Large Recom. Sys., KDD Aug 2007.

Optimization

• Minimize least squares error:

err(P,Q) is defined as

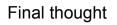
$$\sum_{(u,i) \text{ in } K} (R_{(u,i)} - (PQ^T)_{(u,i)})^2$$

for K the set of (u,i) for which $\mathsf{R}_{(u,i)}$ has a value

Simple Step: Gradient Descent
Minimize for one element change:
(choose one element of P or one element of Q to vary, say P(r,g) (PQ²)(r,j) = (∑_{k,k±3}) P(r,k) * Q₁(k, j) + x * Q₁(k, g))
ent(P,Q) becomes equation with one unknown
look at only terms involving x
et sum over j for which R_{(r,j}) has a value of:
(R_{(r,j}) - (PQ^T)(r,j)² = (R_{(r,j}) - (∑_{k,k±3}) P(r,k) * Q₁(k, g)) - x * Q₁(k, g)²)²
take derivative wrt x, set to 0, solve



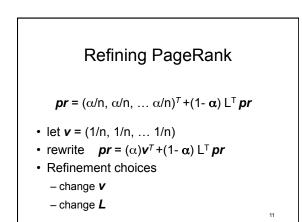
- · Looked at several techniques to modify search
- · explicit user feedback
- user behavior: history
 user history
 - crowd history
- collaborative history: "people like you"
- · role of social networks
 - general analysis
 - relationships
- · models of recommender systems

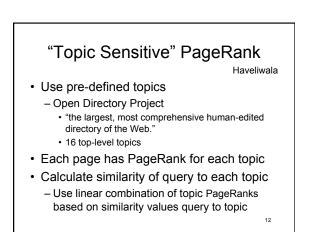


All techniques we've seen behavior or topic oriented

What about links? What about PageRank?

10





Personalized PageRank Kamvar et. al.

- Random leaps are biased by personal interests change \pmb{v}
- Combined with use of block structure to make more efficient:
 - Divide Web graph into blocks (clusters)
 - Use high-level domains (e.g. princeton.edu)
 - Calc. local PageRank within each block
 - Collapse each block into 1 node new graph
 Weighted edges between nodes
 - Calc. PageRank with biased leaps for block structure

13

Weight local PageRanks with block PageRank
Use to initialize power calculation