|  |  |
| --- | --- |
| **For both MovieLens 20M and NFLX** | |
| **Parameters fixed** | **Parameters changed** |
| p=0.7; k=50; alpha=1 | lamb (1, 10, 100) |
| p=0.7; k=50; alpha=0.1 | lamb (1, 10, 100) |
| p=0.7; k=50; alpha=0.01 | lamb (1, 10, 100) |
| p=0.7; k=100; alpha=1 | lamb (1, 10, 100) |
| p=0.7; k=100; alpha=0.1 | lamb (1, 10, 100) |
| p=0.7; k=100; alpha=0.01 | lamb (1, 10, 100) |
| p=0.9; k=50; alpha=1 | lamb (1, 10, 100) |
| p=0.9; k=50; alpha=0.1 | lamb (1, 10, 100) |
| p=0.9; k=50; alpha=0.01 | lamb (1, 10, 100) |
| p=0.9; k=100; alpha=1 | lamb (1, 10, 100) |
| p=0.9; k=100; alpha=0.1 | lamb (1, 10, 100) |
| p=0.9; k=100; alpha=0.01 | lamb (1, 10, 100) |

**NFLX**

Parameters fixed Parameters changed

p=0.7, k=50, alpha=1 lamb (1, 10, 100) 96.3344%, 91.3271%, **61.0266**%

p=0.7, k=50, alpha=0.1 lamb (1, 10, 100) 97.9888%, 97.8130%, 91.3925%

p=0.7, k=50, alpha=0.01 lamb (1, 10, 100) 97.9878%, 97.8059%, 91.3563%

p=0.7, k=100, alpha=1 lamb (1, 10, 100) 98.3988%, 95.8885%, **80.2701**%

p=0.7, k=100, alpha=0.1 lamb (1, 10, 100) 98.9981%, 98.9496%, 95.9888%

p=0.7, k=100, alpha=0.01 lamb (1, 10, 100) 98.9975%, 98.9480%, 96.1034%

p=0.9, k=50, alpha=1 lamb (1, 10, 100) 95.8686%, 89.9838%, **56.2083**%

p=0.9, k=50, alpha=0.1 lamb (1, 10, 100) 97.9947%, 97.8990%, 91.8691%,

p=0.9, k=50, alpha=0.01 lamb (1, 10, 100) 97.9952%, 97.9076%, 92.3017%

p=0.9, k=100, alpha=1 lamb (1, 10, 100) 98.2607%, 95.1714%, **77.6233**%

p=0.9, k=100, alpha=0.1 lamb (1, 10, 100) 98.9992%, 98.9796%, 96.4929%

p=0.9, k=100, alpha=0.01 lamb (1, 10, 100) 98.9995%, 98.9805%, 96.5180%

MovieLens (float32 used . Delete it when NFLX is used)

p=0.7, k=50, alpha=1 lamb (1, 10, 100) 93.9395%, 90.2419%, **66.3825%,**

p=0.7, k=50, alpha=0.1 lamb (1, 10, 100) 97.9460%, 97.4862%, 92.3768%

p=0.7, k=50, alpha=0.01 lamb (1, 10, 100) 97.9464%, 97.4624%, 92.3967%

p=0.7, k=100, alpha=1 lamb (1, 10, 100) 97.1312%, 94.9307%, **82.0474%**

p=0.7, k=100, alpha=0.1 lamb (1, 10, 100) 98.9878%, 98.8174%, 96.2750%

p=0.7, k=100, alpha=0.01 lamb (1, 10, 100) 98.9873%, 98.8286%, 96.3212%

p=0.9, k=50, alpha=1 lamb (1, 10, 100) 93.7492%, 89.2140%, **63.4297%**

p=0.9, k=50, alpha=0.1 lamb (1, 10, 100) 97.9834%, 97.6863%, 92.7979%

p=0.9, k=50, alpha=0.01 lamb (1, 10, 100) 97.9792%, 97.6967%, 92.8476%

p=0.9, k=100, alpha=1 lamb (1, 10, 100) 96.9735%, 94.4133%, **80.2497%**

p=0.9, k=100, alpha=0.1 lamb (1, 10, 100) 98.9960%, 98.9113%, 96.6508%

p=0.9, k=100, alpha=0.01 lamb (1, 10, 100) 98.9960%, 98.9095%, 96.6926%

**Transfer-learning results: k=100 is bad, try k=50. (NFLX)**

**~~Explain how data was retreived from Oracle/~~**

**~~Mention movie plots used as features. Cleaning of plots + vocab construction out of corpus plots~~**

**~~Add description of datasets.~~**

**NFLX is good with k=50, Movielense with k=100 (when we print top10 from each topic, and check words)**

**~~Mention 5-fold cross validation~~**

**Estimate confidence interval with 10% subsampling.**

**~~Python challenges faced: how phi’s are stored 3D matrix.~~**

**Higher lamb 3-30%, but high lamb with high alpha less%. Test it (sparsity)**

* Two most common tasks in recommender systems are predicting the score the user might give for a product (the rating prediction task), and recommending a ranked list of most relevant items (the top-N recommendation task)
* [[Take definitions of “in-matrix” and “out-of-matrix’ from CTR paper]]
* Mention that we run topic modelling LDA on **plots** not other features.
* Discussion on my experiments while implementation on Python
* Put top 20 words for each topic from pure or CTMPs LDA
* Train Time + Machine Specs of Google Cloud (maybe make table on time spent for each hyperparameters)
* Where used “document”, where used “item”. Decide which one.
* Sparsity plots
* ~~Why we choose p {0.7, 0.9} and not 0.5? because 0.5 indicates uniform distribution, which was intention behind OPE.~~
* Include somewhere change of doc and user cycle compared to original paper [include in theory, or in experimental part]
* Compare p=0.7 with p=0.9, is higher better? [in BOPE its mentioned so]
* Python libraries used mostly numpy, pandas [+ numba jit]

Mention that 90% of movies from NFLX is also contained in MovieLens 20M. So, compare LDA results from both datasets, explore the common topics they learn.

**During Model Evaluation, put “Interpretable user profiles” somewhere.**

Refer Pavel thesis http://cbir.utia.cas.cz/homepage/projects/phd\_thesis/vacha\_phd.pdf

**Difference on recall&precision between our and original work may be because we use plots instead of tags/features.**

**Generally, talk about iter\_train, iter\_infer is set to max 100 during trainings.**

**Test set sample – 2000 users.**

**Add More**

Put gamma distribution explanation with formula, somewhere

Add Appendix

ADD connectivity between sections [e.g., between Probab. Models for RS and LDA

**References**

[1] <https://en.wikipedia.org/wiki/Topic_model>

[2] <https://www.tidytextmining.com/topicmodeling.html>

[3] <https://dl.acm.org/doi/pdf/10.5555/944919.944937>

[4] <https://stephentu.github.io/writeups/dirichlet-conjugate-prior.pdf>

[5] <https://people.eecs.berkeley.edu/~jordan/papers/variational-intro.pdf>

[6] Univariate Discrete Distributions, vol. 444 [[[[take from CTMP]]]]

[7] <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9138369>

[8] <https://arxiv.org/pdf/1512.03308.pdf>

[9] <https://www.scopus.com/record/display.uri?eid=2-s2.0-84982318199&origin=inward&txGid=117c9f14425c2abc105f8cd8ac63fa5f>

[10] <https://www.sciencedirect.com/topics/mathematics/digamma-function>

[11] Fully Sparse Topic Model (FSTM)

[12] <http://www.diva-portal.org/smash/get/diva2:1219240/FULLTEXT01.pdf>

[13] [https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-%5bNetflix%5d.pdf)

[14] <https://en.wikipedia.org/wiki/Collaborative_filtering>

[15] <https://link.springer.com/referenceworkentry/10.1007%2F978-3-642-04898-2_327>

[16] https://blog.evjang.com/2016/08/variational-bayes.html

[17] <https://dl.acm.org/doi/pdf/10.5555/1378245.1378272>

[18] <https://www.cs.toronto.edu/~amnih/papers/bpmf.pdf>

[19] <https://pubmed.ncbi.nlm.nih.gov/24467759/>

[20] <https://papers.nips.cc/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf>

[21] <https://www-users.cs.umn.edu/~baner029/papers/10/gpmf.pdf>

[22] <https://arxiv.org/pdf/1301.6705.pdf>

[23]<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.564.5299&rep=rep1&type=df>

[24] CTR

[25] CTPF

[26] LDA - <https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>

[27] OPE - https://arxiv.org/abs/1512.03308

[28] BOPE - <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9138369>

[29] Frank–Wolfe – take from BOPE

[30] Natasha2 – take from BOPE

[31] Stochastic Majorization-Minimization – take from BOPE

[32] Concave-Convex procedure – taken from BOPE

[33] Variational Bayesian – taken from BOPE

[34] CGS – taken from BOPE

[35] HAMCMC – taken from BOPE

[36] Berkeley Conjugate Priors - <https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter9.pdf>

[37] Gamma f - <https://en.wikipedia.org/wiki/Gamma_function>

[38] Variational Inference: A Review for Statisticians https://arxiv.org/pdf/1601.00670.pdf

[39] Variational Inference in Nonconjugate Models https://arxiv.org/pdf/1209.4360.pdf

[40] Closed-form solution <https://www.sciencedirect.com/topics/engineering/closed-form-solution>

[41] Closed-form expression <https://en.wikipedia.org/wiki/Closed-form_expression>

[42] mean-field VI https://www.cs.cmu.edu/~epxing/Class/10708-17/notes-17/10708-scribe-lecture13.pdf

[43] variational EM <https://chrischoy.github.io/research/Expectation-Maximization-and-Variational-Inference/>

[44] exponential family conditionals <https://www.cs.cmu.edu/~epxing/Class/10708-17/notes-17/10708-scribe-lecture13.pdf>

[45] exp fam connects to VI <https://people.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf>

[46] Blei-PPT-longer <https://www.eurandom.tue.nl/wp-content/uploads/2019/05/Blei_lectures.pdf>

[47] MovieLens 20M <https://grouplens.org/datasets/movielens/20m/>

[48] Lemma, Stemming <https://arxiv.org/pdf/1608.03995.pdf>

[49] <https://en.wikipedia.org/wiki/Netflix_Prize>

[50] Poisson Factorization https://arxiv.org/abs/1311.1704

[51] Modified LDAhttps://en.wikipedia.org/wiki/Latent\_Dirichlet\_allocation

[52] Variational Inference graph <https://rllabmcgill.github.io/COMP-652/lectures/lecture-17.pdf>

[53] CTMP

[54] learning beta https://link.springer.com/content/pdf/10.1007/978-3-642-33460-3\_37.pdf

https://people.eecs.berkeley.edu/~jordan/papers/variational-intro.pdf -

In particular, they make a link between this lower bound and parameter estimation via the EM algorithm. ALSO ADD variation EM explanation:::

***in-items*** – rated by users

***cold-items*** – not rated by any user

– where is the number of correct items that appear in Top-M recommendation for user u, is the number items that user u had rated positive.

– where is the number of correct items that appear in Top-M recommendation for user u, is the Top-M number.

**in-matrix prediction**

TOP-M recommendations only contain in-items (noncold).

**Recall:**

* is number of in-items that appear in Top-M which are actually like by user u.
* is number of in-items that user u had rated positive. They are all in-items, because if user u had rated them, they are automatically in-items because they have rating information.

**Precision**

* is number of in-items that appear in Top-M which are actually like by user u.
* is the Top-M number.

**out-of-matrix predicition**

TOP-M recommendations contains both in-items and cold-items.

**Recall:**

* is always in-items that appear in Top-M which are actually like by user u + cold-items that appear in Top-M ???
* is same as in-matrix prediction.

**Precision**

* is always in-items that appear in Top-M which are actually like by user u + cold-items that appear in Top-M ???
* is the Top-M number.

Question 1) Do we count cold items in calculations above?

Question 2) Cross-validation how? What is training data and testing data?