Correspondence

October 28, 2015

Contents

1 Correspondence Winter 2015

1.1 1/8/15 BH-AT

Updates:
- Found a useful book ”Graph Mining: Laws, Tools & Case Studies”
- Few chapters on tensor (chpts 14 & 15)
- Free online checkout via WWU

Notes:
- Group meeting presentation on recommender systems?
- Multi-task DNN?
- How many shared layers? How many different layers?
- Tensor DNN?
- Factored tensor?

Next steps:
- BH read scoop paper
- Identify our novelty
- Discuss a first plan
- Discuss features of items

Thanks,
Brian

1.2 1/15/15 BH-AT

Updates:
- Found a paper that takes a hybrid approach to recommender systems

Notes:
- Novelty and diversity both desirable features of a recommender system, formalized in a paper cited in the hybrid paper
- Recommender system for classes? Clubs? Activities? Music? Books? Movies? Restaurants?
- Aaron will update doodle poll

Thanks,
Brian

1.3 1/17/15 BH-AT

Thanks. Slides for today will go in

/home/hutch_research/doc/20150114

The first person there needs to create the directory; e.g.,

/home/hutch_research/doc/20150114

AND change permission so that other group members can write to it:

chmod g+wx /home/hutch_research/doc/20150114

Or, if someone else has already created the directory for the week, you can just add to it.

That last portion of the path is a YYYYMMDD time stamp, of course, so it will change each week.

Thanks, Brian

1.4 1/20/15 BH-AT

Updates:
- Issue checking out repo due to directory permissions

Notes:
- Changed meeting time, 3-3:20 TF
- Recommender system for:
- Classes
- Professors
- Clubs
- Activities
- Music
- Books
- Movies
- Restaurants

Next steps:
- Keep reading
- Preparation steps:

1) Identify the sets of entities (e.g. sets of classes, of
restaurants, etc.)

2) Find some seed recommendations (generic, un-catered)
- Restaurants from Yelp?
- Classes and/or profs by publicly available course evals?
Rate professors?
- Movies or books from publicly available datasets?
Heuristics like best-sellers.
3) [Optional] Content associated with each entity
- Design our website/app
- How can we get implicit feedback?
- What form of explicit feedback? (Up/Down, Stars)
- How to normalize user feedback?

- BH email Brittany Schade, Martin Granier(?), 49x instructors(?)

Way down the road:
- Corpus paper?

Thanks,
Brian

1.5 1/22/15 AT-BH

Below is the website for the large annual recsys conference.

http://recsys.acm.org/

I will comb over this sight to get a feel for current and past trends in recommender systems research. Also, there is access to some data from their yearly recommender system challenges.

Aaron

1.6 1/23/15 BH-AT

Updates:
- Found article on evaluating recommender systems

To discuss:
- Toolkits
- Python: scipy, numpy,
- Subversion

Next steps:
- Find a multi-domain corpus and/or single domain corpora in the domains of interest
- Send BH the paper on evaluating recommender systems
- Send BH the links to the ”three main” toolkits
- Go over paper(s) in greater detail
- Dig into recommender system combination (hybrid systems?)
- Preparation steps:
1) Identify the sets of entities (e.g. sets of classes, of restaurants, etc.)
2) Find some seed recommendations (generic, un-catered)
- Restaurants from Yelp?
- Classes and/or profs by publicly available course evals? Rate professors?
- Movies or books from publicly available datasets? Heuristics like best-sellers.
3) [Optional] Content associated with each entity
- Design our website/app
- How can we get implicit feedback?
- What form of explicit feedback? (Up/Down, Stars)
- How to normalize user feedback?
- BH ask registrar or CS support for sanitized registration data, class lists and descriptions

Notes:
- Recommender system for:
- Classes*
- Professors(?)
- Clubs*
- Activities
- Hobbies*
- Music
- Books
- Movies
- Restaurants*

Thanks,
Brian

1.7 1/23/15 AT-BH

Brian,

Here are the links to software tools that were mentioned in papers that I was going to check out.

RiVal: recommender evaluation toolkit (open source):

http://rival.recommenders.net/

MyMediaLite: Recommender System Library (open source):

http://www.mymedialite.net/

Lenskit: open-source tools for recommender systems

http://lenskit.org/

Apache mahout: scalable machine learning library (free?):

http://mahout.apache.org/

I've found several papers on recommendation evaluation but the one attached is from a recent recsys conference. I'll put

some stuff in the repo when I'm back at campus.

Aaron

Brian,
Below is the short comment about data that the scoop paper makes.

4.1 Amazon Data

Amazon 1 is the most famous e-business website to sell diverse
products, such as books, DVDs, shoes, etc. The dataset [10] was
crawled from Amazon website and it contains 1,555,170 users and
1-5 scaled ratings over 548,552 different products covering four
domains: 393,558 books, 103,144 music CDs, 19,828 DVDs and
26,132 VHS video tapes. Obviously, the users preferences are
dependent across these domains, so it is very suitable to test CDCF
algorithms over this dataset.

Data Preparation: In this experiment, we selected Book and Music
CD as the target domain to evaluate respectively. We filtered out
users who have rated at least 50 books or 30 music CDs so that

there are enough observations to be split in various proportions of training and testing data for our evaluation. Finally, 2,505 users were selected, and in addition we retrieved all items rated by these users in these four domains and set aside top K rated items for each domain respectively. Table 1 shows the statistics of the data for evaluation. Then, we constructed rating matrices over filtered out data for each domain.

Sparse Data Case: To simulate the sparse data problem, we constructed two sparse training sets, TR 20 and TR 75 , by respectively holding out 80 % and 25% data from the target domain Book, i.e. the remaining data of target domain for training is 20% and 75%. The hold-out data servers as ground truth for testing. Likewise, we also construct two other training sets TR 20 and TR 75 when choosing Music as the target domain. Unacquainted World Case: We randomly select half users and hold out all their data from target domain to simulate the unacquainted world phenominon. The training set used for this case is denoted as TR uw .

I think we can build a data set that matches the red number in users.

”We filtered out users who have rated at least 50 books or 30 music CDs...”

This quote makes me wonder:
How do recommendation systems that employ a scale, fare when the number of users with meaningful feedback,
in proportion to the number of users who use the system and don’t contribute, varies?
This probably depends on how much implicit feedback the system incorporates.

A couple of things:
If it will take too much effort to gather descriptions for items lets forgoe that avenue for now, but if it doesn’t take
too much effort to get the descriptions with the listings and user stats we should get them looking out for future research projects.
Also, we should solicit and list reviews in the recommender system. I think people like this option and we could gather good data this way. If we don’t have fodder for word bags in the first place, if we collect reviews
we’ll have fodder for word bags in the future.

Also attached, a paper, The Dynamics of Viral Marketing, which was cited with a reference to
the data. Maybe, the Australasions got ahold of
the data that backed this very interesting 40 something page paper. Maybe Leskovec et. al, would let us use this data too.

Aaron

1.8 1/27/15 AT-AS

Annika,

I am a graduate student in the Computer Science department at WWU beginning research. The professor I am collaborating with, Brian Hutchinson, expressed interest in developing an algorithm for a cross-domain recommendation system. Recommender systems are the devices that companies like Netflix, Amazon, Google and other corporate and non-profit entities use to filter the wide variety of options available and display preferable content to consumers. I ventured to Brian the idea of a Western recommendation system that could recommend classes, clubs, professors, events, restaurants and possibly books and movies. Professor Hutchinson is on board with this idea, and so I am contacting you to determine the level of student interest in such a system. I would like to talk to someone from the AS about the reception and interest of students concerning such a system. Basically, there would be a website/mobile app which students anonymously logged in to that would provide recommendations for items in the above categories. On the research end we would track the preferences of users (anonymous of course) in the system and use this information to provide recommendations to students. I think this would be a great resource to students. The system could help students find resources and opportunities relevant to them that otherwise they might not have noticed.

No other university to date has implemented such a recommender system. So, in this respect the idea is cutting edge. Recommendor systems which cater to the interests of a local community are not typically employed. However, at the beginning of developing such a system I think it is important to gage how valuable students feel such a resource could be and to get their input on what would be important to them in a recommender system. I am willing to put together a talk about recommender systems, and the possibility for designing one catered to the needs and preferences of Western students. Afterwords, we could have a forum and some kind of vote about if students would like a recommendor system as part of their Western experience, and what features they would appreciate in such a system.

I think that this is an interesting project and avenue of research that may be beneficial to our community of students at Western. Please let me know what your thoughts are and direct me (or forward my email) to any other people I should talk to concerning the project.

Thanks for the time, Aaron Tuor

1.9 1/27/15 AS-AT

Aaron,

Thank you so much for reaching out to us about this, I admittedly dont know much about recommender systems but I would love to talk to you about how you see it helping to support the Western Community and what think our role in helping you could be. I would love to set up a time to meet with you this week if possible, what does your schedule look like?

Cheers,

Josie Ellison Communications Director Associated Students

1.10 1/27/15 BH-AT

Updates:
- Found excellent data Amazon reviews dataset
- AS wants to help with the recommender system

Next steps:
- Look over data and meta data
- See what our possibilities are
- Decide what to focus on
- Find the number of domains each user has reviews in
- Compute the domain entropy for each user:

1) Normalize the counts of reviews in each domain
- Both: read this http://i.stanford.edu/ julian/pdfs/recsys13.pdf
- Decide if we want to do the user interface aspect
- Arrange two meetings:
- Schade
- AS

Thanks,
Brian

1.11 1/31/15 AT-BH

Updates:

Notes:

- http://snap.stanford.edu/data/web-Amazon-links.html

Discuss:
- Looking over data, computing things

Next steps:
- Create a document that has four tab-delimited columns:
userid numberReviews numberDomains domainEntropy
- See your notes for the details
- Both: read this http://i.stanford.edu/ julian/pdfs/recsys13.pdf
- Cancel Schade meeting?
- Cancel AS meeting

Thanks,
Brian

1.12 2/3/15 BH-AT

Updates: - Did the analysis of the distribution of items and categories; a few wrinkles - Items can be in multiple categories - Can use his breakdown of items - 20-40 minutes to run - Some items not in categories - Meeting scheduled Wednesday 2/11 1-2pm in Fine Arts 109

Discussion: - Issue: items can be in multiple categories, some ideas: - Create a graph where nodes are categories, and edges are weighted by number of cross-listed - Create a bipartite graph linking items to categories - Create an item-category matrix, do SVD, pick max category

Next steps: - Fix entropy - Make a graph, nodes are categories, weights on edges are the number of users who have posted reivews in both categories - Keep investigating small vs large file issues

Thanks, Brian

1.13 2/13/15 BH-AT

Updates: - Made even fancier pictures than before

Next steps: - New graphs: - Add top K edges, normalized - Each category has edges to top 5 co-categories, with unnormalized line width - Radial arrangement, show *all* edges, but line width is log of unnormalized counts - Use some of the standard recommender system tools to train single-domain systems - Try training this on top 10 (all?) of our categories - Measure performance - Try using all as domain? Everything is an item. - Create a version of all data with no unknown users

Down the road: - Develop our tensor or NN models?

Thanks, Brian

1.14 2/13/15 BH-HR

A few upcoming items to have on your radar...

- Unless we just started our project, you should plan to present a poster at WWU Scholar's Week* (May 11-15th). Even if we just started, we still might want to present. Let's discuss it. - Grad students should plan to present a poster at the WWU Graduate Conference** (May 17th).

These venues are great opportunities to get useful feedback on your work and raise its visibility. You can list your participation in these events on your resumes/CVs.

Also, upcoming paper plans that I have on my radar are: - DP submit latest T-DSN work to IEEE Signal Processing Letters - CN submit TIMIT TELLAR work to Interspeech 2015. Deadline 3/17. - KA+SK submit paper to JQAS or NESSIS once wavelet-style work is done. Deadline 6/15.

If we've discussed upcoming papers that I don't have listed here, please remind me.

Thanks, Brian

http://www.wwu.edu/scholars/index.shtml
http://www.wwu.edu/gradschool/gradconfsubmissions.shtml

1.15 3/19/15 AT-BH

Brian, The guy in this video is describing a university recommender system like we were discussing.

http://www.youtube.com/watch?v=kHodCjWP1Us

Aaron

Brian,

We were talking about reading this attached paper earlier. Having finished reading the paper I don't think we should evaluate our Mahout systems using the Mahout evaluation tools. We need an evaluation tool that can produce evaluations independent of the framework. We could code our own evaluations but I don't think it's necessary at this point to start from scratch. The RiVal recommender evaluation tool the attached paper is plugging was the tool used in evaluating the Netflix prize competition. I don't know what kind of metric Netflix was using but I am sure the metric used is documented somewhere that I could find with a little searching. From my understanding RiVal acts as a wrapper for a recommender system that provides a testing framework which can evaluate under different strategies with full access to the parameters of a given evaluation strategy. Mahout performs worse than the other two frameworks in the attached paper using an out of framework evaluation (RiVal) and a comparison of cross-framework evaluations. This may be due to the movieLens data set being an integral dataset in the development of Lenskit. The occurence of "lens" in the two names is not a coincidence.

Fair travel,

Aaron

1.16 3/26/15 AT-BH

Brian,

I just finished reading J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. , the paper linked to the dataset website. This paper is really good, and also relevant to both groups working with the Amazon dataset. I would be interested in discussing this paper with the other group if they feel like reading it. I understood most of the paper and am with a little preparation I could explain the basic latent factor model which is based on ratings alone.

Aaron

1.17 3/26/15 BH-HR

I've come up with a plan for individual meetings and group meetings that should work based on the doodle poll answers. See the schedules here:

https://hutchresearch.cs.wwu.edu/index.php/Individual_meetings

Please double check and make sure that these will work for your schedule. Let me know ASAP if there's a conflict. I'm planning to start meetings next week (first week of the quarter).

Thanks, Brian

1.18 3/28/15 BH-TS

If you're on this email, then I have included you in the rank, sparsity and tensor subgroup. As the group meetings page shows, this subgroup will meet each Wednesday from 4-5pm. Due to scheduling conflicts the first two weeks of the quarter, our first meeting won't happen until April 15th, when we'll discuss the following paper:

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7050385

You should be able to download the pdf if you access the page from any WWU machine. It's very unlikely we'll finish it, but let's try to make a good jump on it.

Please read it carefully ahead of time and come prepared to discuss your understanding of it (including asking questions about parts that aren't clear to you). You should probably budget a few hours of prep time for reading / googling terms / etc. At that meeting we will set a plan for future meetings.

Thanks, Brian

1.19 3/31/15 BH-AT

Updates: - Created expt - 3 diff rec systems - 28x3 combinations - One per subtopics - Didn't evaluate any of them yet - Printed out top 5 items predicted for first 1k users - Needed to map alphanumeric IDs to numeric ideas

Discuss: - Watches

Next steps: - Dig deeper into the actual algorithms being used - May want to read at some point:
http://research.microsoft.com/en-us/um/people/gzweig/Pubs/SLT2012.pdf

- Look into Rival javadocs (http://rival.recommenders.net/site/0.1/apidocs/) - BH read evaluation paper (recsys-benchmarking) - BH read combined text and matrix paper

Thanks, Brian

2 Correspondence Spring 2015

2.1 4/3/2015 BH-AT

Updates: - Found good collab filtering paper

Discuss: - Matrix factorization approaches to collaborative filtering

Notes: - Important characteristics: - Scalable - Add new user without complete retraining - Idea: - RNN transduction from product to review, per user & item - Sparse + low rank exponential model where users/items are factors

Next steps: - Dig deeper into the actual algorithms being used - May want to read at some point: http://research.microsoft.com/en-us/um/people/gzweig/Pubs/SLT2012.pdf - Look into Rival javadocs (http://rival.recommenders.net/site/0.1/apidocs/) - BH read evaluation paper (recsys-benchmarking) - BH read combined text and matrix paper

Thanks, Brian

2.2 4/7/2015 BH-AT

Updates:

Next steps: - Joint train: - Ratings-based recommender - Classifier that maps from product metadata (including description) to rating - Then have a per-user system to map from product metadata to rating - Generalizes very easily to new product (zero ratings needed) - Doesn't generalize to new users - Some questions: 1) What has been done related to this? - Look into the Related Work cited in McAuley & Leskovec 2) What form should our metadata to rating model take - Idea: log-linear models, where each user's weights are a column in a low rank matrix - Sigmoid output? - Columns for diff. users are linearly dependent 3) Jointly solve for matrices A, B, C in a combined optimization problem - Search for prior work - (Simultaneous OR joint OR collective) matrix factorization AND recommender system - See if there are any links to/from this paper: http://www.cs.cmu.edu/~ajit/pubs/Singh2008.pdf

Thanks, Brian

2.3 4/7/2015 AT-BH

Brian,

The idea you proposed today seems promising. In the literature I have seen CF algorithms give good

recommendations for new users, but none that give good recommendations for new items. Content based

systems can give good recommendations for new items, and in a sense if we think of product descriptions

as clues to the content of an item, what you are proposing is

a blend of content based and CF methods. I think it would be most attractive to design a model that

can accommodate new users and new items. For our MF component we should look into Assymetric SVD

(section 4, Koren, Factorization Meets the Neighborhood), which gives comparable (better)

results to standard matrix factorization approaches, but has the added benefit of good recommendations

for new users.

Aaron

2.4 4/8/2015 AT-BH

Brian,

Nothing new in ”Solving the cold start problem using product related content”. This is a student’s final project for some class wherein she uses the HFT model developed in the McAuley, Leskovec paper with the product descriptions from the Amazon dataset instead of customer reviews as a text resource. Also she collected some outside resource text related to items as well for movies. The HFT model does about the same using descriptions instead of reviews for the amazon dataset.

Aaron

2.5 4/10/2015 BH-AT

Updates: - Lit review for simultaneous matrix factorization

Notes: - Famous LDA paper: http://machinelearning.wustl.edu/mlpapers/paper_files/BleiNJ03.pdf

Next steps: - Make poster - Column 1: - Problem overview - Recommender system overview - Basic matrix factorization model - Column 2: Cold Start New Users - State the problem - Discuss existing solutions (should include figures/illustration, can include results) - Column 3: Cold Start New Items - State the problem - Discuss existing solutions (should include figures/illustration, can include results) - Column 4: Proposed solution to both cold start problems - State the problem - Our proposed model - Data - Joint train: - Ratings-based recommender - Classifier that maps from product metadata (including description) to rating - Asymmetric SVD for generalization to users - Then have a per-user system to map from product metadata to rating - Generalizes very easily to new product (zero ratings needed) - Doesn’t generalize to new users - Some questions: 1) What has been done related to this? - Look into the Related Work cited in McAuley & Leskovec 2) What form should our metadata to rating model take - Idea: log-linear models, where each user’s weights are a column in a low rank matrix - Sigmoid output? - Columns for diff. users are linearly dependent 3) Jointly solve for matrices A, B, C in a combined optimization problem - Search for prior work - Online training of recommendations - The cold start problem - Enrollment/registration/initiation for recommender systems (could also be under name of ”active learning”)

Possible contributions of the research: - Rigorous assessment on cold start context

Thanks, Brian

2.6 4/11/2015 AT-BH

Brian,

The hottest trend in addressing the cold start user problem is to make decision trees of small depth from a ratings matrix and use the trees and

a set of queries to estimate a user profile. Then you can use the item profiles from the matrix factorization and dot product them with the estimated user profile to estimate ratings. I'm looking into decision tree methods, and I haven't seen any based on extracting eigenvalues from

a UV decomposition derived from matrix factorization. Once a complete matrix of ratings is estimated via latent factor we could SVD this matrix

and find items which participate most heavily in latent factors in order to build an efficient decision tree. This type of model for building trees seems

simple enough to be scalable. Maybe we could talk about this on Monday or Tuesday.

Aaron

2.7 4/12/2015 AT-BH

Brian,

We don't have this kind of data but we could make a model by randomly generating demographic information according to some distribution.

This looks nice because we could generate item and user profiles for cold start. Probably it's too simple. Picture attached.

Aaron

2.8 4/17/2015 BH-AT

Notes: - AT has leads on data

Next steps: - AT send readings and videos to KM - AT port information about data over to KM - Amazon - MovieLens - AT port information about baseline systems over to KM - KM run baseline experiments - Split the data into train, dev and test - Cut unknown user(s) from data - Could be included in item global baseline - Just a single category (probably books) - All data thrown in together - Use mahout three SVD-based approaches - Tune parameters - Port info to wiki - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks

Poster notes: - can't -¿ cannot - Use consistent form for the two cold starts $\sum_u \sum_i$

Later steps: - Nail down our new model

Thanks, Brian

2.9 4/17/2015 AT-BH

Brian,

We discussed the possibility today of pruning the dataset for a denser observation set. For the baseline project that Katy is getting started on I think it would be interesting to vary the sparseness of data for several baseline models and see how robust they are with respect to sparseness. For the amazon dataset this would entail eliminating portions of the data and seeing how much the accuracy improves. For an artificial dataset generated according to some distribution this would entail eliminating entries in the dataset and seeing how much the accuracy diminishes. The best metrics for recommender systems are currently a debated topic, and I haven't seen anything in the literature I've surveyed that addresses this aspect. If this hasn't been done before, I think something like this would be an interesting and productive experiment for Katy.

Aaron

2.10 4/21/2015 BH-AT,KM

Updates: - AT and KM met to discuss relevant background - Basic methods - Intro to matrix factorization approach - Presented poster on Saturday

Dicuss: - Objective

Next steps: - Work on the gradient w.rt. U, P, and W for our objective - See matrix cookbook - AT port info about data to KM - Amazon - MovieLens - AT port info about baseline systems over to KM - Look into Rival, see if it does what we need - KM run baseline experiments - Split the data into train, dev and test - Cut unknown user(s) from data - Could be included in item global baseline - Just a single category (probably books) - All data thrown in together - Use mahout three SVD-based approaches - Tune parameters - KM Port info to wiki

- Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks

Thanks, Brian

2.11 4/28/2015 BH-AT,KM

Updates: - Started reading paper (involves boosting)

Next steps: - KM Port info to wiki - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT port info about data to KM - Amazon - MovieLens - AT port info about baseline systems over to KM - Look into Rival, see if it does what we need - AT and KM arrange weekly meeting time - KM run baseline experiments - Split the data into train, dev and test - Cut unknown user(s) from data - Could be included in item global baseline - Just a single category (probably books) - All data thrown in together - Use mahout three SVD-based approaches - Tune parameters - AT derive gradient - AT send me a list of possible papers for reading group

Thanks, Brian

2.12 5/1/2015 BH-AT

Aaron, I spent some time working on the gradients for our joint factorization model. Attached are my notes. The first page defines my variables (sort of) and the last page has the gradients (the circle dot denotes element-wise multiply). The rest of can probably be ignored – I only include it for context. I'm about 90 percent sure on these gradients... sure enough to try to code it up and see if our results make sense. I didn't include global, user or item biases, and I didn't include any regularization terms. The latter will be easy to add, the former shouldn't be too bad, either.

You need to produce the following matrices (where N is the number of users, M is the number of items, and V is the number of word types in our vocabulary):

- Omega (NxM). This is the ratings mask. Contains a 1 in (i,j) if user i rated item j, and 0 otherwise. - R (NxM). This is the ratings matrix. (i,j) contains the rating user i gave item j. - Q (VxM). This is the word-document matrix. The (i,j) entry should be 1 if word i appears in the document associated with item j, and 0 otherwise.

Once we have these three matrices, I think our algorithm will only take 10 lines of Matlab code to implement, and should be fairly fast since all of the operations are vectorized and Matlab will parallelize them locally.

Thanks, Brian

2.13 5/2/2015 AT-BH

Brian,

I like this idea, initializing P and U with an SVD of R, and I have heard it mentioned before without any explanation of the details. I thought about this for a bit

and here is my first idea for making this sensible. We can't just SVD R and use these results for initialization because R is not complete. We have to make a decision about how to fill in the missing entries. I think there are two reasonable approaches. We could find optimal global attributes (user bias, rating mean and item bias) and subtract these from the entries in R to make R prime, in which case we can make the missing entries zero. Then we could do an SVD on R prime, intiialize P and U as you have described, and have an objective function on R prime which doesn't care about the global attributes since we already guessed them. Or else we could find the optimal global attributes and use these to estimate ratings which we place in the missing entries of R to make a different R prime. Then we do an SVD on this R prime and use this to initialize P and U as you have described, and our objective function is concerned with the original R (in this case it doesn't matter if we use R or R prime since Omega will filter). We should find a sensible heuristic for initializing our P and U for hopefully quicker convergence. Having said this, in the papers I have read on MF approaches to recommending, if anyone ever mentions how they initialize, they claim random initialization. Random may make sense if you are optimizing global predictions at the same time as user-item predictions.

Aaron

2.14 5/3/2015 BH-AT

I think it's worth considering these initialization options, but a good rule is to start with the simplest approach and work outward from there. If people claim to do random initialization, then we should try that. The main advantage of the SVD is that it is simple but more intelligent. If we see a big win there, we should consider some of the variants you propose.

2.15 5/3/2015 AT-BH

Brian,

When I was first came to you about research I was entertaining the idea of modelling vagueness in natural language with a sigmoid function.

Logistic regression would be a good tool to learn these functions (associated with a word or concept and a context) from data (if we can find the right sort).

We could make a very simple artificially intelligent machine with vague logic for the semantics of simple sentence.

I will look into this if we have a break from the recommender research.

2.16 5/5/2015 AT-BH

Brian,

I finally ran across the paper which was the catalyst for a lawsuit against IMDb, and probably led to the lawsuit against Netflix for the Netflix Prize Competition. IMDb now only allows restricted access to random samplings from their database of user profiles.

Title: Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset)

Probably neither of us will have time to read this but it looks interesting.

Also, the movielens metadata has titles in IMDb format, and IMDb has for each movie, a collection of plot descriptions made by users. So, with some amount of work we can get a text dataset for the movielens data.

There is a pre 2008 (pre-aforementioned paper) movielens 100K dataset that has user profiles (age, zip code, and a few other things) which if we want we could use for a 3-way collective factorization model that I mentioned. We could factor a one-hot sparse matrix of user demographic features (every zip code, every whatever other discrete category, every sex, ... etc for a row). I have seen a recent paper (I think the one you sent me and we discussed briefly because of the cool evaluation technique) where the authors use this dataset to address user cold start.

2.17 5/5/2015 BH-AT,KM

Aaron

Updates: - Met with Katy on Friday - Processed data into category csvs, removed top two users (Unknown and some pro reviewer) - Katy will focus on recommender system for Movies amazon category

Discuss: - Sparse matrix

Next steps: - AT to focus on our model, basic MF, collective embeddings - KM to focus on SVD, SVD++, FunkSVD - Each of the dev or test subsets has 5 % of the ratings - Download MovieLens and get it into right format - All sizes - AT add new figure and our gradients to the poster - AT tune R on dev set - AT process text - AT start coding in Matlab - matload/matsave for dense matrices - smatload/smatsave for sparsematrices - KM Port info to wiki - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT use old text normalization script (from SLP), and then use export PERL5LIB=$PERL5LIB:/home/hutch_research/lib /home/hutch_research/bin/generateVocab.pl

Matlab ascii format (each line is one non-zero element):

rowindex1 colindex1 value1 rowindex2 colindex2 value2 ...

in matlab... X = load('myFile.txt'); X = spconvert(X);

2.18 5/11/2015 BH-AT

I'll take a look at the poster this morning. The script is

Thanks, Brian

2.19 5/12/2015 AT-BH

Brian,

For our model, the reason I don't think the upper factorization from our picture will add anything to the model is because omega is applied to both factorizations. What if we replaced R on the bottom with UP^T. The we have a U on the left of both sides of the bottom factorization so they cancel. The bottom factorization would then be $P^T = W^TQ$. We could then not apply omega to the bottom factorization and use all the information from Q. This is basically the Local Collective Embeddings model rewritten which is unfortunate but I do think this may still be a superior base factorization to fancy up, as no solution of an equation is necessary online. Attached is a poor drawing that will make the preceding comments clear.

2.20 5/12/2015 AT-BH

Brian,

Here's a rough idea. The reviews of users for products in the Amazon dataset could be put into a user-item-term tensor, and we could use a collective factorization with the item-user ratings matrix for a model. I'm not sure how this relates to your idea elaborating our model using tensors and neural networks but it seems similar from what I gathered at last Friday's meeting.

Aaron

2.21 5/12/2015 BH-AT

I like the direction here. What does the (i,j,k)th element of the user-item-tensor denote/signify? My neural net idea was slightly different, I was thinking of joint factorization of a user-item matrix with one of the following two:

- A user-hidden matrix, where "hidden" is a deep representation of the document (this would be the last weight layer in a neural net to predict a scalar rating value). - A user-term-hidden matrix, where we map from user and term vector to a hidden representation (which is used to predict a scalar rating value).

But I'm interested to discuss your idea more.

Thanks, Brian

2.22 5/13/2015 AT-BH

I was thinking at (i,j,k) index there is a 1 if word k occurs in the review of item j by user i. When thinking about this tensor I was thinking, what do you get when you toss out information? If we sum over the term dimension to collapse this tensor into a matrix (and turn all non-zero entries into 1's) we have a matrix like the user-document matrix from LCE (1 if user commented on news article zero

otherwise). On the other hand if we sum over the user dimension to collapse this tensor into a matrix (and turn all non-zero entries into 1's) we have a document term matrix where a document is the collection of reviews associated with an item.

Maybe we could do something interesting with a tensor like this.

Aaron

2.23 5/14/2015 AT-BH

Brian,

We've done a lot of research and thinking on the recommender subject. I think we both have a good understanding on several

newish extensions of the MF approach. Right now we are on the cusp of implementing some models which are variations of joint factorization.

While we are fleshing out ideas for a more sophisticated model, the results from these simpler models will be fodder for the idea mill.

My paper idea is this: Exploring Cold Start Recommending Through Matrix Factorization

-with the data sets we've gathered and are making we can explore some interesting questions

-How do models fair on sparse item data?

-How do models fair on sparse user data?

-How do models fair on sparser data in general?

-How does the user/item ratio affect the importance you place the the above metrics?

-How to achieve an optimum balance and temper the affect of sparsity on the success of a system?

-What is the best model to achieve a minimum slope on a function from sparsity to accuracy?

If you think this might be a good topic I can search around to see if anyone has directly addressed research on sparsity in recommender systems or some analoguous

branch of study.

2.24 5/15/2015 BH-AT,KM

Update: - Script is running, doesn't finish in 2 hours - Regenerating data since it seems like some reviews were lost - Looked over mahout code

Notes: - Basic idea for using screen is: 1) Type screen (this creates a new screen session) Start something running now... 2) To detach type "Ctrl+a, d" 3) To reattach, one of two options: screen -x (if there's only one screen session) screen -r SOMENAME (if there are multiple)

Next steps: - Redownload tar file(s) - AT dig up links - KM focus on making new map - KM run mahout baseline - AT to focus on our model, basic MF, collective embeddings - KM to focus on SVD, SVD++, FunkSVD - Is there a way to save our models? - AT tune R on dev set - AT start coding in Matlab - matload/matsave for dense matrices - smatload/smatsave for sparsematrices - KM Port info to - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks

Thanks, Brian

2.25 5/19/2015 BH-AT,KM

Updates: - Made 5 dictionaries for Amazon dataset - Ran dataset split on Amazon and MovieLens - But will replace this later with Matlab - Discussed Mahout - Global idx to category idx map done

Discuss: - DONE: Vocabs - DONE: New model

Notes: - Most important things in Matlab: save('filenamegoeshere','variablenamegoeshere'); save('my.mat','X'); load('filenamegoeshere'); X * Y; X .* Y; X'; sum(sum(X)); X(3,:); X(:,5); size(X,1); size(X,2); eye(5); zeros(5,10); 1 - X; x x;

Next steps: - KM: First experiments will use Mahout's auto dev split, then using our testing scripts for evaluation in the end - Fix Mahout random seed so its comparable with itself - AT finish data prep script (use hash tables) - Discard if multiple matches or no match - AT to focus on our model, basic MF, collective embeddings - KM to focus on SVD, SVD++, FunkSVD - Is there a way to save our models? - AT start coding in Matlab - matload/matsave for dense matrices - smatload/smatsave for sparsematrices - KM Port info to - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT: Try vocab with cutoff of 5

2.26 5/22/2015 BH-AT,KM

Updates: - Training SVD baselines - The few are the best - Tuning K (rank), lambda (L2 regularization coeff), # iterations - Wiki info added

Discuss: - Results

Next steps: - KM: - Try fewer iteration, see how it affects RMSE - Try SVD++, FunkSVD, others? - Figure out how to save models trained - Set up ssh keypair - Keep up the good work porting info - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT: - Give KM relevant data - Matlab path: /usr/central/bin/matlab - Implement our models in Matlab

Thanks, Brian

2.27 5/23/2015 BH-AT

Brian,

Here is a link to a collection of matlab scripts for methods related to Latent Semantic Analysis.

http://lear.inrialpes.fr/ verbeek/software.php

Aaron

2.28 5/26/2015 BH-AT,KM

Updates: - Start coding in Matlab - Loading data, making mask, etc.

Next steps: - AT+KM collaborating on evaluating baseline - KM runs baselines, outputs predictions - AT provides matlab evaluation code - Visualize RMSE as a function of both user and item sparsity - KM: - Try fewer iteration, see how it affects RMSE - Try SVD++, FunkSVD, others? - Figure out how to save models trained - Set up ssh keypair https://support.cs.wwu.edu/index.php/Setting_up_SSH_keys_in_Unix_Linux - Keep up the good work porting info https://hutchresearch.cs.wwu.edu/index.php?title=Recommendation_System - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT: - Give KM relevant data - Matlab path: /usr/central/bin/matlab - Implement our models in Matlab

2.29 5/29/2015 BH-AT,KM

Updates: - Start coding in Matlab - Loading data, making mask, etc.

Next steps: - AT+KM collaborating on evaluating baseline - KM runs baselines, outputs predictions - AT provides matlab evaluation code - Visualize RMSE as a function of both user and item sparsity - KM: - Try fewer iteration, see how it affects RMSE - Try SVD++, FunkSVD, others? - Figure out how to save models trained - Set up ssh keypair https://support.cs.wwu.edu/index.php/Setting_up_SSH_keys_in_Unix_Linux - Keep up the good work porting info https://hutchresearch.cs.wwu.edu/index.php?title=Recommendation_System - Relevant papers - Background info - Books, videos, etc. - Tools - Tips & Tricks - AT: - Give KM relevant data - Matlab path: /usr/central/bin/matlab - Implement our models in Matlab

Thanks, Brian

2.30 6/5/2015 BH-AT,KM

Updates: - Swamped, no progress - Getting better at relevant languages

Next steps: - KM fill in AT on everything done - Send him your script - If KM gets script from AT, can include it in the pipeline - AT - Finish lit review - Port lit to wiki - Eval script (finished this weekend?) - AT+KM collaborating on evaluating baseline - AT provides matlab evaluation code - KM runs baselines, outputs predictions?? - Try SVD++, FunkSVD, others? - Implement our models in Matlab - Visualize RMSE as a function of both user and item sparsity

Later steps: - Figure out KM 49x project (recommender related?)

Thanks, Brian

2.31 9/7/2015 BH-AT

Agenda: - CHECK Discuss progress/updates - Discuss notes - Discuss ways a 49x team could support this work - Set up a rough timeline

Updates: - Implemented basic MF model - Implemented evaluation script - During evaluation, things were going wrong, so AT wrote up data notes - Haven't gone back and finished experiments - Wrote up detailed matrix factorization models

Notes on notes: - The notes look great - I haven't gone over the equations with a fine-toothed comb - Regularize the bias? - Over mu notation? - Use equation numbers for everything - BH note to self: check equations w/ U and P extended with 1's and a's and b's - doc j is a collection of texts

Next steps: - Meet again on Monday? - Try to find papers that compare stochastic vs minibatch vs batch - Try to find MF for collaborative filtering survey - Set up a recommender systems elective - Start in Fall?

Timeline / gameplan Papers: 1) Conference paper with a novel technique [cold start emphasis?] (AT+BH or AT+BH+49x) 2) Survey journal paper (AT+BH+49x) 3) Toolkit conference paper (49x+AT+BH) Evaluations: - Stochastic vs MB vs batch - Different models - Different datasets - Coldstart conditions - Time and space complexity - Survey journal paper - Key attributes: - Accessible - Unifying - Thorough evaluation - Some novel twist - Tie together numerous methods, MF, joint, tensor - Trade-offs in time and space - Batch vs stochastic vs minibatch - Throw in non-linearities - Figure out what the best journal in rec sys is

How to integrate with 49x: - Can AT take 60x this year?? - Students implement toolkit in Python and run thorough evaluations - App or web interface?

AT academic plan: - Fall 2015: 352 RecSys elective ** - Win 2016: 367 511 - Spr 2016: 322 601 ** - Sum 2016: EL3 602 - Fall 2016: 603 OS - Winter 2017: EL4 EL5 - Spring 2017: EL5 Prog Lang

2.32 10/2/2015 BH-AT

FYI it appears I never hit Send on this Friday...

Updates: - Gradient models that struggled on Watches attributed to sparsity, does better on Netflix - 0.94 RMSE on Netflix - 1000 vs 100 epochs for stochastic, although may be slower, expts still running - KM did some lit review - Found some papers, use the NN to fill in entries - Neural network fusion (same domain)

Possible directions: - 49x - Neural networks - Cross-domain - AT 60x - Cold start - Tensors? - Neural networks - RNN timestamps

Notes: - Wiki: hutchresearch.cs.wwu.edu - MovieLens might have user metadata (user cold-start???)

Next steps: - KM add NN papers to wiki - Explore the computational network toolkit - Look at Leskovec papers for data filtering/pre-processing ideas - AT: re-carve up recommender paper to bypass time modeling - Lit review: cross domain - Lit review: NN recommenders - Implement a basic NN: - In torch - In Matlab

Long term plans: - SIGIR (Jan/Feb) or RecSys (April)

Thanks, Brian

2.33 9/14/2015 BH-AT

Agenda: - Response to Brittany Schade? - 49x students contribution - PARTIAL Discuss papers and survey paper contributions - LATER Discuss code

49x students focus on: - Emphasis on toolkit and running experiments - Neural networks jointly trained with MF (LR in output space, backpropagate through NN) - Compare to probabilistic matrix factorization - Something from Sergey Feldman?? - Learn a parametric sigmoid?? - Implement an recommender system app - Multi-domain - BH: document neural network models (tensor ones, rank regularized output, etc.)

Notes: - An app and a toolkit tie together - Look at code?

Updates: - Sent 60x email schedule to Phil Nelson (no response yet) - Implemented basic model and local collective embeddings - In Matlab using AT's unified structure - Their Matlab implementation is 2x fast

Next steps: - Report dev results on three sets: 1) Just dev stars (known items) 2) Just dev columns (unknown items) 3) 1 union 2 - For tuning - Simple models only use set 1 for early stopping - Coldstart models use set 3 - AT identify reasonable chunks of Koren Bell paper for the reading group - Think about how to use tensors as the unifying framework

Recommender paper reading plan: - 7 Wednesdays 1) Advances in Collaborative Filtering (40 pages, 4 meetings) 2) Unified View of Matrix Factorization Models (16 pages, 2 meetings) - Restricted Boltzmann paper? (1 meeting) - Algorithms for Non-Negative Matrix Factorization (7 pages, 1 meeting)

Thanks, Brian

2.34 9/29/2015 BH-recGroup

Updates: - Wiki not working for AT - SVN issues for AT - Data mostly ready (Amazon, MovieLens, Netflix)

Tools: - Torch - Matlab (write our own) - Computational network toolkit (https://cntk.codeplex.com/)

Possible directions: - 49x - Neural networks - Cross-domain - AT 60x - Cold start - Tensors? - Neural networks - RNN timestamps

Next steps: - BH: Get AT access to wiki - Lit review: neural networks and recommender systems - Come up with a plan for preprocessing Amazon data - Explore the computational network toolkit

Long term plans: - SIGIR (Jan/Feb) or RecSys (April)

Thanks, Brian

2.35 10/3/2015 AT-recGroup

Group, Today, Elijah and I made a matlab script to put our .sparse format ratings data into the basic format described in the meeting for neural networks. I fixed the svn issue and my stuff is updated now so you can poke around if you want. At updates on Monday I'll give a brief description of the tools and documentation I've created that may facilitate your research. I think working with the data should be pretty painless at this point using the scripts and preprocessed data that have been created so far. If people want to meet sometime next week, Monday and Tuesday are not good for me. My availability for other days can be found at aarontuor.xyz. There is schedule link on the side bar. See y'all next week.

Aaron

Here are the spotty notes from today,

Elijah:

raw data path: /home/hutch$_research/data/amazon_reviews_category/textRaw/all.txt$

python scripts path documentation (good notes on Amazon data): /home/hutch$_research/projects/recomm$

used amazonSparseMaker.py on all.txt to make: /home/hutch$_research/projects/recommender_systems/aa$

used sparsemap.py to map amazon ids to integer ids (made all$_mapped.sparse$).

aaron's matlab scripts: /home/hutch$_research/projects/recommender_systems/aaron_copy/expts/mf/and$
$previous_path > /functions/$

you can use these from the folder you are working in by calling addpath(¡path¿) in matlab;

use twohotmaker.m on fullamazon.mat (in /home/hutch$_research/projects/recommender_systems/aaron_co$

Aaron

2.36 10/8/2015 BH-recGroup

Updates: - Found some very relevant literature: 1) Scooped by Xiaodong He 2) Cross-domain pairs 3) General cross-domain - Basic baseline models implemented and optimized - Multiplicative updates don't work (diverging) - KM/AS implementing NN in Matlab - EB converting data to torch format (almost ready) - CNTK has lots of features and documentation - Not installed / running yet - Not user friendly - Use one 7-way split (70-5-5-5-5-5-5)

Next step: - Read the paper that probably scooped us Friday (and maybe again Monday)! http://research.microsoft.com/pubs/238334/frp1159-songA.pdf - AT get stochastic to work - AT implement three-matrix joint factorization for both cold-start problems

Brainstorm: - RNN over text feed hidden to DNN (also MovieLens user features) - Tensor and matrix and joint factorization

Thanks, Brian

2.37 10/9/2015 AT-recGroup

Group, –I started working on the basic nnet Brian described for CF. I talked to Katy about this and due to the learning curve of a new programming language we considered the possibility of their contingent giving me this task and focusing on learning to use the nnets toolkit from microsoft.

–I have sketches of cold start approaches for our data using the deep semantic neural net used as a baseline from the paper on Wed. If Brian can figure out some of the math for me I would like to implement these models in matlab.

–I have the local collective embeddings model working now. When I was working on multiplicative updates for basic MF I noticed a bug that was infecting all my models. Next thing, I will fix our alternative (hopefully superior) collective factorization model and package both for experiments.

–Multiplicative updates is up and running for basic MF (great convergence!) but bias is not supported as of yet.

–Basic MF is now packaged as one function that takes 7 arguments and returns a mfmodel object with data from training and characteristics of the model (hyperparameter values, update method, and trained factors). There is an argument which specifies an update method. The choices are stochastic and batch gradient descent and multiplicative updates. mfmodels have a function for plotting training history, and I will add a function for predictions.

–I can run an experiment for basic MF with a grid search, record all the best dev accuracies in a text file with hyperparameter specs, and save one or a few trained models that have the best dev accuracy. This way we can look at pictures of the training for the most promising models when the experiment is done and also see the overall trends in performance for combinations of hyperparameters. I'm just going to use batch for this as it seems to get comparable results to stochastic with a little better speed.

–I can draw some pictures of ideas to show everyone tomorrow for comparative models to train and test on our data. I am hoping Brian has an epiphany for something new after he sees these.

–I have some thoughts for progressing. I now have reciprocal designs (mostly unoriginal) for user/item cold start approaches using both MF and neural nets. I can implement all of these in matlab and test them on a level platform which is something out of the norm for state of the art methods. If I proceed in doing this we will have a wide array of baseline results which other group members can use for comparing against a new fancy model they implement. In doing this baseline stuff, and comparing neural nets against MF approaches on a level platform, I think we can learn a lot about trade offs between the different approaches. My guess is that neural nets are going to give better accuracy for each task but they will take more time and space. How ever it turns out, results like this on a level platform would be informative for the recommender systems field.

–I need suggestions on prioritizing my efforts.

About multiplicative updates: Multiplicative updates are essentially a scaling of the step size of the gradient which depends on the gradient itself during each update in gradient descent. Unlike traditional gradient descent there is no hyper parameter for step size which has to be small enough for convergence to a local minima. It has been demonstrated empirically many times that multiplicative updates have better convergence properties than normal gradient descent. There a couple of caveats here. The research on these updates focuses on the matrix factorization problem and in order to guarantee convergence you must (a) have a matrix for factorization whose entries all have the same sign and (b) know the signs of all the terms in the gradient. This method was popularized for non-negative matrix factorization but it could work just as well for non-positive matrix factorization. The method is so popular in part because the update rules are very simple. Given a parameter u the update rule is u = u* (abs(negative$_gradient_terms)/positive_gradient_terms$).

If I could get multiplicative updates going with bias, training the basic MF should go a lot faster. It is basically batch with the gradient step scaled for each iteration. The problem is that for the updates you have to know the sign of all the quantities in the gradient. However bias terms could be either positive or negative and if you constrain them to be one or the other you lose the advantage of having bias. I have an idea for fixing this issue. We can think of a bias term b$_iforitemiasb_i = bplus_i + bminus_i. Ifthetrainedb_ineedsbepositivethenthemagnitudeofbplus_iisgr$

See y'all tomorrow, Aaron

2.38 10/9/2015 AT-recGroup

Group, I tarred up the basic MF code with the processed amazon baby data. If you want to give it a spin here is what you do:

(1) download the file

(2) extract the file: I use tar -zxvf ¡tarfilename¿

(3) get in a matlab terminal or gui in the directory the files were untarred in.

(4) load the data: load coldBabytest/coldBaby.mat

(5) run mf.m (you'll get back an mfmodel object):

model = mf('batch', train, dev, item$_{d}ev_{c}old, user_{d}ev_{c}old, both_{d}ev_{c}old, 20, 500, 1, 0.001, 20, true, true$);

function signature: function model = mf(update, R, Dev, itemdevcold, userdevcold, bothdevcold, k, iterations, maxbadcount, alpha, lambda, verbose, bias) parameters: (1) for update you can choose 'batch', 'stoch', or 'mult'. (2-6) These parameters are sparse data matrices that got loaded into the workspace in (4). I made this geared towards cold start. (7) For k choose a positive integer. This is the number of latent factors. (8) For iterations choose a positive integer. This is the number of epochs the model will train for. (9) maxbadcount is turned off right now so it doesn't matter what integer you put in this field (10) alpha is the step size so it can be any positive real number (11) lambda is the regularization term for L2 so it can be any positive real number. (12) verbose is a boolean. in matlab booleans are literally true or false. (13) bias is a boolean the same as verbose. You always get poorer results with this model without bias for super sparse data.

(6) If you want to evaluate on the test sets you can do this: test set: model.predict(test); cold item test set: model.coldpredict(item$_{t}est_{c}old,'item'); coldusertestset : model.coldpredict(user_{t}est_{c}old,'use$
$model.coldpredict(both_{t}est_{c}old,'both')$;

If you want to get cold start pictures you can do this: model.plotcold(test, item$_{t}est_{c}old,'item'); ormodel.plot$

If you get a chance to look at my code please let me know if there are bugs or if there is anything I should optimize or clean up.

Aaron

2.39 10/20/2015 BH-recGroup

- No EB here

Agenda: - Updates - XH - Pick the reading group paper - Project direction - Roles - CNTK

Updates: - AT almost has multiplicative updates working with bias - Most of our baselines are implemented, some things are just slow

Note: - AT: look into Poblano (https://software.sandia.gov/trac/poblano) - Possible direction: - Joint matrix factorization and NN training? - Exploit cases where you don't have info about all items? All users? All ratings? - RNN to "digest" the documents - Other corpora: wiki edits, digg, reddit

Next steps: - Implement line search for stochastic gradient descent - Figure out the deep semantic gradient equations - Things we need: - Acquire new datasets (KV) - Processing new datasets (KV) - Work with CNTK (AS) - Work with Matlab (AT) - Work with Torch (EB) - Read up on methods to "digest" a sequence (KM) - https://cs.stanford.edu/ quocle/paragraph$_{v}ector.pdf(re$
$http://research.microsoft.com/pubs/257676/SentenceEmbedding1502.06922v2.pdf$

Thanks, Brian