STCS 6701: Foundations of Graphical Models: Reading 5

Maxime TCHIBOZO (MT3390)

October 2020

1 Gopalan et al. (2015) – Scalable Recommendation with Hierarchical Poisson Factorization

In this paper, Gopalan, Hofman and Blei propose a recommendation system based on probabilistic modeling of user behavior: Hierarchical Poisson Factorization (HPF). HPF extends Matrix Factorization by representing user and item embeddings as latent variables with constraints generated by a specific Poisson-based Graphical Model. Each observed rating in the $\{user$ x $product\}$ matrix is assumed to be drawn from a Poisson distribution centered on the inner product of user and item latent vectors. In cases where the outcome in the $\{user$ x $product\}$ matrix is a binary variable, Poisson Matrix Factorization still yields satisfactory results.

User data in web experiments is usually sparse, and this sparsity is acounted for in HPF by placing Gamma priors on the latent variables, which differentiates HPF from probabilistic Matrix Factorization or Bayesian Matrix Factorization. The addition of the Gamma priors makes the model hierarchical and capable of capturing the long-tailed nature of user bahavior (most users interact with few products, but a few users interact with large quantites of items) – one of the known limitations of classical Matrix Factorization.

To approximate posterior latent distributions with the Hierarchical Poisson model, the authors of this paper use Mean Field Variational Inference, which is scalable and appropriate in a context of thousands of users interacting with thousands of products.

HPF was found to outperform competing methods (NMF, LDA, probabilistic MF with user and item biases, Collaborative filtering - CliMF) on scientific publication data, music data, news data and Netflix data.

One known limitation of some competing methods is their inability to convincingly account for missing values, i.e items for which we have not yet measured any outcome. HPF overcomes this, as well as the outcome imbalance problem, where negative outcomes appear less frequently, and should be weighted differently. Some other limitations are computational: Probabilistic MF uses MCMC for posterior inference, and thus does not scale to large datasets, Bayesian Personalized Ranking is too memory intensive, and some other methods simply yield unsatisfactory results.

Additionally, HPF yielded consistently satisfactory results over different datasets, whereas competing methods performed unevenly.

2 Boyd-Graber et al. (2017) – Applications of Topic Models (ch. 7-10)

- **Chapter 7:** Computational Social Science
- **Chapter 8:** Multilingual Data and Machine Translation
- **Chapter 9:** Building a Topic Model
- **Chapter 10:** Conclusion

This supplementary reading provided insights into Topic Models for my final project. Rather than a comprehensive summary of the book, this section contains a collection of findings which I found insightful.

For a topic model to be aware of bias/sentiment, it must contain a non-textual variable which represents bias/sentiment or is at least a proxy for bias/sentiment. The bias/sentiment and topic should be modelled jointly, with a specific data generating process which takes into account whether the bias/sentiment should come before (upstream) or after topics (downstream) in the generative story. Downstream models, in which bias/sentiment comes after the topics are "supervised" topic models, where topics are used to predict a document's bias/sentiment. The bias/sentiment could be modeled by a Gaussian distribution whose mean is the inner product of the normalized vector of topics used in the document, and a weight vector w describing the sentiment of each topic (see Blei and McAuliffe, 2007).

Beyond sentiment analysis, Boyd-Graber also highlights the use of topic models to infer the political stance of an author, or to infer the stance of a person on an issue (e.g. are they for or against a proposal?). Researchers who have published their findings on this subject include: Zhai et al. (2004), Lu and Zhai (2008), Paul and Girju (2010), Jo and Oh (2011), Nguyen et al. (2013), Nguyen et al.

(2015b). In these papers, the generative process assumes two sides can generate the observed data, and each side has both a distribution over words that it uses generally, and its own specific approach to generating topics. Words from each document are taken either from a side's general distribution, their version of the topic, or from the topic's neutral words. An important step in deriving the Data-Generating Process according to Boyd-Graber is the choice of an upstream or downstream bias/sentiment generation.

The nested Dirichlet process (Blei et al. 2010) offers a probabilistic framework to model such problems.

Another interesting application of Topic Models is in Multilingual Language Models: polylingual LDA (Minmo et al., 2009, pLDA) assumes that documents in different languages share the same topic distribution, but that each language has its own distribution over its word types. This approach is typically used for Exploratory Data Analysis. A specificity of Multilingual Language Models is the frequent misalignment between translation of documents into different languages. Typically, the length of a document varies greatly depending on the language (Wikipedia page translations are a good example), and the alignment of the two versions must be taken into account. This is an area of research in itself (Zhang et al., 2010; Boyd-Graber and Blei, 2009).

Multilingual Language Models are also useful for Statistical Machine Translation, which aims to convert sequences of words from one language to another. This typically works well for documents specific to a particular domain (e.g. sports).

Boyd-Graber also details how to design a topic model for a specific task, and highlights the need to find a balance between four key components: fidelity, performance, tractability and interpretability. Fidelity is the assessment of whether the statistical model (or data-generating process) matches the true process which produced the data. All models are approximations, and humans do not generate text according to Dirichlet processes. Fidelity often comes at the cost of performance, and in practice, downstream models seem to work better (Nguyen et al., 2013) even though they are less realistic. It is unfortunately impossible to know a priori whether a model will work well for a given task.

At times, a compromise on the type of variables (discrete, continuous) in the process must be made to allow for simplified implementations. Boyd-Graber views this as a tractability tradeoff. Complicated models are sometimes necessary, but they often do not yield better results than simple models. Boyd-Graber advises to start with the simplest possible model that could work, and to use this model as a baseline.

The final key component in model building is interpretability, the measure of how easily humans can understand the results of a model. Boyd-Graber highlights the concerning trend of researchers focusing on the quantified performance of their performance without inspecting the learned parameters of a model. Deep Learning has unfortunately increased this trend, but researchers developing probabilistic models should not underestimate the importance of model interpretability.

Describing a generative story (i.e. the sequence of probabilistic steps which could have generated the observed data) is usually sufficient to define a model. Once the model is established, we use probabilistic inference algorithms to retrieve the underlying parameters (latent variables) which most likely generated the observed data according to the model. From a practical point of view, framework such as Stan (works best with R), Infer.Net (Windows) and Automatic Differentiation can be used to define objective functions and perform inference using Monte Carlo methods, Variational Inference or Expectation Maximization. Here the key steps are: deriving the objective function, choosing a variational distribution, and checking the convergence of posterior predictive likelihood and latent variables.

Model checking (validation) can be performed by generating synthetic data by following the generative story and using the inferred parameters. One can also update the distributions used to model random variables both in the generative story and the variational inference mean field setup. Comparison between topic models is challenging, and one should first compare complicated models to implementations of simpler yet similar models in the same code. Posterior predictive checks (Minmo and Blei, 2011) allow researchers to evaluate whether LDA topic or word assignments successfully capture underlying patterns within the data which was not directly accounted for in the Data Generating model. Additionally, perplexity, which measures the likelihood on the validation or test set, is a useful metric for model checking, but may not be the most informative metric for a given task.

The final step for a zealous topic modeler is to share their findings with others. This implies providing sufficient evidence to convince sceptics that their model is in fact behaving as advertised. Topics should not be cherry-picked, but rather, should be randomly selected. Quantitative metrics (accuracy, likelihood, translation quality, precision at rank K) should be reported, and one should remember that latent variable models are learning distributions over a given dataset, and not point estimates. This implies a need to constructi confidence intervals, to run multiple inferences with different initializations – and more generally - to gather enough evidence to convincingly prove that the score of the model is not simply due to chance.

Fantastic read.