Junior Independent Work, Princeton Department of Mathematics
January 7, 2013
Advised by Professor David Blei

Zero-Inflated Poisson Factorization for Recommendation Systems

Max Simchowitz

1 Introduction

One of the most prevalent applications of contemporary machine learning, automated recommendation systems provide personalized item recommendations by extracting patterns from prior user behavior [2, 7]. The applications of recommendation systems are widespread, ranging from e-commerce, social networks, and academic paper recommendations matching [7, 6].

A recommendation system confronts two opposing problems. In order to be practical, recommendations systems need to be quick, efficient, and scale well in both computational complexity and memory cost [7, 23]. This is frequently achieved by reducing the dimensionality of the data [7]. For example, suppose we have N users, M items, and a review matrix $\mathbf{R} \in \mathbb{R}^{N \times M}$ consisting of positive integer ratings. Existing recommendation systems will try to find sparse or low rank approximations to $\mathbf{R}$, capturing an intuition that high dimensional rating observations are governed by relatively low dimensional preferences [23, 25, 16, 9].

On the other hand, modern review data sets include *sparse communities*: subsets $\mathcal{S}_U$ and $\mathcal{S}_I$ of the users and items, respectively, such that users $u \in \mathcal{S}_U$ have overwhelmingly reviewed items $i \in \mathcal{S}_I$, and vice versa. While sparse communities may indicate user preferences, frequently we may what to recommend an item in one one sparse community to a user belonging to another sparse community.

Our motivating example of sparse communities is restaurant review data. In online restaurant review services like Yelp, the majority of users rate restaurants in a concentrated geographic locations; generally, their place of residence. As a concrete example, consider a Yelp user Samantha who lives in Dallas. If Samantha wants to find a Dallas restaurant, then it is easy to compare her preferences to those of other locals, since they are likely to have visited many of the same establishments. However, if Samantha travels to Vancouver for the first time, then it is considerably harder to find other users who have frequented the same restaurants as Samantha, and who have also posted reviews about restaurants in Vancouver.

Difficulties associated with data sparsity are discussed in [7], which emphasizes the sheer paucity of user behavioral data in real datasets. The problem of missing data is treated from a theoretical perspective in [46], which gives provable bounds on “completing” low rank a matrix with unfortunately missing entries. However, unobserved reviews due to sparse community structure are not “missing” uniformly.

Indeed, sparse communities can introduce spurious structure into the data that may need to be un-learned. As we shall soon see, conventional recommendation systems may learn geo-spatial locations in place of what we might intuitively call “preferences”, such as tastes in cuisine, ambience, and price range. When it comes time to recommend a restaurant to a user when she travels, the latent feature space may be so saturated with geographic information that it cannot provide any universal insights into users tastes in food or atmosphere.

This paper will will introduce a Bayesian Zero-Inflated Poisson Factorization (ZIPF) model for recommendation systems that is more robust to sparse communities than existing methods. Like other Matrix Factorization methods, [18, 16, 20, 17, 14], ZIPF takes as an input nonegative integers which positively correlate with user preference, such as numerical ratings or click-through rates. We will refer to these data interchangably as reviews or ratings.

ZIPF has two components: a community membership model, which detects the sparse communities, and a rating model, which determines user preferences. Roughly speaking, ZIPF conditions the rating model on the community membership model, so that preferences are learned which are *independent* of the superficial sparse community structure.

We begin our exposition by reviewing the popular Nonnegative Matrix Factorization recommendation algorithm [18, 16, 20, 17, 14] in Section 2, mentioning its variants and extensions, and evaluating its efficacy in data sets with sparse communities. Section 3 discusses Bayesian Poisson Factorization (BPF) [2], outlining its advantages over conventional NMF while pointing out some assumptions about BPF that fail is data sets with sparse communities. Numerical results are provided which demonstrates the extent to which BPF learns sparse community structure.

Section 4 introduces the Zero-Inflated Poisson (ZIP) distribution [34], and outlines the (non-bayesian) EM algorithm for fitting ZIP models. Though it builds some helpful intuition for the following sections, it may be omitted on a first read. Section 5 places ZIP in Bayesian context, and describes a generic generative process for a class of models we will call Zero-Inflated Poisson Factorization (ZIPF). Here, we will present complete conditionals for latent variables representing user preferences and item features. This will inform a qualitative analysis of the benefits of ZIPF over BPF in sparse community data sets. Section 6 describes a variational inference algorithm for generic ZIPF models, and provides conditions under which this inference is tractable. Combining BPF with the AMP model described in [4], Section 7 provides a specific example of a ZIPF model and gives detailed variational inference algorithm. We implement this model on a small data set, and present preliminary numerical evidence for the strengths of ZIPF over BPF in a sparse community data set. Section 8 will adress issues of computational complexity, and present a stochastic algorithm in which each iteration takes sublinear time.

Section 9 describes the Reverse Zero Inflated Poisson algorithm, and compares it to the model described in Section 7. Both stochastic and non-stochastic inference are presented. Section 10 concludes with a discussion future extensions of and challenges for Bayesian Zero Inflated Poisson Models.

1.1 The Yelp Academic Dataset: An Example of Sparse Community Structure

The Yelp Academic Dataset exhibits archetypically sparse community structure [1]. The dataset consists of 330071 reviews of 13481 restaurants in the continental United States, provided by 130873 users. The data is certainly sparse: only .0019% of the ratings are provided, and each user reviews only 2.52 restaurants on average, though this number varies considerably: the variance in reviewed restaurants approximately 4.083, and very right skewed[1].

In addition to the scarcity of reviews, sparse communities arise as a result of geographic location. For each user u, let $\mu_u = (\mu_u^x, \mu_u^Y)$ denote the mean location, in longitude and lattidue, of all the restaurants which u has reviewed. The average standard deviation of each restaurant i reviewed by u from its mean lattidude and longitude are .69094 and 3.1637, repsectively. For reference, the lattitude and longitude coordinates for Los Angeles are $34.0500N, 118.2500W$, and $40.6700N, 73.9400W$ for New York. This lends strong evidence to the (quite unsuprising) hypothesis that users overwhelming review restaurants in one location.

1.2 Notation

In this paper, we will denote the number of users and number of items in our dataset by N and M, and the dimension of latent features K. Users will be indexed by u, items by i, and item features k. A review of item i by user U will be denoted by (u, i). The total number of reviews will be denoted R; in many real world datasets, $R << NM$ [7]. The following letters will denote sets: $\mathcal{N}$ the set of items, $\mathcal{M}$ the set of items, and $\mathcal{R}$ the set of observed review pairs (u, i). We will also let $\mathcal{R}_u$ denote the *set* of items reviewed by user u,

[1] We computed the skewness of the distribution of the number of restaurants visted by each user, $\gamma \triangleq \frac{1}{\sigma^3}\sum_i (X_i - \mu)^2$. Here σ is the standard deviation, and μ is the average number of restaurants reviewed. Our result was $\gamma \approx 11.600325017153084$

and $\mathcal{R}_i$ the set of users which have reveiwed item i. We will use the notation u/i to denote an index with is either a user u, or item i. The function $\|\cdot\|$ will denote the cardinality of a finite set.

Except in Section 9, we will adopt the convention that a non observed review r_{ui} is treated as a zero, which is standard practice in the recommendation system literature [7, 14, 6, 24]. Section 9 will formally differentiate between unobserved and zero reviews.

2 Non-Negative Matrix Factorization

In recent years, Nonnegative Matrix Factorization (NMF) has become one of the most popular frameworks for building recommender systems from high dimensional data [6, 24, 2, 7]. Since its introduction, NMF has also found applications text analysis, computer vision, pattern recomendation [13, 18, 20, 26, **?**].

NMF approximates a given N by M data matrix $\mathbf{R}$ by as a rank K factorization of *entrywise nonnegtive* matrices $\widehat{\mathbf{R}} \triangleq \mathbf{U}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{N\times K}$ and $\mathbf{V} \in \mathbb{R}^{M\times K}$ [18]. In the setting of recommender systems, $\mathbf{R} = (r_{ui})$ is the matrix of user-item ratings , where $\mathbf{R}_{ij}$ contains the u^{th} users rating of the i^{th} item. For user-item pairs ui for which ratings, the entry of $\mathbf{R}_{ui}$ is set to zero; that is, an unobserved review is regarded as a zero review [6].

The low rank factorization amounts to approximating observed ratings by the inner product of $K-$dimensional user and item features vectors in the rows of $\mathbf{U}$ and $\mathbf{V}$ [18, 14]. The strength of the approximation is measured by some cost function $\mathcal{C}(\mathbf{X}, \mathbf{U}, \mathbf{V})$.

Lee and Seung's landmark 1999 paper which introduced the NMF adopts the following cost functions:

$$\mathcal{C}_1(\mathbf{R}, \mathbf{U}, \mathbf{V}) = \|\mathbf{R} - \mathbf{U}\mathbf{V}^T\|_2 \quad \text{or} \quad \mathcal{C}_2(\mathbf{R}, \mathbf{U}, \mathbf{V}) = D(\mathbf{R}||\mathbf{U}\mathbf{V}^T) \tag{1}$$

where $\|\cdot\|_2$ denotes the entrywise l_2 norm, and $D(\mathbf{A}||\mathbf{B}) := \sum_{i,j} \mathbf{A}_{ij} \log \frac{\mathbf{A}_{ij}}{\mathbf{B}_{i,j}} - \mathbf{A}_{ij} + \mathbf{B}_{ij}$ denotes the entrywise Kullback-Leibler Divergence between two matrices $\mathbf{A}$ and $\mathbf{B}$ of the same dimension.

Many other cost functions exist in the literature: NMF has been modified to include l_1 and l_2 regularization to encourage sparsity, penalties based on the graph Laplacian, and extensions to semi-supervised learning problems [14, 25, 17, 13]. [24], [23], and [43] developed scalable algorithms for nonnegative matrix factorization, while [8] gives an in-depth of the complexitity of both exact and approximate NMF.

Experimental results have found NMF to work very well on review datasets for video streaming services such as Netflix, where users presuambly have equal access to all available content [24, 23, 6].

This is in part due to the nonnegativity that $\mathbf{U}$ and $\mathbf{V}$ leads to a "parts-based" representation of the data as opposed to PCA which learns eigenfeatures [20, 13]. For example, when NMF is used to study images of human faces, features in the $K-$dimensional subspace will correspond to "parts" of a face - mouths, noses, ears, whereas PCA will learn overall image shapes [20].

However, when sparse communities result from hard geographic bariers, NMF starts to falter. As mentioned above, unobserved reviewes are not differentiated from "0" reviewed. In the restaurant setting, the model does not differentiate between users who have not reviewed a restaurant because they live in a different city, or perhaps have just never heard of it, and users who have not reviewed a restaurant because they found it unexceptional. Indeed, if $r_{ui} = 0$, the loss functions $\mathcal{C}_1$ and $\mathcal{C}_2$ increase monotonically in the value of the predicted review $\widehat{\mathbf{R}}_{ui}$. Paradoxically, the NMF algorithm is more likely to recommend a restaurant that the user has given 1 or 2 stars, than a restaurant the user has never seen at all. In the next section, we shall provide graphical and numerical results with demonstrate the tendency of Bayesian Poisson Factorization to learn geographical features.

3 Bayesian Poisson Factorization

Bayesian Poisson matrix factorization (BPF) shows marked improvements over existing NMF techniques in both prediction quality and scalability [2]. Like NMF, BPF associates each user u and each item i with a K dimensional vector of weights $\theta_u = (\theta_{uk})$ and $\beta_i = (\beta_{ik})$ respectively. The weights are modeled by the generative process:

1. Draw $\theta_{uk} \sim \text{Gamma}(a, b)$
2. Draw $\beta_{ik} \sim \text{Gamma}(c, d)$
3. Draw feature contributions $x_{uik} \sim \text{Poisson}(\theta_{uk}\beta_{ik})$
4. Observe $r_{ui} = \sum_k x_{uik}$

From the additivity properties of the Poisson distribution, we see that $r_{ui} \sim \text{Poisson}(\theta_u^T \beta_i)$.

The BPF model tacitly assumes that the user has limited resources with which to consume [**?**], and so even if $\theta_u^T \beta_i$ is large, there is still nonzero that y_{ui} is zero. Indeed, traditional NMF assumes Gaussian loss [23], so that the probability of observing a zero review given a true review x decays as $\exp(-x^2/2)$. On the other hand, the probability of observing a zero given in the Poisson setting decays as $\exp(-x)$.

Consequently, BPF gives more weight to observed high ratings than to rating which are 0, and thus potentially unobserved [2]. While [37] modifies NMF to a similar end, the Poisson model is well motivated by the consumer behavior studies conducted in [38], and scales more efficiently than [37].

3.1 Numerical Tests of BPF

Though we have see that BPF penalizes a non-observed review less harshly than Gaussian loss NMF, numerical and graphical data demonstrates the BPF does in fact learn geographic features. We ran the BPF algorithm from [2] on the Yelp Academic set twice, with $K = 100$ features. On the first run, we with-held observed reviews uniformly by location. On the second run, we computed the mean longitudes and latitudes of the restaurants which each user had reviewed, and then disproportionately with-held reviews of restaurants which were further from this mean location (see Appendix D for details). Note that, in both trials, the same number of reviews were held out.

On the first run, the average l_1 and l_2 norms of the restaurant feature vectors β_i where $6.578 \times 10^{-4}$and 6.037×10^{-4}, respectively. On the second run, the average l_1 norm decreased slightly to 6.575×10^{-4}, while the average l_2 norm increased slightly to 6.064×10^{-4}. This preliminary test gave cursory evidence that increasing geographic sparsity, while fixing the total number of reviews, leads to sparser review features.

We then looked at restaurants in four geographic regions: Austin, Texas, Ann Arbor, Michigan, West Los Angeles, California, and East Los Angeles/Pasadena, California. Austin, Ann Arbor, and Los Angeles are all geographically distant, whereas West and East Los Angeles neighbor one another. The data set contained 500 restaurants from Austin, Ann Arbor and East LA/Pasadena, and 999 from West Los Angeles.

For each trial and each location, we plotted the mean strength of the features β_{ik}, as well as the fraction of restaurants for which β_{ik} was the largest feature (that is, $\beta_{ik} > \beta_{ik'} \forall k' \neq k$). Plots are displayed in Appendix E.

We see at once that the features for Texas, West Los Angeles, and especially Michigan are particular sparse. Indeed, over 50% of feature strength for Michigan restaurants is attributed to a single feature. Feature seems strength seems to be more spread out among East Los Angeles/Pasadena restaurants. Nevertheless, feature sparsity is conspicuously greater when reviews are heldout by location in trial 2 than with uniform holdouts in trial 1. This further supports the hypothesis that BPF learns location features very strongly.

3.2 Theoretical Drawbacks of BPF for Sparse Community Data Sets

In the Bayesian Poisson Factorization model, the quantity $\theta_u^T \beta_i$ completely specifies the distribution of r_{ui}. Intuitively, this quantity corresponds to the "compatibility" between the users preferences, encoded by θ_u, and the items attributes β_i. Thus, the larger the quantity $\theta_u^T \beta_i$, the more likely that user u will have a positive experience with item i, and provide a good review. In particular, substituting $\theta_u^T \beta_i$ into the Poisson density gives

$$Pr(r_{u,i} = 0) = \exp(-\theta_u^T \beta_i) \tag{2}$$

While BPF penalizes zero reviews less harshly than NMF [2], it nevertheless follows the recommendation system convention that unobserved reviews r_{ui} are treated as zero [6, 24, 23, 7]. Thus, the above equation stipulates that the probability of not observing is solely a function of the compatability of user preferences and item features. This property fails very quickly in many real-world examples where sparse communities arise:

Scenario 1 Consider a user in Tennessee whose prefers Turkish cuisine over conventional Southern fare. Contrary to the Poisson model, she has probably provided far more strictly positive reviews for smokehouses and burger joints in Nashville than for any eateries in Istanbul. It would be an embarrassment if, upon traveling to Turkey, our recommendation system suggest she visit an American chain, and not highlight the traditional restaurants which she would find more appealing.

We now ask, how could we possibly know that this user likes Turkish food? Since Turkish food is perhaps less prevalent than Barbecue in the users neighborhood, we get more information about the users preferences by observing a review of a Turkish restaurant than of a barbecue establishment. In a rough and intuitive way, we should try to "weight" her reviews from the restaurants that she is less likely to visit more than her evaluation of a place that she walks by every day.

It is even possible that this user has never visited a Turkish restaurant, but has shown preferences for Israeli and Greek cuisine, or perhaps has exceptionally similar tastes to friends who have visited Turkish restaurants. Again, we should not penalize the users potential preferences for Turkish food too much. Ideally, we should try to learn underlying preferences that inform the users predilection for, say, Eastern-Mediterranean cuisine, even though the user has had only little exposure to the type of cooking. Indeed, there are "heat diffusion" recommendations which tend to give diverse, and novel product recommendations, but this diversity is often at the expense of adhering to what the users most like [7]. Rather, an effective recommendation should introduce the user to items she would not have otherwise discovered on her own because of some geographic or informational barrier, but which the user would have otherwise greatly enjoyed. In the next section, we will develop a model that captures precisely this intuition. But to crystallize our thoughts, we should try to imagine a couple more examples.

Scenario 2 The barriers which prevent a review from being provided need not be geographic. When Academic Journals like ArXiv try to recommend interesting material to their contributors, click data or article reviews can be deciding. Indeed, there are many research groups in Computer Science, Operations Research, and Physics departments which study problems in optimization and statistical inference. Frequently, the structure of academic departments may impede dialog across these nominally different disciplines, even though the aim, if not the content, of different groups' work is quite similar. A recommendation approach which fails to recognize these separate communities will fail to expose the Computer Science Post-Doc working with graphical models to a physics paper which introduces highly relevant techniques, but applies them to analyze data from the Large Hadron Collider. Somehow, we need to un-bias our recommendation from over-fitting to reviews in disciplines which the user has already seen (though not too much, since discipline is certainly a good heuristic).

Scenario 3 The issue in the previous scneario becomes even more pronounced for social recommendation sites. Take, for example, GoodReads, where users can discuss and rate books amongst friends. While it is reasonable to assume that book preferences are somewhat uniform across social circles, a mature Recommendation System should be able to distinguish the nuances in preferences between users tastes, and even expose users to potential friends users whose preferences may be more similar than their current connections.

For example, suppose Johnathan and all his friends have posted favorable reviews for the latest young-adult vampires series. We learn far more about Johnathan's preferences when we see his glowing review of Jane Eyre, or his lackluster takeaway from Freakonomics, than when he gives the fifth book in the aforementioned series the same rating as his friends have.

In the next section, we introduce Zero Inflated Poisson Factorization (ZIPF), discuss the algorithm, and conjecture about how this technique yields improved reviews in the three scenarios above.

4 Frequentist Zero-Inflated Poisson

4.1 Introduction to the Zero Inflated Poisson Distribution

Following [34, 31, 33], we say that a random variable X has the Zero Inflated Poisson (ZIP) Distribution when

$$Pr(X = k) = p\mathbb{1}_{k=0} + (1-p)\frac{e^{-\lambda}\lambda^k}{k!} \tag{3}$$

As shorthand, we write $X \sim \text{ZIP}(p, \lambda)$. We can think of the ZIP as the product of a Poisson Random Variable $Y \sim \text{Poisson}(\lambda)$ with a Bernoulli random variable $B \sim \text{Ber}(1-p)$ which is independent of Y. For example, a ZIP random variable is the outcome of flipping a weighted coin with probability of observing a heads as $1-p$: if heads , draw from a Poisson distribution with parameter λ; if tails, we observe a zero. That is, the weighted coin "inflates" the probability that we observe a zero.

Thus, if $r_{u,i} \sim \text{ZIP}(\theta_u^T\beta_i, p_{ui})$, we see that p_{ui} bears most of the responsibility for whether or not we observe $r_{u,i}$, while that $\theta_u^T\beta_i$ still controls the value of r_{ui} in the event that it is observed. Nevertheless, there is a nonzer zero is drawn from the Poisson distribution, which decays exponentially in $\theta_u^T\beta$. Thus, ZIP retains the intuition that, the more the user is likely to enjoy an item, the more he or she is likely to purchase and review it.

For example, the user's preference for Turkish food in **Scenario 1** may be encoded into a high value of $\theta_u^T\beta_i$ for restaurant i in Istanbul, while her unlikeliness to have visited restaurants like i will be encoded in a high value of p_{ui}. Once we have derived complete conditions for zero inflated poisson, the theoretical advantages of ZIP models over BPF can be made more precise.

In the following sections, we will describe the Expectation Maximization Algorithm (EM) and how it fits Zero-Inflated Poisson in a non-Bayesian setting. While the subsequent sections of this paper will be Bayesian, the strategy of using latent variables for represent the "zero-inflation" in EM is the cornerstone of Bayesian ZIP inference. Moreover, the Variational Inference introduced in Section 6 is essentially a Bayesian adaption of EM [21].

4.2 Maximum Likelihood Estimation and the EM Algorithm

In frequentist parametric statistics, one assumes that the data set $\mathbf{X} = X_1, \ldots, X_n$ are observations from a distribution P_θ in a family of distributions indexed by parameters $\theta \in \Theta$, where Θ is generally taken to be a subset of $\mathbb{R}^d$ [41].

Let f_θ be the density corresponding to the probability P_θ with respect to some base measure μ, otherwise known as the Radon-Nikodyn derivative $dP_\theta/d\mu$ [39][2]. Given observations of random variables $X_1, X_2, \ldots, X_n$, the goal is to compute the parameter

$$\theta^* := \arg\max_{\theta\in\Theta} f_\theta(X_1, \ldots, X_n) = \arg\max_{\theta\in\Theta} \mathcal{L}(\theta; X_1, \ldots, X_n) \tag{4}$$

where the likelihood function $\mathcal{L}(X_1, \ldots, X_n)(\cdot)$ is the mapping $\theta \mapsto f_\theta(X_1, \ldots, X_n)$. The parameter θ^* is known as the maximum likelihood estimator, or MLE.

The MLE satisfies many convenient properties; under relatively mild conditions, it is an asymptotically optimal estimator of the true parameter θ_0 [41]. Fixing the observations $X_1, \ldots, X_n$, we define the likelihood function $\mathcal{L}_{\theta;X_1,\ldots,X_n} := f_\theta(X_1, \ldots, X_n)$. When $X_1, \ldots, X_n$ are assumed to be independent and identically distributed, the likelihood factors as the product of likelihoods

$$\mathcal{L}_{\theta;X_1,\ldots,X_n} = \prod_{i=1}^n \mathcal{L}(\theta; X_i) \tag{5}$$

[2] For example, we can parameterize 1-dimension Gaussian distributions by $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}^+$, where the densities are $f_{(\mu,\sigma)} = \frac{\exp(-(x-\mu)^2/2\sigma^2)}{\sqrt{2\pi}\sigma}$

This form motivates considering the log of the likelihood function, since sum are generally easier to optimize than products.

The MLE is often difficult to compute directly. One solution is to introduce latent variables $\mathbf{Z} = Z_1, \ldots, Z_k$, and observe that

$$\log \mathcal{L}(\theta; \mathbf{X}) = \sum_i \log f_\theta(X_i) = \sum_i \int_{\mathbf{Z}} \log f_\theta(X_i, \mathbf{Z}) \tag{6}$$

Expectation Maximization them optimizes $\log \mathcal{L}(\theta; \mathbf{X})$ over θ by iterating over two steps until convergence is achieved [40]:

1. The Expectation Step, or "E step", computes the expectation of the log-likelihood of θ, given $\mathbf{X}$ and $\mathbf{Z}$. That is, it computes the t^{th} estimate of θ
$$\mathcal{Q}(\theta, \theta^{(t)}) = \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^t} \log \mathcal{L}(\theta; \mathbf{X}, \mathbf{Z})$$
2. The Maximization Step, or "M step", which sets θ^{t+1} to maximize the expected log likelihood:
$$\theta^{(t+1)} \leftarrow \arg\max_\theta \mathcal{Q}(\theta, \theta^{(t)})$$

4.3 An EM Algorithm from i.i.d. observations from a ZIP

Following [31] and [33], we demonstrate how an EM algorithm can learn a ZIP model.

$$\log \mathcal{L}(\lambda, p; X_1, \ldots, X_n) = \sum_{i=1}^n \log \left(p\mathbb{1}_{X_i=0} + (1-p)\frac{e^{-\lambda}\lambda^k}{k!} \right) \tag{7}$$

This form is difficult to work with, since there is a sum inside the log. However, suppose know, for each zero X_i, whether or not the zero came from the Poisson part or from the Bernoulli part. We could then introduce the auxillary indicator variable ι_i which akes the value 1 if X_i is zero because of the Bernoulli part, and 0 if we observed X_i from the Poisson part [34, 31] . Thus, our log likelihood would take the form.

$$\begin{aligned} \log \mathcal{L}(\lambda, p; X_1, \ldots, X_n, \iota) &= \sum_{i=1}^n \iota_i \log p + (1-\iota_i)(1-p_i)\frac{e^{-\lambda}\lambda^k}{k!}) \\ &= \sum_{i=1}^n \iota_i \log\left(\frac{p}{1-p}\right) + \log(1-p) + (1-\iota_i)\log\left(\frac{e^{-\lambda}\lambda^k}{k!}\right) \end{aligned} \tag{8}$$

Here $(\lambda, p) \in \mathbb{R}_+ \times [0, 1]$ is the underlying parameter, and ι_i are the latent variables. Wecan apply the EM algorithm [34, 31]:

1. First, the "E" step computes the expected values of ι_i which maximize the expected log likelihlood. This is given by substituing

$$\iota_i^{(t)} \quad \leftarrow \quad \frac{p\mathbb{1}(X_i = 0)}{(1-p^{(t)})e^{-\lambda^{(t)}} + p^{(t)}\mathbb{1}(X_i = 0)}$$

into the log likelihood expression $\log \mathcal{L}(\lambda, p; X_1, \ldots, X_n, \iota)$ to obtain

$$\mathcal{Q}(\lambda, p; \lambda^{(t)}) \quad = \quad \sum_{i=1}^n \{\iota_i^{(t)} \log\left(\frac{p}{1-p}\right) + \log(1-p) + (1-\iota_i^{(t)} \log(\frac{e^{-\lambda}\lambda^k}{k!})\}$$

where we recall that $\iota^{(t)}$ is a function of $\lambda^{(t)}$.

2. In the "M" step, we see that $\mathcal{Q}(\lambda, p; \lambda^{(t)}, p^{(t)})$ splits into a sum which depends only on p and another sum which depends only on λ. Thus, we can set $p^{(t+1)}$ and $\lambda^{(t+1)}$ to the values of p and λ which maximize each of the

$$\lambda^{(t+1)} \leftarrow \arg\max_\lambda \sum_{i=1}^n (1-\iota_i^*)\log\left(\frac{e^{-\lambda}\lambda^k}{k!}\right) \qquad p^{(t+1)} \leftarrow \arg\max_p \sum_{i=1}^n \{\iota_i^* \log\left(\frac{p}{1-p}\right) + \log(1-p)\}$$

Both maximizations can be easily computed by taking derivatives.

5 Bayesian ZIPF for Recommendation Systems

5.1 Specification of Model

Here we introduce a general framework for Zero-Inflated Poisson Factorization, or ZIPF. As mentioned above, we will model the review of item i by user u as coming from a Zero-Inflated Poisson distribution, ZIP$(\theta_u^T \beta_i, p_{ui})$, where p_{ui} depends on both the user and item.

With the discussion of frequentist ZIP inferrence in mind, introduce the hidden variables $\iota_{u,i}$, such that $\iota_{u,i} = 1$ if $r_{u,i} = 0$ due to the Bernoulli part, and 0 otherwise. We call ι_{ui} a community membership indicator, informed by the restaurant review setting in which a user is more likely to review a restaurant if the two belong to the same community. We choose the term “community” over location to denote that communities can be abstract, and need not reflect geography. Indeed, communities can arise due to common interests, political ideology, or social network affiliations.

The indicators ι_{ui} form a bipartite user-item graph, which we can model with existing network models in the literature [28, 5, 4]. At this time, we shall assume that ι relies on latent community membership variables ζ_j, but will not specify a distribution over these latent variables. After introducing variational inference, we will present conditions on the community membership model which permit tractable variational inference. By an abuse of language, we will refer to the ι_{ui} as “links” on our graph, in the sense that the event $\iota_{ui} = 1$ corresponds to a graph link between users u and item i, and the event $\iota_{ui} = 0$ refers to a nonlinks [5]. Bear in mind that in ZIPF, and unlike other network models, the links are not necessarily known. While a positive rating gaurantees the presence of a link, a zero rating does not guarantee its absence. Thus, conditioned on our observations, the indicators ι_{ui} form a random network, whereas, in [28, 5, 4], the observations make the network links deterministic.

Following [2], we assume that user preferences θ_u and item features β_i lie in the nonnegative orthant $\mathbb{R}_+^K$, and place the prior $\theta_u \sim \text{Gamma}(a, b)$ and $\beta_i \sim \text{Gamma}(c, d)$. We take then take the observed review to be $r_{u,i} \sim \mathbb{1}_{z_{u,i}>0}\text{Poisson}(\theta_u^T \beta_i)$. We also introduce the latent variables $x_{i,u,k} = \mathbb{1}_{z_{ui}>0}\text{Poisson}(\theta_{uk}\beta_{uk})$. It follows from the well known additivity properties of the Poisson distribution that $\sum_k x_{uik} = r_{ui}$. The generative process is summarized below:

1. Community Membership Model
   (a) Draw underlying community membership paramters ζ_k from some generic distribution, to be specified later.
   (b) Draw community membership indicator $\iota_{ui} = I[\sum_{k'} z_{uik'} > 0]$.

2. Rating Observation Model
   (a) Draw latent user preferences $\theta_{ui} \sim \text{Gamma}(a, b)$.
   (b) Draw latent item features $\beta_{ik} \sim \text{Gamma}(c, d)$.
   (c) Draw $x_{i,u,k} = \iota_{ui}\text{Poisson}(\theta_{uk}\beta_{uk})$.
   (d) Observe $r_{u,i} = \sum_k x_{i,u,k}$

We therefore see that $r_{ui} \sim \text{ZIP}(1 - \iota_{ui}, \theta_u^T \beta_i)$. The generative process is specified in the Graphical Model shown in Figure 1.

5.2 Complete Conditionals for x_{uik}, β_i, and θ_u

From Appendix A, the complete conditions for global variables $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ are

$$\theta_{uk}|\boldsymbol{\beta}, \boldsymbol{\iota}, \mathbf{x} \sim \text{Gamma}(a + \sum_i \iota_{ui} x_{iuk}, b + \sum_i \iota_{ui} \beta_{ik}) \quad \text{and} \quad \beta_{ik}|\boldsymbol{\theta}, \boldsymbol{\iota}, \mathbf{x} \sim \text{Gamma}(a + \sum_u z_{iu} x_{iuk}, b + \sum_u \iota_{ui} \theta_{uk}) \tag{9}$$

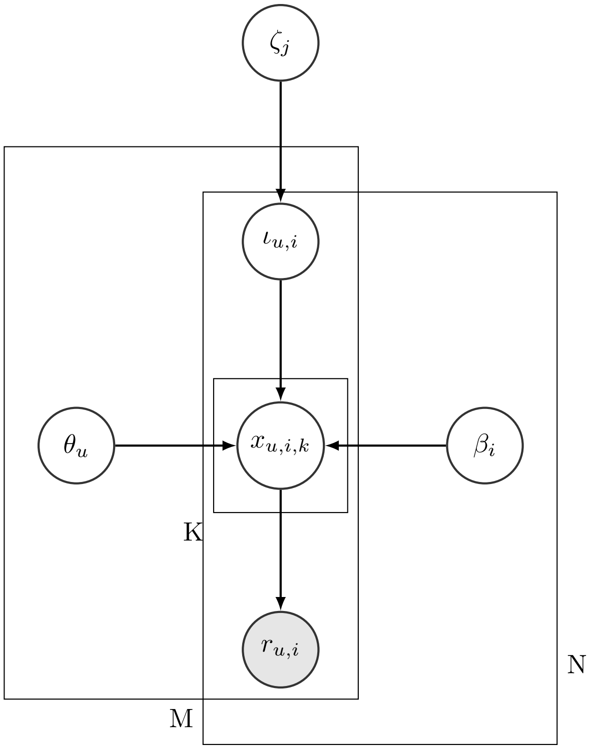

Figure 1: Graphical Representation of a Generic Bayesian Zero-Inflated Poisson Factorization Model

Note then that if $\iota_{ui} = 0$, then $x_{uik} = 0$ deterministically, so that we can simplify

$$\begin{aligned} \theta_{uk}|\boldsymbol{\beta}, \boldsymbol{\iota}, \mathbf{x} &\sim \text{Gamma}(a + \sum_i x_{iuk}, b + \sum_i \iota_{ui}\beta_{ik}) \\ \beta_{ik}|\boldsymbol{\theta}, \boldsymbol{\iota}, \mathbf{x} &\sim \text{Gamma}(a + \sum_u x_{iuk}, b + \sum_u \iota_{ui}\theta_{uk}) \end{aligned} \tag{10}$$

The complete conditions for the local variables are the same as in the non zero inflated setting [2]:

$$x_{i,u,k}|\mathbf{r}, \boldsymbol{\iota}, \boldsymbol{\theta}, \boldsymbol{\beta} \sim \text{Mult}(r_{ui}, \frac{\theta_u * \beta_i}{\theta_u \beta_i^T}) \tag{11}$$

since if $\iota = 0$, $r_{ui} = 0$, while if $\iota = 1$, then $x_{uik} \sim \text{Poisson}(\theta_{uk}\beta_{ik})$. Here the $*$ operator denotes the Hadamard (pointwise) product.

5.3 Qualitative Comparison of ZIPF with BPF

Now, let's compare these conditional distributions to the ones given by [2]:

$$\begin{aligned} \theta_{uk}|\boldsymbol{\beta}, \mathbf{x} &\sim \text{Gamma}(a + \sum_i x_{iuk}, b + \sum_i \beta_{ik}) \\ \beta_{ik}|\boldsymbol{\theta}, \mathbf{x} &\sim \text{Gamma}(a + \sum_u x_{iuk}, b + \sum_u \theta_{uk}) \end{aligned} \tag{12}$$

We see that the only difference is that, for the ZIP conditionals, the second parameter Gamma Distribution parameter only contains the sum over β_{ik} or θ_{uk} for which ι_{ui} is not zero. This makes sense; it is simply the conditional one would get if we only used the reviews r_{ui} for which $\iota_{ui} = 1$, or more intuitively, given only the reviews which were not hidden by some geographic or informational barrier. We can say even more.

Recall that that if $X \sim \Gamma(a, b)$, then its density function is $\frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx} dx$ [41]. We see then that as the value of b increases, more weight is placed is placed on lower values of X, and conversely, as b decreases, X takes on larger values with higher probability. On the other hand, large values of a give more weight to larger values of X, and vice versa for small a.

Returning to the conditionals presented in [2], we see that a users preference for feature k, θ_{uk}, is more likely to be large when the sum of the contribution of feature k to all the reviews, $\sum_i x_{iuk}$, is large, and when all the items being reviewed could have possibly seen contain little of feature k; i.e, $\sum_i \beta_{ik}$ is small. We can illustrate this intuition by the following scenario:

Scenrio 4: Suppose that Movie-o is a fictional video streaming website. If Bob is a Movie-o user who has given very positive reviews to many Foreign-Language films, then obviously we expect that Bob enjoys foreign movies. Suppose k is a feature corresponding to foreign film, let u be Bob's user index, and let i range over the indices for all Movie-o films. If Movie-o only only offers very few foreign language movies, then $\sum_i x_{uik}$, that is, the total of Bobs reviews which are informed by his preference for foreign film, will be rather small. Nevertheless, $\sum_i \beta_{ik}$ will also be small, so that $\theta_{uk}|\beta, \mathbf{x}$ will still tend to be large. Thus, even when a scarcity of foreign film gives us sparse information about Bobs movie preferences, Bayesian Poisson Factorization still performs quite well.

Now, lets assume that Movie-o actually does offer lot of foreign film, but for some reason, Bob has no idea. Perhaps Movie-o has a clunky interface which prevents exploration, or Bob knows too little about the genre to search for foreign movies, or that, unbeknownst to Bob, Movie-o drastically expanded their collection in the last two months. Bayesian Poisson Factorization starts to falter, because $\sum_i \beta_{ik}$ is now very large, but $\sum_i x_{iuk}$ stays fixed.Thus, the BPF recommender system will mistakenly conclude that Bob has lost his taste for foreign film.

Here, ZIP shows substantial improvement. If we are given the latent variables ι_{ui} that tell us which movies Bob might have come across, the vast collection of foreign about which Bob has no idea knows not factor into the ZIP complete conditional of θ_{uk}. In effect, ZIP Bayesian Factorization gives more weight to Foreign film because Bob has given positive reviews, despite his limited exposure. We can therefore see that the this model captures the intuitions which informed **Scenario 1-3**.

6 Inference for Bayesian ZIPF

6.1 Variational Inference

In this section, we describe mean field variationa inference, the Bayesian analogue of the EM algorithm in Section 4 [21]. Mean field variational inference approximates the true posterior distribution with a family of probability distributions q over hidden variables which is indexed by free parameters, such that each of the latent variables are independently distributed given these parameters [40, 29, 3].

For ZIPF, the variational distribution factors as

$$q(\beta, \theta, x, \iota, \xi) = \prod_{ik} q(\beta_{ik}|\lambda_{ik}) \prod_{u,k} q(\theta_{uk}|\gamma_{uk}) \prod_{u,i} q(x_{ui|\theta_{ui}}) q(\iota_{ui}|s_{ui}) \prod_{j} q(\xi_j|\zeta_j) \tag{13}$$

where the distribution of the random variables β, θ, x, ι and ξ given their free parameters are in the same families as for the complete conditionals. Here, we have that $\lambda_{kk} = \langle \lambda_{ik}^{\text{shp}}, \lambda_{ik}^{\text{rte}} \rangle$ and $\gamma_{uk} = \langle \gamma_{ik}^{\text{shp}}, \gamma_{ik}^{\text{rte}} \rangle$, corresponding the shape and rate parameters of the Gamma distribution, respectively. Written explicitly,

$$\begin{aligned} \beta_{ik} \sim \text{Gamma}(\lambda_{ik}^{\text{shp}}, \lambda_{ik}^{\text{rte}}) \quad &\text{and} \quad \theta_{uk} \sim \text{Gamma}(\gamma_{uk}^{\text{shp}}, \gamma_{uk}^{\text{rte}}) \\ x_{ui} \sim \text{Multinomial}(r_{ui}, \phi_{ui}) \text{ for } r_{ui} > 0 \quad &\text{and} \quad \iota_{ui} \sim \text{Bernoulli}(s_{ui}) \text{ for } r_{ui} = 0 \end{aligned} \tag{14}$$

where x_{ui} is the zero vector when $r_{ui} = 0$, and $s_{ui} = 1$ when $r_{ui} > 0$.

By minimizing the KL divergence between the variational and posterior distributions, we obtain the by maximizing the evidence lower bound, or ELBO on $\log p(\mathbf{r})$, the logarithm of the marginal probability of the observed reviews [29, 40, 22, 3]. Using Jensen's inequality and the concavity of log, we compute the ELBO for the ZIPF Model

$$
\begin{aligned}
\log p(\mathbf{r}) &= \log \int p(\mathbf{r}, \mathbf{x}, \beta, \theta, \iota, \xi) d\mathbf{x}, d\beta, d\theta, d\iota, d\xi \\
&= \log \int p(\mathbf{r}, \mathbf{x}, \beta, \theta, \iota, \xi) \frac{q(\mathbf{x}, \beta, \theta, \iota, \xi)}{q(\mathbf{x}, \beta, \theta, \iota, \xi)} d\mathbf{x}, d\beta, d\theta, d\iota, d\xi \\
&= \log \mathbb{E}_q \left[\frac{p(\mathbf{r}, \mathbf{x}, \beta, \theta, \iota, \xi)}{q(\mathbf{x}, \beta, \theta, \iota, \xi)} \right] \\
&\geq \mathbb{E}_q \left[\log p(\mathbf{r}, \mathbf{x}, \beta, \theta, \iota, \xi) \right] - \mathbb{E} \left[\log q(\mathbf{x}, \beta, \theta, \iota, \xi) \right] \\
&\triangleq \mathcal{L}(q)
\end{aligned}
\tag{15}
$$

We can maximize the ELBO iteratively by optimizing one variational parameter whilst holding the others constant. This amounts to taking the gradient of the ELBO with respect to one variational parameter, and choosing that parameter so that the gradient is zero. For larger datasets in which iterative optimization may become intractable, we often a stochastic gradient ascent algorithm based on sub-sampling mini-batches of users and items [5, 3].

From the graphical model in Figure 1, we see that $p(\mathbf{r}, \mathbf{x}, \beta, \theta | \iota, \xi)$ is just $p(\mathbf{r}, \mathbf{x}, \beta, \theta | \iota)$. Thus, we have

$$
\begin{aligned}
\mathcal{L}(q) &= \mathbb{E}_q \left[\log p(\mathbf{r}, \mathbf{x}, \beta, \theta, \iota, \xi) \right] - \mathbb{E}_q \left[\log q(\mathbf{x}, \beta, \theta, \iota, \xi) \right] \\
&= \mathbb{E}_q \left[\log p(\mathbf{r}, \mathbf{x}, \beta, \theta | \iota, \xi) \right] - \mathbb{E}_q \left[\log q(\mathbf{x}, \beta, \theta) \right] + \mathbb{E}_q \left[\log p(\iota, \xi) \right] - \mathbb{E}_q \left[\log q(\iota, \xi) \right] \\
&= \mathbb{E}_q \left[\log p(\mathbf{r}, \mathbf{x}, \beta, \theta | \iota) \right] - \mathbb{E}_q \left[\log q(\mathbf{x}, \beta, \theta) \right] + \mathbb{E}_q \left[\log p(\iota, \xi) \right] - \mathbb{E}_q \left[\log q(\iota, \xi) \right] \\
&\triangleq \mathcal{L}_1(q) + \mathcal{L}_2(q)
\end{aligned}
\tag{16}
$$

where $\mathcal{L}_1(q) \triangleq \mathbb{E}_q \left[\log p(\mathbf{r}, \mathbf{x}, \beta, \theta | \iota) \right] - \mathbb{E}_q \left[\log q(\mathbf{x}, \beta, \theta) \right]$ and $\mathcal{L}_2(q) \triangleq \mathbb{E}_q \left[\log p(\iota, \xi) \right] - \mathbb{E}_q \left[\log q(\iota, \xi) \right]$.

We will occasionally refer to $\mathcal{L}_1$ as the *rating model objective* and $\mathcal{L}_2$ as the *community membership model objective*. Note then that the only free parameters on which $\mathcal{L}_1(q)$ and $\mathcal{L}_2(q)$ simultaneously depend are s_{ui}, the free parameters for ι. Thus, each step of our coordinate ascent algorithm amounts to first optimizing $\mathcal{L}_1(q)$ over the variational parameters for the rating parameters x, β, θ, then optimizing $\mathcal{L}_2(q)$ for the community membership parameters ξ, and finally optimizing the parameters governing ι_{ui}.

6.2 The Exponential Family, Conditional Conjugacy, and the Natural Gradient

Before introducing the coordinate ascent algorithm, we need a couple preliminaries. A family of probability distributions $\{P_\theta, \theta \in \Theta\}$ indexed by parameters θ are said to be in the *exponential family* if each has a density which can be expressible in *canonical exponential form* [3, 29, 41]:

$$
p(X|\theta) = h(x) \exp(\eta(\theta)^T t(x) - \alpha(\eta(\theta))) \tag{17}
$$

The functions $h(\cdot)$ and $a(\cdot)$ are respectively denoted the *base measure* and *log normalizer*, and the vector-valued functions $\eta(\cdot)$ and $t(\cdot)$ are the *natural parameter* and the *sufficient statistic* [3]. Here $h(\cdot)$ can either be a density, in the case of a continuous random variable, or a function in the case of discrete discrete random variables. It can be shown that α is twice differentiable as a function of η [41, 29], and we shall assue that η itself is a twice differentiable function. The exponential family covers a wide range of distributions, including the Normal, Poisson, Gamma, and Multinomial distributions.

In this paper, we shall make use of the fact that the natural parameter of a gamma distribution with $X \sim \Gamma(a, b)$ is just the 2-dimensional vector (a, b). For a multinomial distribution $(X_1, \ldots, X_k) \sim \text{Multi}(v, m)$, where $v = (v_1, \ldots, v_k)$ is entry-wise nonnegative and whose entries sum to one, the natural parameter is given by $(\log v_1, \ldots, \log v_k)$.

Let $\theta_1, \ldots, \theta_k$ be a parameters drawn from prior distributions $P_{\theta_1} \ldots P_{\theta_n}$, and X a random observation from a distribution depending on $\theta_1, \ldots, \theta_k$. We say that a model is *conditionally conjugate* if the distribution of the *complete conditional* $\theta_j | X, \theta_{-j}$ is in the same family as P_{θ_j} with natural parameters $\eta(X, \theta_{-j})$[3] [3, 29].

[3] By θ_{-j}, we mean $\{\theta_i, i \neq j\}$

Let's consider the ELBO for a conditionally conjugate exponential family model, where each θ_j is governed by the variational parameter λ_j, such that λ_j is the natural exponential family parameter. If the priors on each θ_j all lie in the exponential family, then the gradient of the ELBO with respect to a parameter λ_j is given by [3]

$$\begin{aligned}\nabla_{\lambda_j}\mathcal{L}(q) &= \nabla_{\lambda_j}\mathbb{E}_q\left[\log p(X,\theta) - \log q(X,\theta)\right] \\ &= \nabla_{\lambda_j}\mathbb{E}_q\left[\log p(\theta_j|X,\theta_{-j}) - \log q(\theta_j)\right] \\ &= \nabla_{\lambda_j}\left(\mathbb{E}_q\left[\eta(X,\theta_{-j})^T t(\theta_j)\right] + \alpha(\eta(X,\theta_{-j})) - \lambda_j^T t(\theta_j) + \alpha(\lambda_j)\right] \\ &= \nabla_{\lambda_j}\left(\mathbb{E}_q\left[\eta(X,\theta_{-j})^T t(\theta_j)\right] - \lambda_j^T t(\theta_j) + \alpha(\lambda_j)\right]\end{aligned} \tag{18}$$

where we drop constants terms in $\mathcal{L}_q$ which have no λ_j dependence in the second line, and the term $\alpha(\eta(X,\theta_{-j}))$ in the third line. We can then apply the exponential family identity that the expectation of the sufficient statistics is the gradient of the log normalizer to further simplify

$$\begin{aligned}\nabla_{\lambda_j}\mathcal{L}(q) &= \nabla_{\lambda_j}\Big(\mathbb{E}_q\left[\eta(X,\theta_{-j})\right]^T \nabla_{\lambda_j}\alpha(\lambda_j) - \lambda_j^T\nabla_{\lambda_j}\alpha(\lambda_j) + \alpha(\lambda_j)\Big) \\ &= \nabla^2_{\lambda_j}\alpha(\lambda_j)\left(\mathbb{E}_q\left[\eta(X,\theta_{-j})\right] - \lambda_j\right)\end{aligned} \tag{19}$$

This gradient is zero when we set $\lambda_j = \mathbb{E}_q\left[\eta(X,\theta_{-j})\right]$, that is, by setting the variational parameter equal to the expected value of the natural parameter of its complete conditional, under the variational distribution q [3, 21, 29, 40].

The gradient of the ELBO gives the direction of steepest ascent, in Euclidean distance [3]. If instead we define distance between two variational distributions q and q' is defined by the symmetrized KL divergence

$$D^{\text{sym}}_{KL}(q(\theta), q'(\theta)) = \mathbb{E}_{q(\theta)}\left[\log\frac{q(\theta)}{q'(\theta)}\right] + \mathbb{E}_{q'(\theta)}\left[\log\frac{q'(\theta)}{q(\theta)}\right] \tag{20}$$

then the direction of steepest ascent is given by the *natural gradient*:

$$\widehat{\nabla}_{\lambda_j}\mathcal{L}(q) = G(\lambda_j)^{-1}\nabla_{\lambda_j}\mathcal{L}(q) \tag{21}$$

where $G(\lambda_j)$ is the *Fisher Metric* [35, 3]. For exponential family distributions with natural parameter λ, we have $G(\lambda) = \nabla^2\alpha(\lambda)$. Thus, for our conditionally conjugate exponential family model,

$$\widehat{\nabla}_{\lambda_j}\mathcal{L}(q) = \mathbb{E}_q\left[\eta(X,\theta_{-j})\right] - \lambda_j \tag{22}$$

We see at once that setting the natural gradient to zero is equivalent to setting the gradient to zero. Moreover, the natural gradient is quicker to compute for gradient ascent algorithms, since we do not need to compute the Hessian of the log normalizer $\nabla^2_{\lambda_j}(\alpha(\lambda_j))$. The natural gradient is also theoretically well motived, as symmetrized KL divergence provides a more intuitive "distance" between probability distributions [35].

For ease of exposition, we will often choose variational parameters which are not the canonical exponential family parameters. For example, if $X \sim \text{Bernoulli}(p)$, it is much more intuitive to work with p as our variational parameter instead of $\eta(p) = \text{logit}(p) \triangleq \log\left(\frac{p}{1-p}\right)$. This is not a problem: we just need to remember to compute our (natural) gradients with respect to the natural parameter, and transform back afterwards.

6.3 Mean-Field Updates For θ, β, and ϕ

In this section, we compute the natural gradients of the ELBO with respect to the variational parameters. Again, we emphasize that the updates for the community membership parameters are agnostic to the choice of community membership parameters.

Let's begin by finding the updates for ϕ, γ and λ. We have that

$$\mathcal{L}_1(q) \quad = \quad \mathbb{E}_{q(\iota)}\left[\mathbb{E}_{q(r,x,\beta,\theta)}\left[\log p(\mathbf{r},\mathbf{x},\beta,\theta|\iota) - q(\mathbf{x},\beta,\theta)\right]\right] \tag{23}$$

where $q(\iota)$ and $q(x,\beta,\theta)$ are the variational distributions of ι and x,β,θ respectively. Recall the variational parameters ϕ, γ and λ for x, β, and θ respectively. Under mild conditions stipulated by the dominated convergence theorem [39], the gradient commutes with expectations so that

$$\nabla_{\phi,\gamma,\lambda}\mathcal{L}_1(q) \quad = \quad \mathbb{E}_{q(\iota)}\left[\nabla_{\phi,\gamma,\lambda}\mathbb{E}_{q(x,\beta,\theta)}\left[\log p(\mathbf{r},\mathbf{x},\beta,\theta|\iota) - q(\mathbf{x},\beta,\theta)\right]\right] \tag{24}$$

while clearly $\nabla_{\phi,\gamma,\lambda}\mathcal{L}_2(q) = 0$. By abuse of notation, denote $\mathcal{L}_1(q|\iota)$ to be the quantity inside the expectation over $q(\iota)$ given above. We then have that $\nabla_{\phi,\gamma,\lambda}\mathcal{L}_1(q) = \mathbb{E}_{q(\iota)}\left[\nabla_{\phi,\gamma,\lambda}(\mathcal{L}_1(q|\iota)\right]$.

Equations 10 and 11 show that, given ι, the variables x, is conditionally conjugate and in the exponential family. Thus, the natural gradients can computed by subtracting the natural parameter from the expectation of the natural parameter.

Following the computation in [2], we the facts that rate and shape of a gamma distribution are its natural parameters and that the expectation of a random variable $X \sim \text{Gamma}(a, b)$ is $\frac{a}{b}$ to compute

$$\begin{aligned}\widehat{\nabla}_{\gamma_{uk}}\left(\mathcal{L}_1(q)|\iota\right) = &\quad \langle a + \sum_i r_{ui}\phi_{uik} - \gamma^{\text{shp}}_{uk}, b + \sum_i \lambda^{\text{shp}}_{ik}/\lambda^{\text{rt}}_{ik}\iota_{ui} - \gamma^{\text{rte}}_{uk}\rangle \\ \widehat{\nabla}_{\lambda_{ik}}\left(\mathcal{L}_1(q)|\iota\right) = &\quad \langle c + \sum_i r_{ui}\phi_{uik} - \lambda^{\text{shp}}_{ik}, d + \sum_i \gamma^{\text{shp}}_{uk}/\gamma^{\text{rt}}_{uk}\iota_{ui} - \lambda^{\text{rte}}_{ik}\rangle\end{aligned} \tag{25}$$

We only need to update those ϕ_{ui} for which $r_{ui} > 0$. In this case, we take the gradient with respect to the natural parameter $\log\phi_{ui} = \langle\log\phi_{ui}, \ldots, \phi_{uiK}\rangle$. We compute

$$\widehat{\nabla}_{\log\phi_{ui}}\left(\mathcal{L}_1(q)|\iota\right)_k \quad = \quad \frac{1}{N(\gamma,\lambda)}\exp\{\Psi(\gamma^{\text{shp}}_{uk}) - \log\gamma^{\text{rte}}_{uk} + \Psi(\lambda ik^{\text{shp}}) - \log\lambda^{\text{rte}}_{ik}\} - \log\phi_{uik} \tag{26}$$

where $N(\gamma,\lambda)$ is a normalizing constant ensuring that $\sum_k \phi_{uik} = 1$. We see that only the the rate parameters for γ and λ depend on ι. Taking expectations over $q(\iota)$ gives

$$\begin{aligned}\widehat{\nabla}_{\gamma_{uk}}\left(\mathcal{L}_1(q)|\iota\right) = &\quad \langle a + \sum_i r_{ui}\phi_{uik} - \gamma^{\text{shp}}_{uk}, b + \sum_i \lambda^{\text{shp}}_{ik}/\lambda^{\text{rt}}_{ik}s_{ui} - \gamma^{\text{rte}}_{uk}\rangle \\ \widehat{\nabla}_{\lambda_{ik}}\left(\mathcal{L}_1(q)|\iota\right) = &\quad \langle c + \sum_i r_{ui}\phi_{uik} - \lambda^{\text{shp}}_{ik}, d + \sum_i \gamma^{\text{shp}}_{uk}/\gamma^{\text{rt}}_{uk}s_{ui} - \lambda^{\text{rte}}_{ik}\rangle\end{aligned} \tag{27}$$

where we recall that that $\mathbb{E}_{q(\iota)}\left[\iota_{ui}\right] = s_{ui}$. For Mean-Field inference, this yields the updates

$$\begin{aligned}\gamma_{uk} \leftarrow &\quad \langle a + \sum_i r_{ui}\phi_{uik}, b + \sum_i \lambda^{\text{shp}}_{ik}/\lambda^{\text{rt}}_{ik}s_{ui}\rangle \\ \lambda_{ik} \leftarrow &\quad \langle c + \sum_i r_{ui}\phi_{uik}, d + \sum_i \gamma^{\text{shp}}_{uk}/\gamma^{\text{rt}}_{uk}s_{ui}\rangle\end{aligned} \tag{28}$$

Since the complete conditional for x_{uik} are the same in the ZIPF and BPF models, we update ϕ_{ui} as in [2]:

$$\phi_{ui} \propto \exp\{\Psi(\gamma^{\text{shp}}_{uk}) - \log\gamma^{\text{rte}}_{uk} + \Psi(\lambda ik^{\text{shp}}) - \log\lambda^{\text{rte}}_{ik}\}$$

6.4 Mean-Field Inference for ι and Community Membership parameters

The next step is to compute the gradient and natural gradient with respect to the variational parameters s for ι. Taking deriviatives with respect to the natural variational parameter $\text{logit}(s_{ui})$, we have

$$\nabla_{s_{ui}}\mathcal{L}_1(q) \quad = \quad \mathbb{E}_{q(x,\theta,\beta)}\nabla_s\left[\mathbb{E}_{q(\iota)}\left[\log p(r, x, \theta, \beta|\iota)\right)\right]$$

where we recall that $q(x, \beta, \theta)$ has no ι dependence. If $r_{ui} > 0$, then $p(r, x, \theta, \beta|\iota)$ has no ι dependence. If $r_{ui} = 0$, then

$$p(r_{ui} = 0, x, \theta, \beta|\iota_{ui} = 0) = 1 \quad p(r_{ui} = 0, x, \theta, \beta|\iota_{ui} = 1) = \exp(-\theta_u^T\beta_i)$$

Thus, $\log p(r, x, \theta, \beta|\iota) = \sum_{ui:r_{ui}=0}\iota(-\theta_u^T\beta_i)$, up to a constant independent of ι. For a user item pair (u, i) for which $r_{ui} = 0$, we therefore have

$$\begin{aligned}\nabla_{\text{logit}(s_{ui})}\mathcal{L}_1(q) &= \mathbb{E}_{q(x,\theta,\beta)}\nabla_{\text{logit}(s_{ui})}\left[\mathbb{E}_{q(\iota)}\left[\iota(-\theta_u^T\beta_i)\right]\right] \\ &= -\nabla_{\text{logit}(s_{ui})}\mathbb{E}_{q(\iota)}\left[\iota_{ui}\right]\mathbb{E}_{q(x,\theta,\beta)}\left[\theta_u^T\beta_i\right] \\ &= -\nabla_{\text{logit}(s_{ui})}\mathbb{E}_{q(\iota)}\left[\iota_{ui}\right]\sum_k\frac{\gamma^{\text{shp}}_{uk}\lambda^{\text{shp}}_{ik}}{\gamma^{\text{rte}}_{uk}\lambda^{\text{rte}}_{ik}}\end{aligned} \tag{29}$$

For the exponential family, the expectation of the sufficient statistic is just the gradient of the log normalizer $\alpha(\cdot)$ with respect to the natural parameter $\eta(\cdot)$. Since the sufficient statistics of the Bernoulli random variable is just itself, we have that $\nabla_{\mathrm{logit}(s_{ui})}\mathbb{E}_{q(\iota)}\left[\iota\right] \; = \nabla^2_{\mathrm{logit}(s_{ui})}\alpha(\mathrm{logit}(s_{ui})$. Therefore, the gradient and natural gradients with respect to the natural variational parameter logits_{ui} are given by

$$\nabla_{\mathrm{logit}(s_{ui})}\mathcal{L}_1(q) = -\nabla^2_{\mathrm{logit}(s_{ui})}\alpha(\mathrm{logit}(s_{ui})\sum_k \frac{\gamma^{\mathrm{shp}}_{uk}\lambda^{\mathrm{shp}}_{ik}}{\gamma^{\mathrm{rte}}_{uk}\lambda^{\mathrm{rte}}_{ik}} \quad \text{and} \quad \widehat{\nabla}_{\mathrm{logit}(s_{ui})}\mathcal{L}_1(q) = -\sum_k \frac{\gamma^{\mathrm{shp}}_{uk}\lambda^{\mathrm{shp}}_{ik}}{\gamma^{\mathrm{rte}}_{uk}\lambda^{\mathrm{rte}}_{ik}} \tag{30}$$

We see then that the natural gradients of $\mathcal{L}_1(q)$ with respect to the variational parameters are always tractable, regardless of our choice of a community membership parameters ξ. This makes sense, since the graphical model shows that r, x, θ, β and ξ are conditionally independence given ι. Inference is therefore tractable as long as the natural gradients of $\mathcal{L}_2(q)$ with respect to s and ζ are tractable.

Before computing the natural gradients of $\mathcal{L}_2$ with respect to the variational parameters governing ι and ξ, we rewrite $\mathcal{L}_2(q)$ in a more suggestive form. Recall that $\mathcal{L}_2(q) = \mathbb{E}_q\left[\log p(\iota,\xi)\right] - \mathbb{E}_q\left[\log q(\iota,\xi)\right]$ has no dependence on θ, β, or x. Thus, we may write

$$\begin{aligned}\mathcal{L}_2(q) &= \mathbb{E}_{q(\iota,\xi)}\left[\log p(\iota,\xi)\right] - \mathbb{E}_q\left[\log q(\iota,\xi)\right] \\ &= \mathbb{E}_{q(\iota)}\left[\mathbb{E}_{q(\xi)}\left[\log p(\xi,\iota) - q(\xi)\right] - q(\iota)\right] \\ &= \mathbb{E}_{q(\iota)}\left[\mathcal{L}_2(q|\iota) - q(\iota)\right]\end{aligned} \tag{31}$$

where we denote $\mathcal{L}_2(q|\iota) \triangleq \mathbb{E}_{q(\xi)}\left[\log p(\xi,\iota) - q(\xi)\right]$ the *stand-alone objective.* Indeed, we see that $\mathcal{L}_2(q|\iota)$ is simply for the ELBO for a stand-alone community membership model (that is, one without rating information) where the latent variables are ξ and observed links (fixed observations of) ι_{ui}.

Writing $\mathcal{L}_2$ in the above form therefore permits more sophisticated community membership models for which the ELBO is intractable, and which optimize a weaker lower bound as a variational objective. For example, the AMP model [4] which we will introduce later on this paper optimizes a lower bound on the ELBO obtained by Jensen's inequality. In this case, we can simply substitute the weaker bound $\mathcal{L}'_2(q|\iota)$ in for $\mathcal{L}_2(q|\iota)$. We may assume without loss of generality that our stand-alone objective is a lower bound $\mathcal{L}'_2(q|\iota) \leq \mathcal{L}_2(q|\iota)$, since clearly $\mathcal{L}_2(q|\iota)$ is a lower bound of itself.

At this stage, we take the natural gradients with respect to

$$\mathcal{L}_2(q)' \quad \triangleq \quad \mathbb{E}_{q(\iota)}\left[\mathcal{L}'_2(q|\iota) - q(\iota)\right] \tag{32}$$

Placing the gradient inside the expectation and neglecting constants in ξ_j, we get that $\widehat{\nabla}_\xi\mathcal{L}_2(q)' = \mathcal{E}_{q(\iota)}\left[\widehat{\nabla}_\xi\mathcal{L}'_2(q|\iota)\right]$. Many Bayesian community membership models already provide calculations of the natural gradient of the stand-alone object $\mathcal{L}'_2(q|\iota)$, as the latter is just the objective function obtained by treating the links ι as fixed observations [5, 42, 4]. Thus, computing natural gradients with respect to $\mathcal{L}_2(q')$ amounts to taking the expectation the natural gradients provided in the literature.

We shall refer to the (natural) gradients and natural gradients of $\mathcal{L}_2(q|\iota)'$ with respect to the appropriate parameters as *stand-alone (natural) gradients*, and the (natural) gradients with respect to $\mathcal{L}'(q)$ (which are just the gradients with respect to $\mathcal{L}'_2(q)$) as *complete (natural) gradients.*

For an item pair $(u,i) \notin \mathcal{R}$, the complete gradient with respect to logit(s_{ui}) is computing by

$$\begin{aligned}\nabla_{\mathrm{logit}(s_{ui})}\mathcal{L}_2(q) &= \mathbb{E}_{q(\iota)}\left[\nabla_{\mathrm{logit}(s_{ui})}\mathcal{L}'_2(q|\iota)\right] - \nabla_{\mathrm{logit}(s_{ui})}\mathbb{E}_{q(\iota)}\left[q(\iota)\right] \\ &= \mathbb{E}_{q(\iota)}\left[\nabla_{\mathrm{logit}(s_{ui})}\mathcal{L}'_2(q|\iota)\right] - \nabla^2_{\mathrm{logit}(s_{ui})}(\alpha(\mathrm{logit}(s_{ui}))\mathrm{logit}(s_{ui})\end{aligned} \tag{33}$$

used the facts that logit(s_{ui}) is the natural variational parameter for ι, and that the expectation of the sufficient statistic is the gradient of the log normalizer, as in equation 18. The complete natural gradient is then

$$\widehat{\nabla}_{\mathrm{logit}(s_{ui})}\mathcal{L}_2(q)' \quad = \quad \nabla^2_{\mathrm{logit}(s_{ui})}(\alpha(\mathrm{logit}(s_{ui}))^{-1}\mathbb{E}_{q(\iota)}\left[\nabla_{\mathrm{logit}(s_{ui})}\mathcal{L}'_2(q|\iota)\right] - \mathrm{logit}(s_{ui}) \tag{34}$$

In the case that $\mathcal{L}_2'(q|\iota)$ is equal to $\mathcal{L}_2(q|\iota)$, up to a constant independent of ι, we have that

$$\begin{aligned}&\nabla_{\text{logit}(s_{ui})}\mathcal{L}_2(q)' = \nabla^2_{\text{logit}(s_{ui})}\alpha(\text{logit }(s_{ui}))\left[\mathbb{E}_q\left[\text{logit}(\iota_{ui}|\xi)\right] - \text{logit}(s_{ui})\right] \quad \text{and} \\ &\widehat{\nabla}_{\text{logit}(s_{ui})}\mathcal{L}_2(q)' = \mathbb{E}_q\left[\text{logit}(\iota_{ui})|\xi)\right] - \text{logit}(s_{ui})\end{aligned} \tag{35}$$

so that the natural gradient is tractable as long as $\mathbb{E}_q\left[\text{logit}(\iota_{ui})|\xi)\right]$ is easily computable. The complete natural gradient in this case is just

$$\nabla_{\text{logit}(s_{ui})}(\mathcal{L}_1(q) + \mathcal{L}_2(q)') \quad = \quad \mathbb{E}_q\left[\text{logit}(\iota_{ui})|\xi)\right] - \text{logit}(s_{ui}) + \sum_k \frac{\gamma^{\text{shp}}_{uk}\lambda^{\text{shp}}_{ik}}{\gamma^{\text{rte}}_{uk}\lambda^{\text{rte}}_{ik}} \tag{36}$$

Note that this quantity could have been computed, perhaps more easily, by taking the gradient of the full ELBO $\mathcal{L}_q$ with respect to logit(s_{ui}). However, this exposition aims to separate the rating model objective $\mathcal{L}_1$ from the community model objective $\mathcal{L}_2$ to emphasize the portability of ZIPF, and allow for weaker community membership objectives $\mathcal{L}_2(q)'$, of which we shall make use in the follow section.

7 AMP-ZIPF

One model which satisfies the tractability conditions described is the *Assortative Mixed Membership Stochastic Block Model with Node Popularities*, or AMP [4]. We say item i and user u form a *link* if $\iota_{ui} = 1$, that is, if the rating is observed; otherwise, i and u constitute a *nonlink.* In many community membership models, including AMP, links are taken as observations [28, 5, 4]. In the setting of ZIPF, however, the presence of a link is "semi-random"; if $(u, i) \in \mathcal{R}$, then we know for certain that i and u are linked, but if $r_{ui} = 0$, then the link is the event $\iota_{ui} = 1$.

The AMP model the links in a network are described by varying degrees of membership in K' communities[4] [4]. Each community is assigned a community strength parameter ψ_k, and each node is given both a "popularity" parameter, which up- or downweights the probability of that node linking to a neighbhor, and a community membership parameter describing the relative strengths of affiliation a user has to a given community.

To better capture the bipartite structure of the user-item graph, we treat popularities and community memberships for user-nodes and item notes separately. That is, we is assign a community membership π_u and π_i parameter, determining how strongly each users and item, respectively, belongs to the latent communities. We also assign "node popularity" parameters ξ_u and ξ_i, which affects how likely a user is to give a review, and how likely an item is to be reviewed. We then draw "interaction parameters" $z_{u\to i}$ and $z_{i\to u}$ from the categorical distributions over π_u and π_i respectively. Informally, the interaction parameters correspond to the community membership which manifests itself in the presence or absence of a link between u and i: if $z_{u\to i} = z_{i\to u}$, the probability of an observed link is greater than if $z_{u\to i} \neq z_{u\to i}$. More formally, the probability of observing a link ι_{ui} is [4]

$$\text{logit }(p(\iota_{ui} = 1|z_{u\to i}, z_{i\to u}, \psi, \xi)) \equiv \xi_u + \xi_i + \sum_{k=1}^{K'} \delta^k_{ui}\psi_k \tag{37}$$

where δ^k_{ui} is the indicator that $z_{u\to i} = z_{i\to u} = k$. For ease of notation, set $\widetilde{x}_{ui} = \xi_u + \xi_i + \sum_{k=1}^{K'} \delta^k_{ui}\psi_k$.

The generative process is given by:

1. Draw K' community strengths $\psi_k \sim \mathcal{N}(\mu_0, \sigma_0^2)$
2. For each user u
   (a) Draw community memberships $\pi_u \sim \text{Dirichlet}(\alpha_U)$
   (b) Draw user node popularity $\xi_u \sim \mathcal{N}(0, \sigma^2_{1,U})$

[4] Here, the "prime" distinguish between the number K' of communities in the AMP model, and the number K of latent features in the Poisson factorization review model

3. For each item i
   (a) Draw community memberships $\pi_i \sim \text{Dirichlet}(\alpha_I)$
   (b) Draw user node popularity $\xi_i \sim \mathcal{N}(0, \sigma_{1,I}^2)$
4. For each user/item pair,
   (a) Draw interaction indictaor $z_{u\to i} \sim \text{Categorical}(\pi_u)$
   (b) Draw interaction indictaor $z_{i\to u} \sim \text{Categorical}(\pi_i)$
   (c) Draw a link $\iota_{ui} \sim \text{Bernoulli}(\text{logit}^{-1}(\widetilde{x}_{ui}))$.

7.1 Variational Objective

To distinguish from variational parameters for the rating model, we shall use capital Greek letters for the variational parameters of the community membership model. Following [4], we approximate the posterior with the following family of distributions

$$\begin{aligned} q(z_{u\to i} = k_1, z_{i\to u} = k_2) = \Phi_{ui}^{k_1 k_2}; \quad q(\pi_u) = \text{Dirichlet}(\pi_u; \Gamma_u); \quad q(\pi_i) = \text{Dirichlet}(\pi_i; \Gamma_i); \\ q(\psi_k) = \mathcal{N}(\psi_k; \mu_k, \sigma_\psi^2); \quad q(\xi_u) = \mathcal{N}(\xi_u; \Lambda_u, \sigma_{\xi,U}^2); \quad q(\xi_i) = \mathcal{N}(\xi_i; \Lambda_i, \sigma_{\xi,I}^2) \end{aligned} \tag{38}$$

The standalone ELBO is then given by

$$\begin{aligned} \mathcal{L}_2(q|\iota) = & \sum_u \mathbb{E}_q\left[\log p(\pi_u|\alpha_U)\right] - \sum_u \mathbb{E}_q\left[\log q(\pi_u|\Gamma_u)\right] \\ & + \sum_i \mathbb{E}_q\left[\log p(\pi_i|\alpha_I)\right] - \sum_i \mathbb{E}_q\left[\log q(\pi_i|\Gamma_i)\right] \\ & + \sum_u \mathbb{E}_q\left[\log p(\xi_u|\sigma_{1,U}^2)\right] - \sum_u \mathbb{E}_q\left[\log q(\xi_u|\Lambda_u, \sigma_{\xi,U}^2)\right] \\ & + \sum_i \mathbb{E}_q\left[\log p(\xi_i|\sigma_{1,I}^2)\right] - \sum_{ui} \mathbb{E}_q\left[\log q(\xi_i|\Lambda_i, \sigma_{\xi,I}^2)\right] \\ & + \sum_k \mathbb{E}_q\left[\log p(\psi_k|u_0, \sigma_0^2)\right] - \sum_k \mathbb{E}_q\left[\log q(\psi_k|\mu_k, \sigma_\psi^2)\right] \\ & + \sum_{ui}\left[\mathbb{E}_q\left[\log p(z_{u\to i}|\pi_u)\right] + \mathbb{E}_q\left[\log p(z_{i\to u}|\pi_i)\right] - \mathbb{E}_q\left[\log q(z_{u\to i}, z_{i\to u}|\Phi_{ui})\right]\right] \\ & + \sum_{ui} \mathbb{E}_q\left[\log p(\iota_{ui}|z_{u\to i}, z_{i\to u}), \psi, \xi\right] \end{aligned} \tag{39}$$

The first five lines capture the *global parameters* Γ, Λ, μ. The last two lines contain the *local parameters* Φ_{ui}. As it stands, the ELBO is intractable to optimize, as we cannot differentiate the term in the last line. We instead maximize a lower bound $\mathcal{L}_2'(q|\iota)$ on the ELBO by applying Jensen's equality to the last term [4]

$$\mathbb{E}_q\left[\log p(\iota_{ui}|z_{u\to i}, z_{i\to u}, \psi, \xi)\right] \quad = \quad \iota_{ui}\mathbb{E}_q\left[\widetilde{x}_{ui}\right] - \mathbb{E}_q\left[\log(1 + \exp(\widetilde{x}_{ui}))\right] \tag{40}$$

By the concavity of log, we get that

$$\begin{aligned} -\mathbb{E}_q\left\{\log(1 + \exp(\widetilde{x}_{ui}))\right] \geq & -\log\left[\mathbb{E}_q\left[1 + \exp\widetilde{x}_{ui}\right]\right\} \\ & = -\log\{1 + \mathbb{E}_q[\exp(\xi_u + \xi_i + \sum_{k=1}^{K'} \delta_{ui}^k \psi_k)]\} \\ & = -\log\left\{1 + \mathbb{E}_q\left[\exp\left(\Gamma_u + \sigma_{\xi,U}^2/2\right)\exp\left(\Gamma_i + \sigma_{\xi,I}^2/2\right)t_{ui}\right]\right\} \end{aligned} \tag{41}$$

where $t_{ui} \equiv \sum_{k=1}^{K'} \Phi_{ui}^k k \exp(\mu_k + \sigma_\psi^2/2) + \left(1 - \sum_{k=1}^{K'} \Phi_{ui}^{kk}\right)$ (for details, see [4]).

Observe that we lower bounded a term in the ELBO which *did not depend on* ι_{ui}. Thus, $\mathcal{L}_2'(q|\iota)$ agrees with $\mathcal{L}_2(q|\iota)$ up to a constant in ι. Consequently, the natural gradient of $\mathcal{L}_2(q') = \mathbb{E}_{q(\iota)}\left[\mathcal{L}_2'(q|\iota) - p(\iota|z, \psi, \xi)\right]$ with respect to $\operatorname{logit}(s_{ui})$ is given by[5]

$$\begin{aligned}
\widehat{\nabla}_{\operatorname{logit}(s_{ui})}\mathcal{L}_2'(q) &= \mathbb{E}_q\left[\operatorname{logit}(p(\iota_{ui}|z_{u\to i}, z_{i\to u}, \xi, \psi))\right] - \operatorname{logit}(s_{ui}) \\
&= \mathbb{E}_q\left[\widetilde{x}_{ui}\right] - \operatorname{logit}(s_{ui}) \\
&= \Lambda_i + \Lambda_u + \sum_k \mu_k \mathbb{E}_q\left[\delta_{ui}^k\right] - \operatorname{logit}(s_{ui}) \qquad (42)\\
&= \Lambda_i + \Lambda_u + \sum_k \mu_k \Phi_{ui}^{kk} - \operatorname{logit}(s_{ui})
\end{aligned}$$

Combing with equation 30 gives the update

$$s_{ui} \leftarrow \operatorname{logit}^{-1}\{\Lambda_i + \Lambda_u + \sum_k \mu_k \Phi_{ui}^{kk} - \sum_k \frac{\gamma_{uk}^{\text{shp}}\lambda_{ik}^{\text{shp}}}{\gamma_{uk}^{\text{rte}}\lambda_{ik}^{\text{rte}}}\} \qquad (43)$$

7.2 Community Membership Model Updates

We now derive a non-stochastic gradient descend algorithm to minimize the variational objective given in the previous subsection. We hope the following calculations provide a blueprint for deriving community-membership coordinate ascent updates in generic ZIPF model. The guiding inuition is that we take can simply choose community presented in the literature, which treats links $\iota_{ui} = 1$ between users and items as fixed observations, and instead views links as random variables governed by Bernoulli parameters s_{ui}.

As shown in section, optimal updates are obtained by finding the updates for the fixed-observation model - the stand alone updates for $\mathcal{L}_2'(q|\iota)$, and then taking the expectation of these updates, as functions of the link indicators ι_{ui}, over the variational link-distribution $q(\iota)$. Taking the stand-alone AMP model in [4], we find that the complete updates are very straight-forward updates[6]. The stand alone updates for Γ have no ι dependency, and the updates for μ and Λ linear in the variables ι_{ui}. It then follows from the linearity of expectations that the complete updates for the latter two variables are obtained by just replacing each appearance of ι_{ui} in the stand-alone update with s_{ui}.

Let $\partial_{\cdot|\iota}\Gamma_{u/i}$ denote the standalone natural gradients of $\mathcal{L}_2'(q|\iota)$ with respect to $\Gamma_{u/i}$, and let $\partial_{\cdot|\iota}\Lambda_{u/i}$ and $\partial_{\cdot|\iota}\mu_k$ be the gradients of $\mathcal{L}_2'(q|\iota)$ with respect to $\Lambda_{u/i}$ and μ_k, respectively. From the updates in [4], we have

$$\begin{aligned}
\partial\Gamma_{uk} &= -\Gamma_{uk} + \alpha_k + \sum_i \phi_{ui}^{kk} \quad \text{and} \quad \partial\Gamma_{ik} = -\Gamma_{ik} + \alpha_k + \sum_u \phi_{ui}^{kk} \\
\partial\Lambda_u &= -\frac{\Lambda_u}{\sigma_{1,U}^2} + \sum_i (s_{ui} - l_{ui}t_{ui}) \quad \text{and} \quad \partial\Lambda_i = -\frac{\Lambda_u}{\sigma_{1,I}^2} + \sum_u (s_{ui} - l_{ui}t_{ui}) \qquad (44)\\
\partial\mu_k &= \frac{\mu_0 - \mu_k}{\sigma_0^2} + \sum_{ui} \Phi_{ui}^{kk}\left(s_{ui} - l_{ui}\exp\{\mu_k + \sigma_\psi^2/2\}\right)
\end{aligned}$$

where we have defined

$$l_{ui} \triangleq \frac{\exp\left(\Lambda_u + \sigma_{\psi,U}^2/2\right)\exp\left(\Lambda_i + \sigma_{\psi,I}^2/2\right)}{1 + \exp\left(\Lambda_u + \sigma_{\psi,U}^2/2\right)\exp\left(\Lambda_i + \sigma_{\psi,I}^2/2\right)t_{ui}} \qquad (45)$$

The natural gradients with respect to the local variables $\Phi_{ui}^{k_1k_2}$ can be derived in a similar fashion:

$$\begin{aligned}
\partial\Phi_{ui}^{kk} &= \frac{1}{N_{ui}|\iota}\exp\{\Psi(\Gamma_{ik}) - \Psi(\Gamma_{i0}) + \Psi(\Gamma_{uk}) - \Psi(\Gamma_{i,0})\} \\
&\quad \times (s_{ui}\exp(\mu_k) + 1 - s_{ui})\exp\{-l_{ui}(\exp\{\mu_k + \sigma_\psi^2/2\} - 1)\} - \Phi_{ui}^{kk} \qquad (46)\\
\partial\Phi_{ui}^{k_1k_2} &= \frac{1}{N_{ui}}\exp\{\Psi(\Gamma_{i,k_1}) - \Psi(\Gamma_{i0}) + \Psi(\Gamma_{u,k_2})\Psi(\Gamma_{u0})\} - \Phi_{ui}^{k_1k_2} \qquad k_1 \neq k_2
\end{aligned}$$

[5] We drop dependence on π due to the Independence structure implied by the graphical model;see [40]

[6] The updates shown in [4] are intended for stochastic gradient descent, where updates are computed for a small subset S of nodes. Setting S to be the set of all items and users gives the non-stochastic gradient ascent algorithm presented here

Where N_{ui} is a normalization constant ensuring that $\sum_{k_1 k_2} \Phi_{ui}^{k_1 k_2} = 1$.

7.3 Initialization and Convergence

To maximize the variational objective $\mathcal{L}'$, we compute coordinate ascent updates in parallel. This amounts to optimizing both the local parameters ϕ_{ui} and Φ_{ui}, as well as global parameters γ_{uk}, λ_{ik}, and Γ, so that the natural gradient of $\mathcal{L}'$ with respect to these each of these parameters, separately, is 0. We then increment the parameters μ and Λ along their gradients:

$$\mu_k^{(t)} \leftarrow \mu_k^{(t-1)} + \rho_\mu^{(t)} \partial \mu_k \quad \text{and} \quad \Lambda_{u/i}^{(t)} \leftarrow \Lambda_{u/i}^{(t-1)} + \rho_\Lambda^{(t)} \partial \Lambda_{u/i} \tag{47}$$

where ρ_μ and ρ_Λ are the learning. The algorithm is non-stochastic, so that the learning rates need not be adjusted.

```
Data: Nonnegative Integer Reviews $r_{ui}$
Initialize parameters following 7.3
while not yet converged do
    local step
    Optimize $\phi_{ui}$, $s_{ui}$, and $\Phi_{ui}$ for all $(u, i) \in P$ using Eqs. 29, 43 and 46
    global step
    Update user and item latent features $\beta_u$ and $\beta_i$ for all $u$ and $i$ by Eq. 28
    Update community memberships $\Gamma_{u/i}$, using natural gradients in Eq. 44.
    Update community popularities $\Lambda_{u/i}$, using gradient in Eq. 44.
    Update community strengths $\mu$ using stochastic natural gradient in Eq. 44
end
```

Algorithm 1: Non-Stochastic Variational Inference for AMP-ZIPF

Initialization is performed along the lines of [2] and [4]. The hyperparameters a, b, c, d for $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ are set to .3, plus a small random offset. We then build a network $\mathcal{N}$ on users and items with a link whenever a review exists for a given user/item pair. First, we feed $\mathcal{N}$ to the variational inference algorithm for MMSB [5], and use the approximate posterior community memberships to initialize $\boldsymbol{\Gamma}$, plus a small random offset. Node popularities $\boldsymbol{\Lambda}$ for each user and item are set to the logarithm of the number of corresponding reviews, normalized by the greatest number of reviews provided by a single user, or a of a single item, with a small random offset added. Community strengths $\boldsymbol{\mu}$ are initialized to 0, and zero-inflation paramters s_{ui} are set to 1 if $r_{ui} > 0$, and some $\epsilon > 0$ otherwise. For non-stochastic inference, we use constant step sizes ρ.

In practice, the stability of coordinate ascent is contingent upon the appropriate choice of model variances. Following [4], we set $\sigma_{1,U} = \sigma_{1,I} = 10$, $\sigma_0 = 0$, and $\sigma_{\psi,U} = \sigma_{\psi,I} = .01$. The ZIPF-AMPF the coordinate updates involve frequent exponentiation, so poor choices of variances can lead to numerous overflow and underflow errors.

7.4 Numerical Results

As a proof of concept, we implemented a non-stochastic version of AMP-ZIPF. As the non-stochastic implementation has $O(NMK^2)$ complexity, we use a rather small subset of the Yelp dataset, consisting of 4734 users and 1214 restaurants, split between the New England and New York Areas. We ran the data set with $K = 30$ latent features, and $K' = 4$ latent communities. The data set is quite sparse, contaning only 15148 reviews. We were able to achieve an increasing ELBO and increasing rating likelihoods. The community membership parameters were initialized to constants up to small random offests, and *were not* set to approximate parameters from [5]. However, node popularities were set following subsection 7.3.

We compared non-stochastic ZIPF to BPF on the above dataset, plotting the log posterior likelikehood of the observed reviews; that is $\sum_{(u,i) \in \mathcal{R}} r_{ui} * \log(\theta_u^T \beta_i) - \theta_u^T \beta_i$ [7]. Though each ZIPF iteration takes more time to compute than BPF, the per-iteration increase in posterior likelihood is far more rapid in ZIPF. Plots are provided in Appendix F.

[7] Here, we drop a constant factor $\log(r_{ui}!)$, which does not depend on the variational parameters

As expected, ZIPF has an even more pronounced edge over BPF when we compute the likelihoods for nonobserved reviews, that is

$$\mathcal{L}_0^{BPF} \triangleq -\sum_{(u,i)\notin\mathcal{R}} \theta_u^T\beta_i \quad \text{and} \quad \mathcal{L}_0^{ZIPF} \triangleq \sum_{(u,i)\notin\mathcal{R}} 1 - s_{ui} + s_{ui}\theta_u^T\beta_i \tag{48}$$

We also compared the mean features for restaurants in the two geographic areas and found that the mean features for ZIPF, plotted in Figure 9, Appendix F are far less geographically determined than those for BPF, show in Figure 10. We note that Figure 10 shows two New England peaks, perhaps corresponding to restaurants in Providence and Boston, and many smaller New York peaks, which possibly correspond to many neighborhoods in Manhattan.

Due to technical difficulties involving the XCode software and time constraints, we were not able to compare heldout and test set likelihoods and precisions. In future work, we hope to more comprehensively test ZIPF against existing collaborative filtering methods. To this, we have already implemented a java code which can compute several measures of recommendation system accuracy, including Kendell's τ and Pearsons ρ [7, 27]. These two metrics measure the relative rankings of heldout reviews against the existing reviews. Since ZIPF attempts to predict expected ratings for unreviewed items, conventional methods such as root mean squared error [7] or KL Divergence [18] for measuring recomendation system accuracy do not apply, and relative ranking metrics are preferable.

8 Stochastic Variational Inference Algorithm for ZIPF-AMP

8.1 Inference, Complexity, and Stochastic Variational Inference

On its own, BPF is very computationally efficient. Local parameters ϕ_{ui} only need to be updated when $r_{ui} > 0$, and can thus be updated in $O(RK)$ time, where R is the number of reviews. Since most review data sets are sparse, BPF shows considerable performance advances over conventional matrix factorization techniques [18], as it updates the variational parameters in $O(RK)$ time and $O([N+M]\,K)$ space [2].

Unfortunately, ZIPF is prohibitively expensive on realistically large datasets. Computing the user-item interations Φ_{ui} completes in $O(NMK'^2)$ time and storing the diagonal elements of Φ_{ui} requires (NMK') space. ZIPF therefore does not benefit from sparsity to the same extent as BPF. This is somewhat expected: whereas BPF and Matrix Factorization try to predict user behavior, the goal of ZIPF is to "fill in the blanks": ZIPF attempts both to learn the community structure which is responsible for observed sparsity, and then generate recommendations when a user or items community affiliations change. In this sense, both the presence and absence of a review provides ZIPF with further information.

To overcome the performance limitations, we will describe a stochastic variational inference (SVI) algorithm. In SVI, we optimize the variational objective at time t by taking steps of size $\rho^{(t)}$ along noisy, yet unbiased, estimations of the gradient [3]. As long as our step sizes ρ satisfy the conditions $\sum_t \rho^{(t)} = \infty$, $\sum(\rho^{(t)})^2 < \infty$, then we converge to a local optimum [3, 5]. Stochastic variational inference is quite powerful, since noisy estimates of the gradient are frequently much cheaper to compute. As such, is SVI is used to learn many cutting-edge network models [5, 42], including AMP [4].

8.2 Set-Based Sampling

In set-based sampling, we sample a subset S_t of the total users and items, and update the appropriate gradients corresponding to the global parameters for users and items in S_t. We need not sample the sets uniformly: As long as the union of all sets sampled with non-zero probaility $\bigcup S_t$ contains every item and every user, and as long as each user-item pair occurs in a constant, positive number of sets $c \geq 1$, we can weight our coordinate ascent updates appropriately to obtain unbiased estimates of the true gradient [5].

For example, if $h(t)$ is a probability distribution over the sets, then we can re-write the last line in equation 39 as

$$\sum_{ui} \mathbb{E}_q\left[\log p(\iota_{ui}|z_{u\to i}, z_{i\to u}), \psi, \xi\right] = \mathbb{E}_h\left[\frac{1}{ch(t)} \sum_{u,i\in S} \mathbb{E}_q\left[\log p(\iota_{ui}|z_{u\to i}, z_{i\to u}), \psi, \xi\right]\right] \tag{49}$$

In this case, we can take unbiased estimates $\partial^t_{\cdot|\iota}\Gamma_{uk}, \partial^t_{\cdot|\iota}\Gamma_{ik}, \partial\Lambda^t_u, \partial\Lambda^t_i$ are just the nonstochastic gradients, scaled by the probability of sampling either u or i respectively, if $u/i \in S_t$, and 0 if $u/i \notin S_t$. The approximate gradient with respect to μ_k is estimate is then

$$\partial\mu^t_k = \frac{\mu_0 - \mu_k}{\sigma_0^2} + \frac{1}{ch(t)} \sum_{(u,i)\text{: either } u \in S_t \text{ or } i \in S_t} \Phi^{kk}_{ui} \left(s_{ui} - l_{ui} \exp\{\mu_k + \sigma^2_\psi/2\} \right) \tag{50}$$

Computing these updates takes $O(\mathbb{E}_{h(t)}|S_t|K(N+M))$ time.

The stochastic natural gradients for β and θ are obtained similarly scaling the non-stochastic gradients by the probability that $i/u \in S_t$ and only updating the terms β_{ik} and θ_{uk} such which u and i are S_t. The local terms $\Phi_{ui}, l_{ui}, t_{sui}$ and s_{ui} need only be computed locally; yielding $O(\overline{|S_t|}(N+M)(K+K'^2))$ efficiency. If we sample sets uniformly, then the scaling for the stochastic gradients of β, θ, Γ,and Λ can be absorbed into the learning rate, as in [4].

8.3 Sublinear SVI

In the previous section, we used set-based sampling to develop a linear-time inference algorithm. In this section, we describe a sublinear inference.

To build intuition, suppose we fix values of ι_{ui}. If we now look at the coordinate ascept updates for Γ, μ and Λ, we see can decompose sums into parts which depend on the links of a given user and item, and the parts which depend on the non-links. For example,

$$\partial_{\cdot|\iota}\Gamma_{uk} = -\Gamma_{uk} + \alpha_k + \sum_{i \in \text{links}(u)} \phi^{kk}_{ui} + \sum_{i \in \text{nonlinks}(u)} \phi^{kk}_{ui} \tag{51}$$

Many real-world graphs are sparse, so the summation over nonlinks is both computationally expensive and relatively uninformative [5]. By taking noisy estimates of nonlinks, both [4] and [5] compute updates sublinearly in the number of users and items[8].

From a theoretical perspective, there was nothing special about partitioning the sum over ϕ^{kk}_{ui} into links(i) and nonlinks(i), conditioned on ι_{ui}; any other partition works just as well, at least in theory. This gives use flexibility in depining a parition for ZIPF, where the links in ZIPF are random variables governed by ι_{ui}.

A first attempt might be to fix $\epsilon > 0$, and then handle the set $I_{\epsilon,u} = \{(u,i') : s_{ui'} < \epsilon \text{ for } u \text{ fixed}\}$ stochastically, and all $(u,i) \in I^c_\epsilon$ deterministically. Unfortunately, I_ϵ changes at each interation, and therefore needs to be updated at each step, which could take $O(NM)$ time.

However, if we fix $\epsilon = 1$, then $I_\epsilon = I_1$ is fixed at each iteration: it is the set of user-item pairs $(u,i) \notin \mathcal{R}$; that is, for which no review was observed. Then, for each user or item in S_t, we can specificy distributions $h'_{u,t}(t')$ and $h'_{i,t}(t')$ over sets $S^{t'}_u$ and $S^{t'}_i$ of nonlinks, respectively. If each item appears a the same number $c_u \geq 0$ of sets $S^{t'}_u$ - and similarly for items - then we can apply set based subsampling as in the previous subsection. For example, the stochastic gradient with respect to Γ_{uk} can now be written

$$\partial\Gamma_{uk} = \frac{1}{cPr(u \in S^t)}\{-\Gamma_{uk} + \alpha_k + \sum_{i \in \text{links}(u)} \phi^{kk}_{ui} + \sum_{i \in S^{t'}_u} \frac{1}{c_u Pr(i \in S^{t'}_u)} \phi^{kk}_{ui}\} \tag{52}$$

Assuming uniform subsampling, this becomes

$$\partial\Gamma_{uk} = \Gamma_{uk} + \alpha_k + \sum_{i \in \text{links}(u)} \phi^{kk}_{ui} + \sum_{i \in S^{t'}_u} \frac{M - |I_{1,u}|}{|S^{t'}_u|} \phi^{kk}_{ui} \tag{53}$$

where $I_{1,u} \triangleq \{(u',i') \in I_1 : u' = u\}$. We use the same $S^{t'}_u$ for each index $k \in 1 \ldots K'$, which allows us to compute only $|S^{t'}_u|$ values of Φ_{ui}. For ease of notation, define

$$w^a_{ui} \triangleq 1 + (\frac{M - |I_{1,u}|}{S^{t'}_u} - 1)\mathbb{1}_{i \in S^{t'}_u} \quad \text{and} \quad w^b_{ui} \triangleq 1 + (\frac{N - |I_{1,i}|}{S^{t'}_i} - 1)\mathbb{1}_{i \in S^{t'}{}_i}. \tag{54}$$

[8] In fact, [4] does not bother to even describe non-stochastic variational inference, due to its poor scalability

Keeping uniform sampling, the updates for Λ_u and μ are

$$\begin{aligned}
\partial\Lambda_u &= -\frac{\Lambda_u}{\sigma_{1,U}^2} + \sum_{i \in I_{1,u} \cup S_u^{t'}} w_{ui}^a (s_{ui} - l_{ui} t_{ui}) \\
\partial\mu_k &= \frac{\mu_0 - \mu_k}{\sigma_0^2} + \frac{N}{2|S_U|} \sum_{u \in S} \sum_{i \in S_u^{t'} \cup I_{1,u}} w_{ui}^a \Phi_{ui}^{kk} \left(s_{ui} - l_{ui} \exp\{\mu_k + \sigma_\psi^2/2\} \right) \\
&\quad + \frac{M}{2|S_I|} \sum_{i \in S} \sum_{u \in S_i^{t'} \cup I_{1,i}} w_{ui}^b \Phi_{ui}^{kk} \left(s_{ui} - l_{ui} \exp\{\mu_k + \sigma_\psi^2/2\} \right)
\end{aligned} \tag{55}$$

where S_U and S_I denote the sets users and items in S, respectively, $I_{1,i}$ and $S_i^{t'}$ are defined in the same way as $I_{1,u}$ and $S_u^{t'}$, and the factor of 2 in the noisy gradient of μ_k adresses the double counting in the formula. The gradients for item parameters Γ_{ik} and Λ_i are analogous.

With the same uniform sampling, the stochastic gradients for θ_{uk} and β_{ik} are

$$\begin{aligned}
\widehat{\nabla}_{\gamma_{uk}} \left(\mathcal{L}_1(q) | \iota \right) = &\quad \langle a + \sum_i r_{ui} \phi_{uik} - \gamma_{uk}^{\text{shp}}, b + \sum_{i \in I_{1,u}} \lambda_{ik}^{\text{shp}} / \lambda_{ik}^{\text{rt}} + \frac{M - |I_{1,u}|}{|S_u^{t'}|} \sum_{i \in S_u^{t'}} \lambda_{ik}^{\text{shp}} / \lambda_{ik}^{\text{rt}} s_{ui} - \gamma_{uk}^{\text{rte}} \rangle \\
\widehat{\nabla}_{\lambda_{ik}} \left(\mathcal{L}_1(q) | \iota \right) = &\quad \langle c + \sum_u r_{ui} \phi_{uik} - \lambda_{ik}^{\text{shp}}, d + \sum_{u \in I_{1,i}} \gamma_{uk}^{\text{shp}} / \gamma_{uk}^{\text{rt}} + \frac{N - |I_{1,i}|}{|S_i^{t'}|} \sum_{u \in S_i^{t'}} \gamma_{uk}^{\text{shp}} / \gamma_{uk}^{\text{rt}} s_{ui} - \lambda_{ik}^{\text{rte}} \rangle
\end{aligned} \tag{56}$$

The shape updates summation only over $(u, i) \in R$, which takes on average $O(|S|(\frac{R}{N} + \frac{R}{M})$ updates per mini-batch. Updates for the rate are in taken by sampling in the same spirit as updates for the community membership parameters.

The parameters ϕ_{uk}, Φ_{uk} and s_{ui} are optimized locally, as are the constants t_{ui} and l_{ui}.

At each iteration, we update a global learning rate ρ' for $\boldsymbol{\mu}$, and local learning rates ρ_u and ρ_i for each user and item. The global learning rate is a function of t, the total number of iterations, while the global learning trates are functions of t_u and t_i, the total number of iterations in which $u \in S$ and $i \in S$, respectiely. We set

$$\rho_{u/i}(t) = (\tau_0 + t_{u/i})^{-\kappa} \quad \text{ and } \rho'(t) = (\tau_0 + t)^{-\kappa} \tag{57}$$

Here $\tau_0 \geq 0$ downweights the early iterations, and $\kappa \in (.05, 1]$ ensures that $\sum_t \rho(t) = \infty$, while $\sum_t \rho(t) = \infty$ [3]. User and item parameters updates follow the noisy gradient, scaled by ρ_u and ρ_i:

$$\begin{aligned}
&\Gamma_{uk} \leftarrow \Gamma_{uk} + \rho_u(t) \partial\Gamma_{ik}^t \quad \Gamma_{ik} \leftarrow \Gamma_{ik} + \rho_i(t) \partial\Gamma_{ik}^t \qquad \Lambda_u \leftarrow \Lambda_u + \rho_u(t) \partial\Lambda_u^t \Lambda_i \leftarrow \Lambda_i + \rho_i(t) \partial\Lambda_i^t \\
&\gamma_{uk} \leftarrow \gamma_{uk} + \rho_u(t) \partial\gamma_{uk}^t \qquad \gamma_{ik} \leftarrow \gamma_{ik} + \rho_i(t) \partial\gamma_{ik}^t \quad \lambda_{uk} \leftarrow \lambda_{uk} + \rho_u(t) \partial\lambda_{uk}^t \lambda_{ik} \leftarrow \lambda_{ik} + \rho_i(t) \partial\lambda_{ik}^t
\end{aligned} \tag{58}$$

while community strenghts are updated as

$$\mu_k \leftarrow \mu_k + \rho'(t) \partial\mu_k^t \tag{59}$$

Pseudocode is provided in Algorithm 2.

8.4 Complexity

Let $U_{\max}$ $U_{\max}$ and $I_{\max}$ denote the most ratings given by one user and the most ratings of one item, respectively, and suppose we use a constant sample size $|S'|$ for $S_u^{t'}$ and $S_i^{t'}$. Then, SVI runs in $O(|S|(U_{\max} + I_{\max} + S')(K + K'^2))$ and requires $O((N + M)(K + K'))$ memory.

This bound is quite loose, since SVI requires $(K + K'^2)$ steps for each pair (u, i) in the set

$$P \triangleq \bigcup_{u \in S} (I_{1,u} \cup S_u^{t'}) \cup \bigcup_{i \in S} (I_{1,i} \cup S_i^{t'}) \tag{60}$$

Taking expectations over the unform sampling distribution $\mathbb{E}|I_{1,u}| = R/N$ while $\mathbb{E}|I_{1,i}| = R/M$. Thus, each update computes in a average of $O(|S|(|S'| + R(1/N + 1/M))(K + K'^2))$ steps. For sparse datasets, this constant $R(1/N + 1/M)$ is quite small and yields improved scalability.

Appendix B describes more sophisticated sampling techniques, and adresses their complexity.

```
Data: Nonnegative Integer Reviews $r_{ui}$
Initialize parameters 7.3;
while not yet converged do
    Sample a mini-batch $S$ of users and restaurants.
    For each user $u \in S$ and item $i \in S$, uniformly sample sets of items $S_u^{t'}$ and users $S_i^{t'}$ for which $r_{ui} = 0$.
    Denote by $P$ the set of all user-item pairs such that $u \in S$ and $i \in I_{1,u} \cup S_u^{t'}$, or $i \in S$ and $u \in I_{1,i} \cup S_i^{t'}$.
    local step;
    Optimize $\phi_{ui}$, $s_{ui}$, and $\Phi_{ui}$ for all $(u,i) \in P$ following equations 29, 43 and 46.
    global step;
    Update user and item latent features $\beta_u$ and $\beta_i$ for all $u, i \in S$ using stochastic natural gradient from Eqn. 56 in Eqn. 58
    Update community memberships $\Gamma_{u/i}$, using stochastic natural gradient from Eqn. 53 in Eqn. 58
    Update community popularities $\Lambda_{u/i}$, using stochastic gradient from Eqn. 55 in Eqn. 58
    Update community strengths $\mu$ using stochastic natural gradient from Eqn. 55 in Eqn. 59
    Update step sizes $\rho_0$ and $\rho'$ as in Eqn. 57.
end
```

Algorithm 2: Stochastic Variational Inference for AMP-ZIPF

9 Reverse ZIPF

Recall that ZIPF follows the convention that an unobserved review r_{ui} is set to zero. The basic idea behind ZIPF is that a review is zero either because it is unobserved due to some community membership model, or because the user has low enough preference for the item that they neglect it. This assumption makes intuitive sense for a recommendation system, and represents the situation well in the case the data counts the number of times a user has performed an activity indicating an item preferences (for example, clicks on an article or replays of a purchased song).

Unforunately, the model is clearly fictional in the case of user provided numerical ratings. Indeed, data scientists know precisely which ratings - if any - are zero, and which are simply unobserved.

A more realistic model for reviews might be $r_{ui} \sim \text{ZIP}(1 - \mathbb{1}_{r_{ui} \text{ is observed}}, \theta_u^T \beta_i)$. Unfortunately, this model would discard a lot of valuable data, as a users preferences have a large bearing on whether the review is observed. Therefore, it makes sense to augment the rating model with a binary response ι_{ui} variable indicating the presence of a review. Unlike ZIPF, where ι is hidden, we now treat ι_{ui} as known. We then develop a model where the users preference for an item influences the likelihood that $\iota_{ui} = 1$. Since the zero inflation is treated explicity rather than implciitly, we will call such a model "Reverse-ZIPF".

We now outline a completely conjugate Reverse ZIPF model. Again, let u index the N users, i index the N items, k index the K preference features and k' index the K' community affiliations. Let's introduce the generative process for a model which we call Gamma-Reverse ZIPF, or GR-ZIPF:

1. Rating Model
   (a) Draw latent user preferences $\theta_{ui} \sim \text{Gamma}(a, b)$
   (b) Draw latent item attributes $\beta_{ik} \sim \text{Gamma}(c, d)$.
   (c) Draw feature-to-rating contributions $x_{uik} \sim \text{ZIP}(\theta_u k \beta_{ik}, \mathbb{1}_{r_{ui} \text{ is observed}})$
   (d) Observe review $r_{ui} = \sum_k x_{uik}$.
2. Review Observation Model
   (a) Draw latent user community membership $\xi_{uk'} \sim \text{Gamma}(a', b')$
   (b) Draw latent item community membership $\zeta_{ik'} \sim \text{Gamma}(c', d')$
   (c) Draw latent item-preference scaling $\alpha_{2,i} \sim \text{Gamma}(e, f)$
   (d) Draw latent user-preference scaling $\alpha_{1,u} \sim \text{Gamma}(g, h)$
   (e) Draw preference to observation contributions $y_{uik}^a \sim \text{Poisson}(\alpha_{i,1} \alpha_{u,2} \theta_{uk} \beta_{ik})$
   (f) Draw community to observation contribution $y_{uik'}^b \sim \text{Poisson}(\xi_{uk'} \zeta_{ik'})$

(g) Observe rating indicator $\iota_{ui} = \sum_k y^a_{uik} + \sum_{k'} y^b_{uik'}$

Marginalizing out y^a and y^b, we see that $\iota_{ui} \sim \text{Poisson}(\alpha_i \theta_u^T \beta_i + \xi_u^T \zeta_i)$.

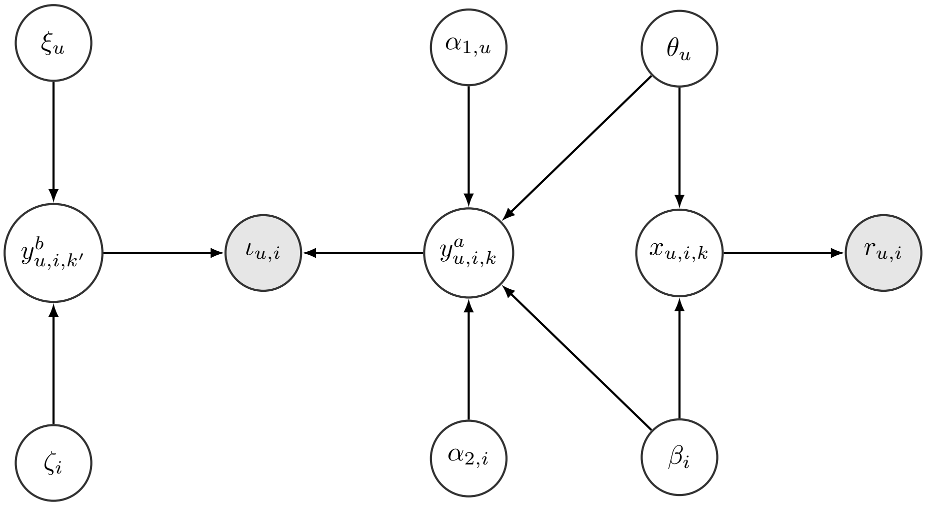

Figure 2: Graphical Representation of a GR-ZIPF

Informally, GR-ZIPF consists of two layers of BPF superimposed on one another: the first layer explains the numerical value of ratings, while the second layer explains the presence of the ratings.

9.1 Complete Conditionals and Variational Inference

Following the same steps as in the derivation of the complete conditions of the global variables for ZIPF, we compute

$$\begin{aligned}
\xi_{uk} | a', b', y^b, \zeta &\sim \text{Gamma}(a' + \sum_i y^b_{uik}, b' + \sum_i \zeta_{ik}) \\
\zeta_{ik} | c', d', y^b, \xi &\sim \text{Gamma}(c' + \sum_u y^b_{uik}, d' + \sum_u \xi_{uk}) \\
\theta_{uk} | a, b, y^a, x, \alpha_2, \alpha_1 &\sim \text{Gamma}(a + \sum_i y^a_{uik} + \sum_i x_{uik}, b + \alpha_{1,u} \sum_i (1 + \iota_{ui}\alpha_{2,i})\beta_{ik}) \\
\beta_{ik} | c, d, y^a, x^a, \alpha_{2,i}, \alpha_{1,u} &\sim \text{Gamma}(c + \sum_u y^a_{uik} + \sum_u x_{uik}, d + \alpha_{2,i} \sum_u (1 + \iota_{u,i}\alpha_{1,u})\theta_{uk}) \\
\alpha_{1,u} | e, f, y^a, \theta, \theta_{1,u}, \beta &\sim \text{Gammma}(e + \sum_{i,k} y^a_{uik}, f + \sum_k \theta_{uk} \sum_i \alpha_{2,i}\beta_{ik}) \\
\alpha_{2,i} | g, h, y^a, \beta, \alpha_{1,u}, \theta &\sim \text{Gamma}(g + \sum_{u,k} y^a_{uik}, h + \sum_k \beta_{ik} \sum_u \alpha_{1,u}\theta_{uk})
\end{aligned} \tag{61}$$

Just as in the BPF model, the complete conditionals for the local parameters parameters x_{ui} are just $x_{ui} | \theta, \beta \sim \text{Mult}(r_{ui}, \frac{\theta_u * \beta_i}{\theta_u^t \beta_i})$ The complete conditional for the vector $y_{ui} \triangleq (y^a_{ui1}, y^a_{ui2}, \ldots, y^a_{uiK}, y^b_{ui1}, y^b_{ui2}, \ldots, y^b_{u1K'})$ is then $y_{ui} \sim \text{Multi}(\iota_{ui}, v_{ui})$ where v_{ui} is the vector

$$v_{ui,j} = \begin{cases} \xi_{uk}\zeta_{ik} & 1 \leq j \leq K \\ \alpha_{1,u}\alpha_{2,i}\theta_{u,j-K}\beta_{i,j-K} & K+1 \leq j \leq K + K' \end{cases} \tag{62}$$

Note that ι is a binary response variable, so in fact we may write that $y_{ui} \sim \iota_{ui}\text{Cat}(v_{uik})$.

We see that the model is completely conjugate, and so we approximate the true distribution p with the mean field family q where the distributions of the global parameters are given by

$$\begin{aligned} q(\xi_{uk'}) &\sim \mathrm{Gamma}(\mu^{\mathrm{shp}}_{uk'}, \mu^{\mathrm{rte}}_{uk'}) \quad q(\zeta_{ik'}) \sim \mathrm{Gamma}(\nu^{\mathrm{shp}}_{ik'}, \nu^{\mathrm{rte}}_{ik'}) \\ q(\theta_{uk}) &\sim \mathrm{Gamma}(\gamma^{\mathrm{shp}}_{uk}, \gamma^{\mathrm{rte}}_{uk}) \quad q(\theta_{ik}) \sim \mathrm{Gamma}(\lambda^{\mathrm{shp}}_{ik}, \lambda^{\mathrm{rte}}_{ik}) \\ q(\alpha_{2,i}) &\sim \mathrm{Gamma}(\eta^{\mathrm{shp}}_{2,i}, \eta^{\mathrm{rte}}_{2,i}) \quad q(\alpha_{1,u}) \sim \mathrm{Gamma}(\eta^{\mathrm{shp}}_{1,u}, \eta^{\mathrm{rte}}_{1,u}) \end{aligned} \tag{63}$$

and the local parameters are governed by

$$q(x_{ui}) = \mathrm{Mult}(r_{ui}, \phi_{ui}) \qquad q(y_{ui}) = \delta_{\iota_{ui}=1}\mathrm{Cat}(\psi_{ui}) \tag{64}$$

Here ϕ_{ui} is a probability distribution on the $K-1$ simplex. If $\iota_{ui} = 1$, then the vector ψ_{ui} is a probability distribution supported on the $K + K' - 1$ simplex. If $\iota_{ui} = 0$, we adopt the convention that ψ_{ui} is just the zero vector.

Coordinate ascent updates are straightforward to compute and are analogous to the BPF and ZIPF updates. For example, the updates for the user parameters are

$$\begin{aligned} \mu_{uk'} &= \langle a' + \sum_i \psi_{ui,K+k'}, b' + \sum_i \nu^{\mathrm{shp}}_{ik}/\nu^{\mathrm{rte}}_{ik} \rangle \\ \gamma_{uk} &= \langle a + \sum_i \psi_{ui,k} + x_{uik}, b + \eta^{\mathrm{shp}}_{1,u}/\eta^{\mathrm{rte}}_{1,u} \sum_i (1 + \iota_{ui}\eta^{\mathrm{shp}}_{2,i}/\eta^{\mathrm{rte}}_{2,i})\lambda^{\mathrm{shp}}_{ik}/\lambda^{\mathrm{rte}}_{ik} \rangle \\ \eta_{1,u} &= \langle e + \sum_i \sum_{k'} \psi_{ui,K+k'} + x_{uik}, f + \sum_k \gamma^{\mathrm{shp}}_{uk}/\gamma^{\mathrm{rte}}_{uk} \sum_i \eta^{\mathrm{shp}}_{2,i}/\eta^{\mathrm{rte}}_{2,i} \cdot \lambda^{\mathrm{shp}}_{ik}/\lambda^{\mathrm{rte}}_{ik} \rangle \end{aligned} \tag{65}$$

Updates for the local parameters ϕ_{ui} are the same as in BPF and ZIPF, and updates for ψ_{ui} are similar:

$$\psi_{uij} \quad \propto \quad \begin{cases} \exp\{\Psi(\eta^{\mathrm{shp}}_{1,u}) - \log \eta^{\mathrm{rte}}_{1,u} + \Psi(\eta^{\mathrm{shp}}_{i}) - \log \eta^{\mathrm{rte}}_{i} + \Psi(\gamma^{\mathrm{shp}}_{uk}) - \log \gamma^{\mathrm{rte}}_{uk} \\ +\Psi(\lambda^{\mathrm{shp}}_{ik}) - \log \lambda^{\mathrm{rte}}_{ik}\} & 1 \le j \le K \\ \\ \exp\{\Psi(\mu^{\mathrm{shp}}_{u,j-K}) - \log \mu^{\mathrm{rte}}_{u,j-K} + \Psi(\nu^{\mathrm{shp}}_{i,j-K}) - \log \nu^{\mathrm{rte}}_{u,j-K}\} & K+1 \le j \le K+K' \end{cases} \tag{66}$$

where the ψ_{ui} are normalized so that $\sum_{1 \le j \le K+K'} \psi_{uij} = 1$. The variational inference algorithm is summarized in Algorithm 3, and Appendix C shows that these updates can be completed in $O((N + M)(K + K'))$ time.

```
Data: Nonnegative Integer Reviews $r_{ui}$
Initialize parameters ;
while not yet converged do
    local step;
    Optimize $\phi_{ui}$ and $\psi_{ui}$ for all $(u, i)$ for which $r_{ui} > 0$ via equations 29 and 66.
    global step;
    For all users $u$, update $\theta_u$, $\eta_{1,u}$, and $\mu_u$ by 67.
    For all $u \in S$, update $\beta_i$, $\eta_{2,i}$, and $\nu_i$ analogous to 67.
end
```

Algorithm 3: Stochastic Variation Inference for GR-ZIPF

9.2 Stochastic Variational Inference

Though GR-ZIPF scales linearly in users and items (Eqn. C), we can scale sublinearly with stochastic variational inference. The samping strategy is essentially the same as for ZIPF. We begin by sampling a minibatch S of items i and users u. For each item u, we define the set $I_{1,u}$ as in 8.3, that is, as the set of items i for which have been reviewed by user u. For each $i \in I_{1,u}$, we update the corresponding multinomial parameters ϕ_{ui} and ψ_{ui}. We then uniformly sample sets $S_u^{t'}$ of items for which not been reviewed by user u. Leting w_{ui}^a be as in 54, the updates are

$$\begin{aligned}
\partial\mu_{uk'} &= \langle a' + \sum_i \psi_{ui,K+k'}, b' + \sum_{i \in I_{1,u} \cup S_u^{t'}} w_{ui}^a \nu_{ik}^{\text{shp}} / \nu_{ik}^{\text{rte}} \rangle - \mu_{uk'} \\
\partial\gamma_{uk} &= \langle a + \sum_i \psi_{ui,k} + x_{uik}, b + \eta_{1,u}^{\text{shp}} / \eta_{1,u}^{\text{rte}} \sum_{i \in S_u^{t'} \cup I_{1,u}} w_{ui}^a (1 + \iota_{ui} \eta_{2,i}^{\text{shp}} / \eta_{2,i}^{\text{rte}}) \lambda_{ik}^{\text{shp}} / \lambda_{ik}^{\text{rte}} \rangle - \gamma_{uk} \qquad (67) \\
\partial\eta_{1,u} &= \langle e + \sum_i \sum_{k'} \psi_{ui,K+k'} + x_{uik}, f + \sum_k \gamma_{uk}^{\text{shp}} / \gamma_{uk}^{\text{rte}} \sum_i w_{ui}^a \eta_{2,i}^{\text{shp}} / \eta_{2,i}^{\text{rte}} \cdot \lambda_{ik}^{\text{shp}} / \lambda_{ik}^{\text{rte}} \rangle - \eta_{1,u}
\end{aligned}$$

and updates for global item parameters are analogous. Updates are similar to in 58, where each parameter is incremented by its natural gradient, scaled by the step size. Letting t_u and t_i be the number of times an item/user has been sampled, we use step size $\rho(t_u)$ and $\rho(t_i)$, where again ρ is the function $t \mapsto \frac{1}{(\tau_0+t)^\kappa}$ Algorithm 4 provides pseudocode:

```
Data: Nonnegative Integer Reviews $r_{ui}$
Initialize parameters ;
while not yet converged do
    Sample a mini-batch $S$ of users and restaurants.
    For each user $u \in S$ and item $i \in S$, retrieve the sets $I_{1,u}$ and $I_{1,i}$
    For each user $u \in S$, uniformly subample $S_u^{t'}$ and $S_i^{t'}$. Let $P$ denote the set of all node pairs $(u, i)$for which either $u \in S$ and $i \in S_u^{t'} \cup I_{1,u}$, or $i \in S$ and $i \in S_i^{t'} \cup I_{1,i}$. Define $P$ as in B
    local step;
    Optimize $\phi_{ui}$ and $\psi_{ui}$ for all $(u, i) \in P$ following equations 29 and 66.
    global step;
    For all $u \in S$, update $\theta_u$, $\eta_{1,u}$, and $\mu_u$ with step size $\rho(t_u)$ and gradient Eqn. 67
    For all $u \in S$, update $\beta_i$, $\eta_{2,i}$, and $\nu_u$ with step size $\rho(t_i)$ and gradient analogous to Eqn. 67
end
```

Algorithm 4: Stochastic Variation Inference for GR-ZIPF

9.3 Comparison with AMP-ZIPF

GR-ZIPF has two advtanges over AMP-ZIPF. The first is that, unlike AMP-ZIPF, GR-ZIPF actually distinguishes between zero and non-observed reviews. In doing so, GR-ZIPF gives a subtler and more flexible account of user behavior. For example, by placing strong priors on the scalings $\alpha_{1,u}$ and $\alpha_{2,i}$, we can control the extent to which our model correlates which objects a user decides to rate with how strongly he or she rates them. Thus, we can derive meaningful information from each users choices over which objects to consume, whilst retaining some invariance to unwanted sparse community structure. The second advtange is in efficiency: by cleverly precomputing terms, GR-ZIPF can run in $O(R\,[K + K'])$ time and $O((M + N)[K + K'])$ space (see Theorem 1 in the Appendix). In sparse review sets, $R << NM$, so that Reverse ZIPF has substantial advtanges over non-stochastic AMP-ZIPF.

On the other hand, GR-ZIPF models the binary response variable ι_{ui} as the outcome of a Poisson distribution, which takes values on the positive integers. While [2] and [28] have used Poisson distribution to model binary data, neither source provides theoretical justification for this decision. More broadly, the ZIPF framework is more flexible than Reverse ZIPF models, as the former is compatible with existing community membership models in the literature. It would be interesting to try to describe a class of models for the response variable ι_{iu} in Reverse ZIPF for which mean field variational inference is tractable. Perhaps this class would include models where the generative process specifices the the ι as Bernoulli, rather than Poisson.

10 Conclusion

This paper has made the case that existing Matrix Factorization algorithms are ill suited to data sets with sparse community structure. We have introduced two models, AMP-ZIPF and GR-ZIPF, based on the

Zero Inflated Poisson distribution which de-emphasize observed data sparsity in order to learn underlying preferences. Preliminary numerical results in Section 7.4 suggest that AMP-ZIPF can better account for real-world data than BPF, and is far more robust to sparse community structure. In future work, we hope to implement stochastic AMP-ZIPF, and both stochastic and non-stochastic GR-ZIPF, and test the two methods with more rigorous recommendation system metrics.

One possible drawback of ZIPF is that it may disregard evidence from unobserved reviews too recklessly. It is conceivable that ZIPF models might learn mean item ratings, and hence fail to personalize results. A consumer's choice to consume one item over another still provides considerable evidence in favor of preference. While GR-ZIPF may mitigate this problem (Section 9.3), we may still have to develop zero-inflated models which distinguish between sparse communities induced by contingent barriers - such as geographic location - and actual user preference, such as price range and taste in cuisine. Future models which combine item descriptions, review text, price, location and tag data will help recommendation systems both learn and *explain* sparse community structure. With these advances, a user could alter their budget setttings for a special occasion, enter a new cuisine for a change of pace, or input a destination when they travel, and the recommendation system would adapt accordingly.

Zero-Inflated models that incorporate text have yet another advantage: In data sets which are very sparse, there may not be enough information in the numerical ratings to learn anything *but* the sparse community structure. Review text could be the rich source of common features for users in different geographic locations. Indeed, while we have successful shown that ZIPF does not learn geographic features, it is currently unclear if more meaningful features are extracted in their place.

Finally, it would be very interesting to explore the connections between zero-inflated models and the exploration/exploitation payoffs in contextual bandit learning [47]. Ultimately, with further research, we hope that zero-inflated models may give rise to a new generation of recommendation systems which are able to diversify, broaden, and even educate users' preferences.

11 Acknowledgements

I would like to thank my advisor, Professor David Blei, for all his help and guidance. I would also like to thank Prem Gopalan for his GAPREC, SVI, and SVIAMP code repositories, and for letting me take up his time with questions. I also thank David Dohan '15 for helping me to setup various coding utilities, and to parse JSON files.

12 Honor Code

This Junior Paper represents my work in Accordance with the University Regulations.

References

[1] *Yelp Academic Data Set.* https://yelp.com/academic_dataset, 2013.

[2] Gopalan, Prem, Jake M. Hofman, and David M. Blei. Scalable Recommendation with Poisson Factorization. *arXiv preprint* arXiv:1311.1704 (2013).

[3] M. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

[4] P.K. Gopalan, C.Wang, and D.M. Blei. Modeling Overlapping Communities with Node Popularities. In *Neural Infromation Processing Systems,* 2013.

[5] P. K. Gopalan and D. M. Blei. Efcient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):1453414539, 2013.

[6] Koren, Yehuda, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer* 42(8),30-37, 2009.

[7] L. Lu, M. Medo, C.H. Yeung, Y. Zhang, Z. Zhang, T. Zhou. Recommender Systems. *Physics Reports* 519(1), 1-49, 2012.

[8] S. Arora, R. Ge, and R. Kannan. Computing a Nonnegative Matrix Factorization - Provably. In *Proceedings of the 44th Symposium on Theory of Computing*, 145-162, 2012.

[9] N. Xia and G. Karypis. Sparse Linear Methods with Side Information for Top-n Recommendations. *Sparse Linear Methods with Side Information for Top-n Recommendations*, 155-162, 2012.

[10] C. Wang and D. Blei. Collaborative topic modeling for recommending scientific articles. *Knowledge Discovery and Data Mining*, 2011.

[11] J. Chang and D. Blei. Relational Topic Models for Document Networks . *Artificial Intelligence and Statistics*, 2009.

[12] S. Wild, J. Curry, and A. Dougherty. Motivating nonnegative matrix factorizations.In *Proc. SIAM Applied Linear Algebra Conf*, 2003.

[13] D. Cai , X. He, X. Wu, and J. Han. Non-negative matrix factorization on manifold. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, 63-72. IEEE, 2008.

[14] Y. He, H. Lu , and S. Xie. Semi-supervised non-negative matrix factorization for image clustering with graph Laplacian. *Multimedia Tools and Applications* 1-23, 2013.

[15] C. Wang and D. Blei. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. *Neural Information Processing Systems*, 2009.

[16] X. Ning and G. Karypis Slim: Sparse linear methods for top-n recommender systems. *In 2011 IEEE 11th International Conference on Data Mining (ICDM)*, 497-506, IEEE 2011.

[17] Z. Yang, T. Hao, O. Dikmen, X. Chen, and E. Oja. Clustering by nonnegative matrix factorization using graph random walk. In *Advances in Neural Information Processing Systems*, 1088-1096, 2012.

[18] D. Seung and L. Lee. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* 13,556-562, 2001.

[19] A. Honkela, M. Tornio, T. Raiko, and J. Karhunen. Natural conjugate gradient in variational inference. In *Neural Information Processing*, pages 305-314. Springer, 2008.

[20] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788-791, October 1999.

[21] Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In Neural *Information Processing Systems*, pages 507-513, 2001.

[22] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul.Introduction to variational methods for graphical models. *Machine Learning*, 37:183-233, 1999.

[23] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 20:1257-1264, 2008.

[24] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880-887. ACM, 2008.

[25] B. Rechet, C. Re, J. Tropp, V. Bittorf. Factoring nonnegative matrices with linear programs. *Advances in Neural Information Processing Systems 25*, 1223-1231, 2012.

[26] W. Xu, X. Liu, Y. Gong. Document Clustering Based on Non-negative Matrix Factorization. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 267-273, 2003.

[27] G. Shani and A. Gunawardana. Evaluating Recommendation Systems. In Recommender Systems Handbook. F Ricci, L. Rokach, B. Sharpira, P. Bantor edgs. 2011.

[28] B. Ball, B. Karrer, and M.E.J. Newman. Efficient and principled method for detecting communities in networks. *Phys. Rev. E*, 84(3), 036103:1-13, Sep 2011.

[29] M.J. Wainwright and M.I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2), 1-305, 2008.

[30] R. Vatsa and S. Wilson. The Variational Bayes Methods for Inverse Regression Problems with an Application To The Paleoclimate Reconstruction. *Journal of Combinatorics, Information and System Sciences*, 35(1-2),221-248, 2010.

[31] M. Chigona and C. Gaetan. Semiparametric zero-inflated Poisson models with applications to animal abundance studies. *Environmetrics*, 18(3), 303-314, May 2007.

[32] B.H. Neelon, A.J O'Malley, and S.T. Normand.A Beysian model for repeated measures zero-inflated count data with application to outpatient psychiatric service use. *Sat Modelling*, 10(4), 421-439, December 2010.

[33] S.K. Ghosh, P Mukhopadhyay, J.C. Lu. Bayesian analysis of zero-inflated regression models. *Journal of Statistical Planning and Inference*, 136(4), 1360-1375, 2006.

[34] D. Lambert. Zero-Inflated Poisson Regression, With an Application to Defects in Manufacturing. *Technometrics* 34(1):1-14, 1992.

[35] S. Amari. Differential geometry of curved exponential Families-Curvatures and information loss. *The Annals of Statistics*, 10(2):357385, June 1982.

[36] J. Yin, Q. Ho and E. P. Xing, A Scalable Approach to Probabilistic Latent Space Inference of Large-Scale Networks. *Advances in Neural Information Processing Systems 27*, 2013.

[37] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. *Eighth IEEE International Conference on Data Mining*, 2008.

[38] G. Goodhardt, S. Ehrenberg, and C. Chat
eld. The Dirichlet: A comprehensive model of buying behavior. *Journal of the Royal Statistical Society, Series A*, 147(5):621-655, 1984.

[39] E. Cinlar. Probability and stochastics. *Springer, Vol. 261.* 2011.

[40] K.P. Murphy. Machine learning: a probabilistic perspective. *The MIT Press*, 2012.

[41] L. Wasserman. All of statistics: a concise course in statistical inference. *Springer*, 2004.

[42] Q. Ho, J. Yin and E. P. Xing, On Triangular versus Edge Representations - Towards Scalable Modeling of Networks. *Advances in Neural Information Processing Systems 26*, 2012.

[43] L. Mackey, A. Talwalkar, M.I. Jordan. Divide-and-Conquer Matrix Factorization. *arXiv:1107.0789v6*, 2012.

[44] J. Rennie and N. Nathan Srebro. Fast Maximum Margin Matrix Factorization for Collaborative Prediction. *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[45] W. Xu, X. Liu and Y. Gong. Document clustering based on non-negative matrix factorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 267-273, 2003.

[46] E.J. Cands and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics 9.6*, 717-772, 2009.

[47] L. Li, W. Chu, J. Langford, and R. E. Schapire.. A contextual-bandit approach to personalized news article recommendation. [**?**]. ACM, 661-670, 2010.

A Complete Conditionals for ZIPF

Theorem 1. *In a generic ZIPF model, the complete conditionals $\theta_{uk}|\boldsymbol{\beta},\boldsymbol{\iota},\boldsymbol{x}$ and $\beta_{ik}|\boldsymbol{\beta},\boldsymbol{\iota},\boldsymbol{x}$ are*

$$\theta_{uk}|\boldsymbol{\beta},\boldsymbol{\iota},\mathbf{x} \sim \textit{Gamma}(a+\sum_i \iota_{ui}x_{iuk}, b+\sum_i \iota_{ui}\beta_{ik}) \quad \textit{and} \quad \beta_{ik}|\boldsymbol{\theta},\boldsymbol{\iota},\mathbf{x} \sim \textit{Gamma}(a+\sum_u z_{iu}x_{iuk}, b+\sum_u \iota_{ui}\theta_{uk}) \tag{68}$$

Proof. From the graph structure, we see that θ_{uk} depends only on the β_{ik}, the $x_{i,u,k}$, and $\iota_{i,u}$. Neglecting constants independent of $\theta_{u,k}$, the complete conditional is

$$P(\theta_{uk}|\beta,\mathbf{x},\iota) \propto P(\mathbf{x}|\iota,\theta_{u,k},\beta) \propto P(\mathbf{x}|\iota,\theta_{uk},\beta)Pr(\theta_{uk}) \tag{69}$$

Recall that we are conditioning on ι, and that $x_{i,u,k}$ is deterministically zero when $\iota_{ui}=0$. Thus, the above can be written as

$$p(\theta_{uk})\prod_{i:\iota_{ui}\neq 0} p(x_{iuk}|\theta_{u,k},\beta) \tag{70}$$

We know that when $\iota_{ui}\neq 0$, we have that $x_{i,u,k}\sim$ Poisson$(\theta_{u,k}\beta_{i,k})$. Writing the Poisson distribution its its cannonical exponential family form gives

$$\begin{aligned}
&p(\theta_{uk})\prod_{i:\iota_{ui}\neq 0}\exp(-\theta_{u,k}\beta_{i,k}+\log(\theta_{uk}\beta_{i,k})x_{iuk})\frac{1}{x_{iuk}!}\\
&= p(\theta_{uk})\prod_{i:\iota_{u}\neq 0}\exp(-\theta_{uk}\beta_{i,k}+\log(\theta_{uk})x_{iuk}+\log(\beta_{i,k})x_{iuk})\frac{1}{x_{iuk}!}\\
&\propto p(\theta_{uk})\prod_{i:\iota_{ui}\neq 0}\exp(-\theta_{uk}\beta_{ik}+\log\theta_{uk}x_{iuk})\\
&= p(\theta_{uk})\exp(-\theta_{uk}\sum_i \iota_{ui}\beta_{ik}+\log\theta_{u,k}\sum_i \iota_{ui}x_{iuk})\\
&= \frac{b^a}{\Gamma(a)}\exp(-b\theta_{uk}+(a-1)\log\theta_{u,k})\exp(-\theta_{uk}\sum_i \iota_{ui}\beta_{ik}+\log\theta_{uk}\sum_i \iota_{ui}x_{iuk})\\
&\propto \exp\left(-\theta_{uk}(b+\sum_i \iota_{ui}\beta_{ik})+\log\theta_{uk}(a-1+\sum_i \iota_{ui}x_{iuk})\right)
\end{aligned} \tag{71}$$

From which we see that

$$\theta_{uk}|\beta,\iota,\mathbf{x} \sim \text{Gamma}(a+\sum_i \iota_{ui}x_{iuk}, b+\sum_i \iota_{ui}\beta_{ik}) \tag{72}$$

A similar calculation gives

$$\beta_{ik}|\theta,\iota,\mathbf{x} \sim \text{Gamma}(a+\sum_u z_{iu}x_{iuk}, b+\sum_u \iota_{ui}\theta_{uk}) \tag{73}$$

□

B A More Sophisticated Sampling Algorith for ZIPF

In 8.3, we defined for each user and item the sets $I_{1,u}\triangleq\{i : r_{ui}>0\}$ and $I_{1,u}\triangleq\{u : r_{ui}>0\}$. For stochastic variational inference, we treated sums over user-item pairs not in these sets $I_{1,u}$ and $I_{1,u}$ stochastic, to improve computational efficiency. This decision is guided by the observation in [5] that non-links are generally less informative than links. In ZIPF, however, the is still a nonzero probability of link between a user u and item i for which $r_{ui}=0$. It was this observation which motivated us to look at the sets $I_{\epsilon,u}\triangleq\{i : s_{ui}>\epsilon\}$ and $I_{\epsilon,u}\triangleq\{u : s_{ui}>\epsilon\}$. Unfortunately, these sets change at each iteration, and are costly to keep updated.

Here, we sketch a compromise algorithm. Of the user-item pairs for which $r_{ui} = 0$, we mainting a set I' of pairs for which s_{ui} is above a certain threshold. For efficiency, we keep $|I'|$ capped by a constant $C \propto R$. During the local step at each iteration, when insert computed values of s_{ui} to I' if either $|I'| < C$ or if $s_{ui} > \arg\min_{u',i':(u',i')\in I'} s_{u'i'}$.In the latter case, we delete the pair (u', i') from I' corresponding to the minimum value of $s_{u',i'}$. Implementing $|I'|$ as a heap, this query takes constant time, and inserts are on the order of $\log R$.

Let S be our minibatch, S' are the sampled users and items for which $r_{ui} = 0$ (see 8.3 for notation), and set $I'_{u,1} \triangleq \{(u, i') \in I'\}$ for fixed u. Define $I'_{i,1}$ analogously, and define P to be the set of all user item pairs for which $u \in S$ and $i \in S_u^{t'} \cup I_{1,u} \cup I'_{u,1}$, or $i \in S$ and $u \in S_i^{t'} \cup I_{1,i} \cup I'_{1,i}$. Compute local coordinate updates for pairs in P, and, when computing global updates, treat sums over pairs $(u, i) : \{i \in I_{1,u} \cup I'_{1,u}, u \in S$ or $u \in I_{1,i} \cup I'_{1,i}$ nonstochastically.

Then, in the worst case, updating I' takes $O(|P| \log R)$, and each pass takes $O(|P|(K + K'^2))$. From 8.4, we recall that $\mathbb{E}|I_{1,u}| = \frac{R}{N}$ and $\mathbb{E}|I_{1,i}| = \frac{R}{M}$, where expectations are taken over the uniform sampling distribution. Thus,

$$\mathbb{E}|P| = O(|S| \left(|S'| + \frac{R}{N} + \frac{N}{M}) + \sum_{u\in S} |\{(u, i') \in I'\}| + \sum_{i\in S} |\{(i, u) \in I'\}| \right) \tag{74}$$

Again, since we keep $|I'| \propto R$, we get

$$\mathbb{E}|P| = O\left(|S| \left(|S'| + \frac{R}{N} + \frac{N}{M} \right) \right) \tag{75}$$

so that the expected complexity over each update is on the order of $|S| \left(|S'| + \frac{R}{N} + \frac{N}{M})(\log R + K + K'^2) \right)$. For realistic values of R, $\log R << K$, so we recover the same complexity as the algorithm given in Section 8.3.

```
Data: Nonnegative Integer Reviews $r_{ui}$
Initialize parameters (see Sec. 7.3);
while not yet converged do
    Sample a mini-batch $S$ of users and restaurants.
    For each user $u \in S$ and item $i \in S$, retrieve the sets $I_{1,u}$, $I_{1,i}$, $I'_{1,u}$, $I'_{1,i}$
    For each user $u \in S$, uniformly subample sets of items $S_u^{t'}$ for which $i \in\notin I_{1,u} \cup I'_{1,i}$. Do the sample for
    items $i \in S$.
    Define $P$ as in B
    local step;
    Optimize $\phi_{ui}$, $s_{ui}$, and $\Phi_{ui}$ for all $(u, i) \in P$ following equations 29, 43 and 46.
    global step, treating user-item pairs in $P$ non-stochastically;
    Update user and item latent features $\beta_u$ and $\beta_i$ for all $u, i \in S$ using stochastic natural gradient from
    Eqn. 56 in Eqn. 58
    Update community memberships $\Gamma_{u/i}$, using stochastic natural gradient from Eqn. 53 in Eqn. 58
    Update community popularities $\Lambda_{u/i}$, using stochastic gradient from Eqn. 55 in Eqn. 58
    Update community strengths $\mu$ using stochastic natural gradient from Eqn. 55 in Eqn. 59
    Update step sizes $\rho_0$ and $\rho'$ as in Eqn. 57
    Update $I'$
    For $u, i$ in $P$, add $(u, i)$ to $I'$ if $|I'|$ is less than a fixed capacity, or if if $s_{ui} > \arg\min_{u',i':(u',i')\in I'} s_{u'i'}$.In
    the latter case, we delete the pair $(u', i')$ from $I'$ corresponding to the mimimum value of $s_{u',i'}$.
end
```

Algorithm 5: Modified Stochastic Variational Inference for AMP-ZIPF

In less sparse review data-sets (click-data, for example), we could also imagine taking subsamples from $I_{1,u}$ and $I'_{i,u}$ as well. The implementation and pseudo code is a straightforward adaptation of the preceding discussion.

C Efficiency of GR-ZIPF

Theorem 1. *Let M be the number of items, N the number of users, R the number of reviews, K the number of latent preference features and K' the number of latent community membership features. Then the coordinate ascent updates for GR-ZIPF compute in $O\left(R\left[K+K'\right]\right)$, and require $O([N+M]\left[K+K'\right])$ space*

Proof. First, we show that the updates for global user parameters μ, γ, and η can be computed in $O((K+K')N)$ space and $O(R(K+K'))$ time. The updates for the shape parameter $\mu_{uk'}$, γ_{uk} and $\eta_{1,u}$ require the computation of

$$\sum_i \nu^{\text{shp}}_{ik'}/\nu^{\text{rte}}_{ik'} \qquad \sum_i \lambda^{\text{shp}}_{ik}/\lambda^{\text{rte}}_{ik} \qquad \sum_i \psi_{ui,K+k'} + x_{uik} \tag{76}$$

$$\sum_k \gamma^{\text{shp}}_{uk}/\gamma^{\text{rte}}_{uk} \sum_i \eta^{\text{shp}}_{2,i}/\eta^{\text{rte}}_{2,i} \cdot \lambda^{\text{shp}}_{ik}/\lambda^{\text{rte}}_{ik} \tag{77}$$

and $\sum_i \iota_{ui}\eta^{\text{shp}}_{2,i}/\eta^{\text{rte}}_{2,i} \cdot \lambda^{\text{shp}}_{ik}/\lambda^{\text{rte}}_{ik} = \sum_{i:r_{ui}>0} \eta^{\text{shp}}_{2,i}/\eta^{\text{rte}}_{2,i} \cdot \lambda^{\text{shp}}_{ik}/\lambda^{\text{rte}}_{ik}$. The first three need to be computed only once for each latent preference/community feature or k or k', so the corresponding updates reqire $O(M(K+K'))$ time and $O(K+K')$ memory. The inner sum in the fourth terms needs to be only computed a total of once per feature, taking $O(K)$ memory and $O(MK)$ time. Computing the outer sum requires $O(NK)$ time. The last term computes in $O(R)$ time for each K feature. Thus, shape parameters compute in $O(R(K+K'))$ time and $O(K+K')$ space. Shape parameters can be computed as

$$\sum_{i:r_{ui>0}} \psi_{ui,K+k'} \qquad \sum_{i:r_{ui}>0} \psi_{ui,k} \qquad \sum_{i:r_{ui}>0} x_{uik} \tag{78}$$

which take $O(R(K+K'))$ time and $O(K+K')$ space. An analogous argument show sthat global item parameters have the same time and memory complexity. It is clear from the given local parameter ascent that updates can be found in $O(R(K+K'))$ time. THe local parameter updates only need to be stored locally, and added to the appropriate shape parameter update. This takes $O(R(K+K'))$ time and constant space. Storing takes all the variational parameters then takes $O((M+N)(K+K'))$ space. □

D Holding Out Reviews by Location

In 3.1 , we held out reviews both at random, and then disproportionately based on location. We describe these two methods here in more detail. First, for each user, we counted the total number of restaurants which had been reviewed. For users which had reviewed fewer than 3, no restaurants are held out. Restaurants with between 3 and 8 reviews, and resaurants with 9 or greater reviews, are treated on separate sliding scales.

Let u denote a user, and let R_u denote the set of restaurants reviewed by u, and for each restaurant $i \in R_u$, let i^X denote its longitude and i^Y its lattidude. In the first round of tests, the restaurants to be withheld are chosen unifornly. In the second round of tests we compute the mean longitude and lattidue of the restaurants which user u has reviewed: $\mu^X_u = \sum_{i\in R_u} i^X$ and $\mu^Y_u = \sum_{i\in R_u} i^X$. Then, we compute the geographic deviation d_{ui} of each $i \in R_u$ from (μ^X_u, μ^Y_u) as $d_{ui} = (i^X - \mu^X_u)^2 + (i^Y - \mu^Y_u)^2$. Summing up all the deviations gives us the location variance σ^2_u. We let construct the distribution $\mathcal{D}^u$ over the restaurants in R_u:

$$\mathcal{D}^u_i = \left(\frac{d_{ui}}{\sigma^2_u} + \alpha\right) / \left(1 + \alpha|R_u|\right) \tag{79}$$

where $\alpha = .02$ is a regularization that smooths out the distribution (note that $\sum_i \mathcal{D}^u_i = 1$). We then draw restaurants to withhold from $\mathcal{D}^u$, without replacement.

E BPF Graphs

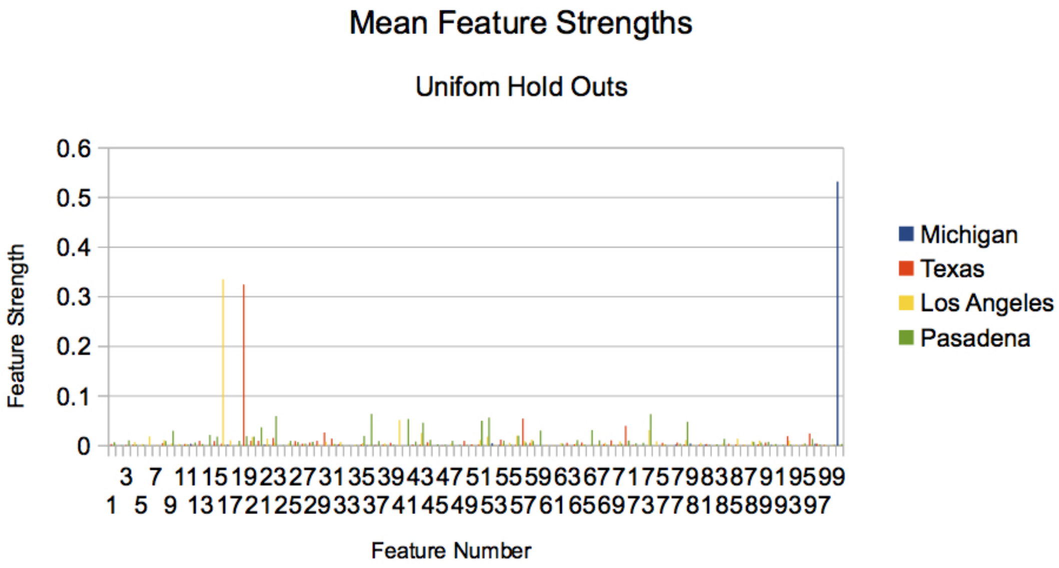

Figure 3: Mean Feature Strength for First BPF Trial

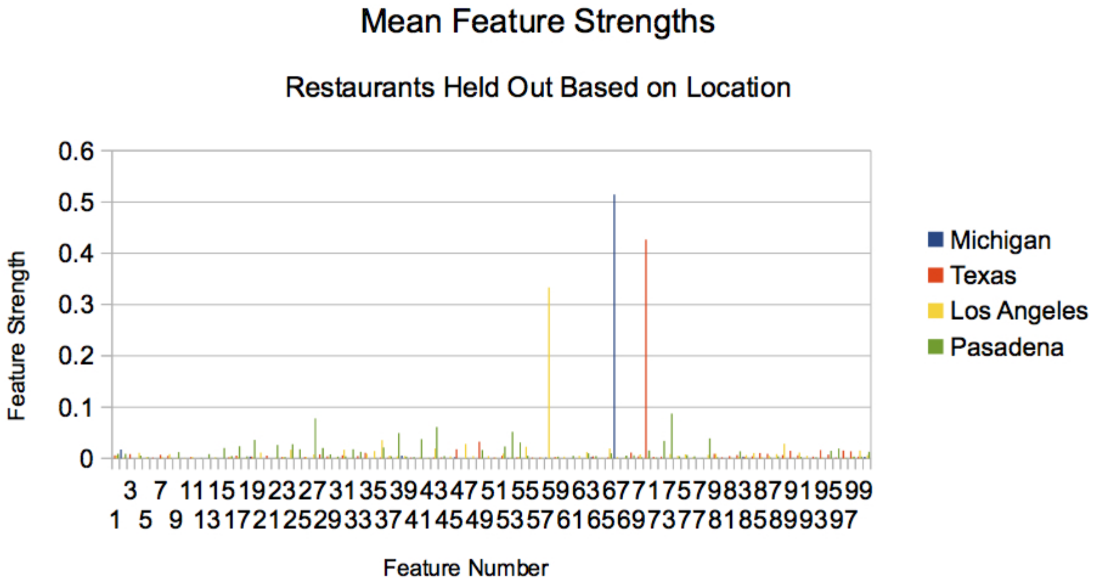

Figure 4: Mean Feature Strength for Second BPF Trial

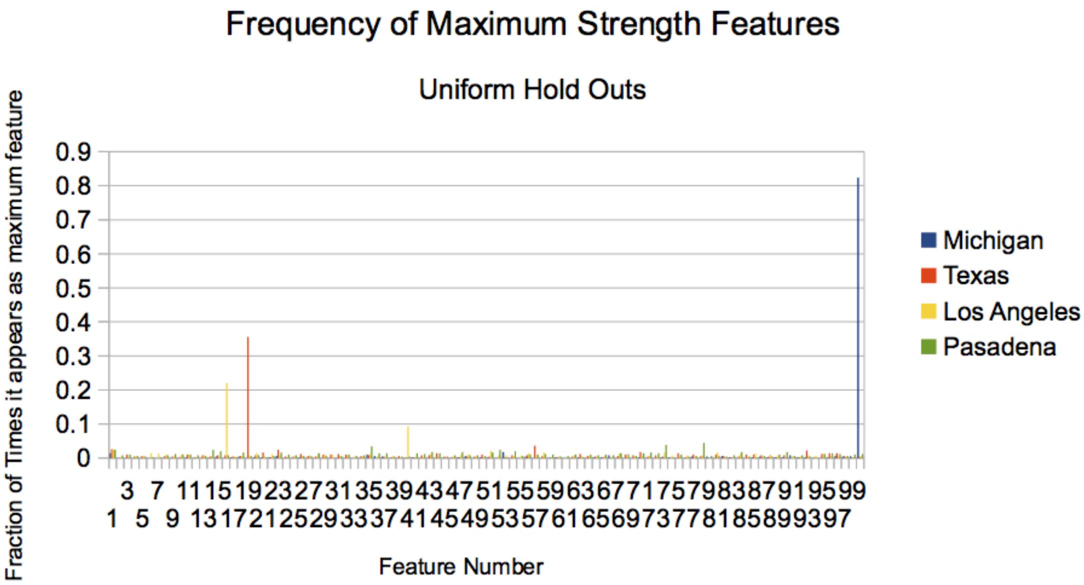

Figure 5: Fraction of Maximum Strength Features for First BPF Trial

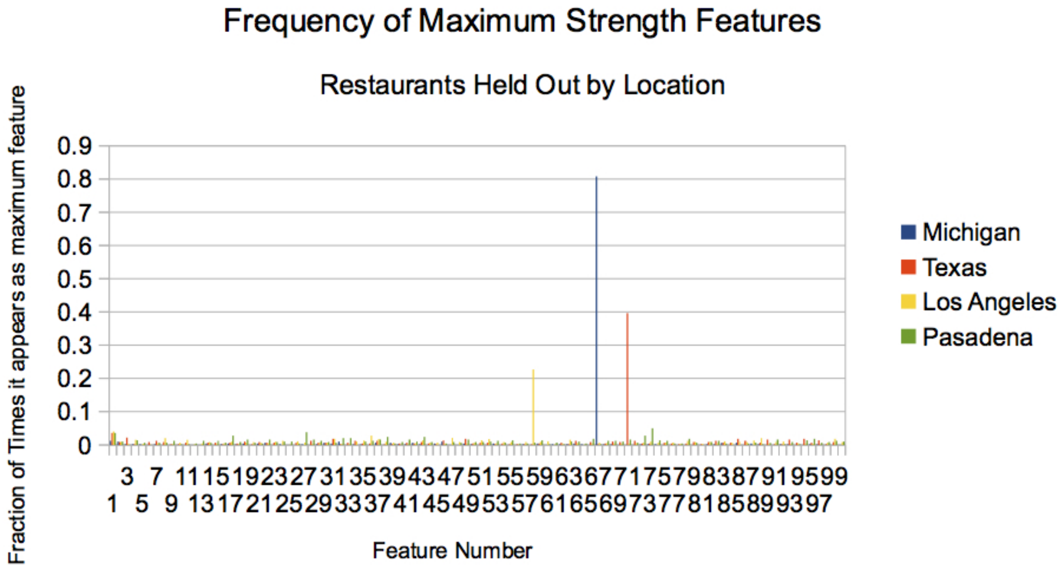

Figure 6: Fraction of Maximum Strength Features for Second BPF Trial

F ZIPF Graphs

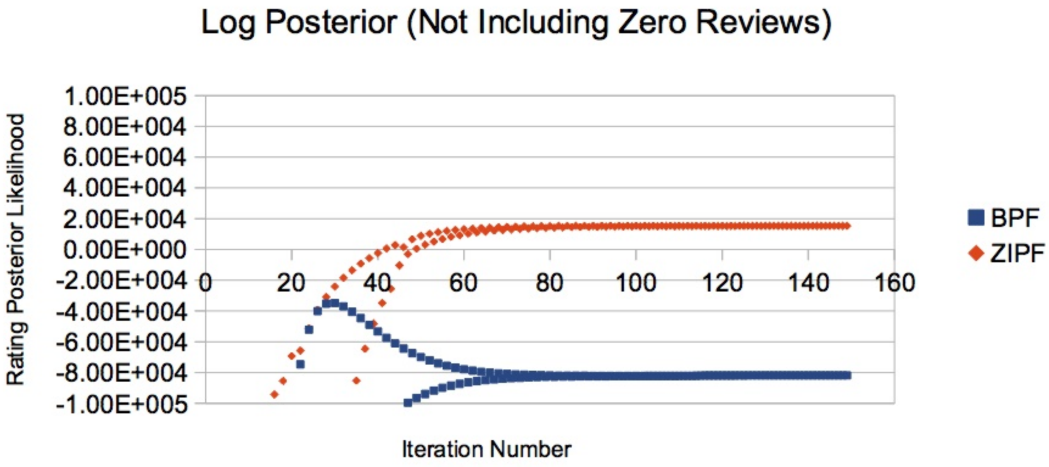

Figure 7: Comparison of Log Posterior (Not Including Zero/Unobserved Reviews)

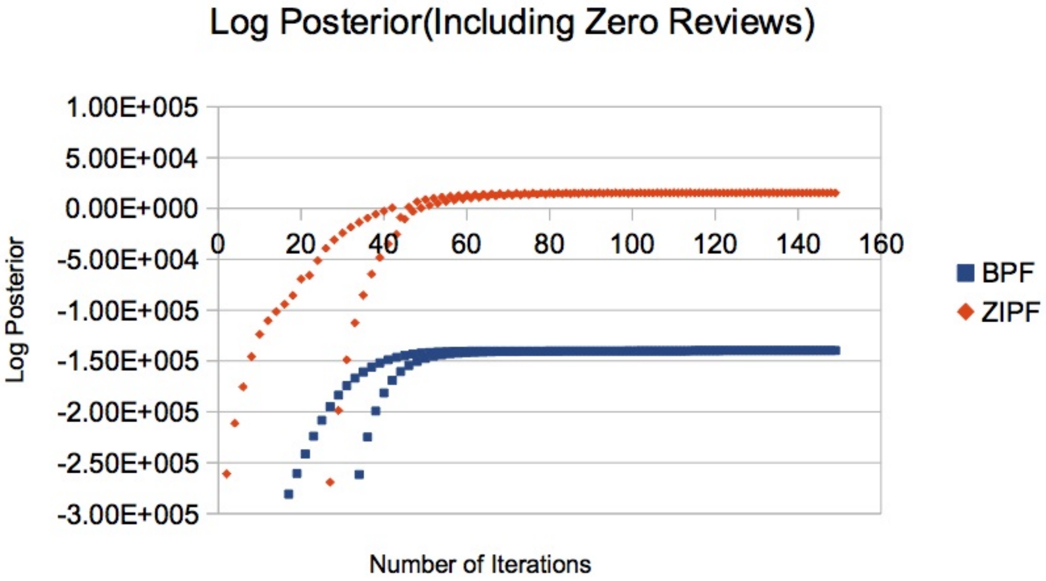

Figure 8: Comparison of Log Posterior (Including Zero/Unobserved Reviews)

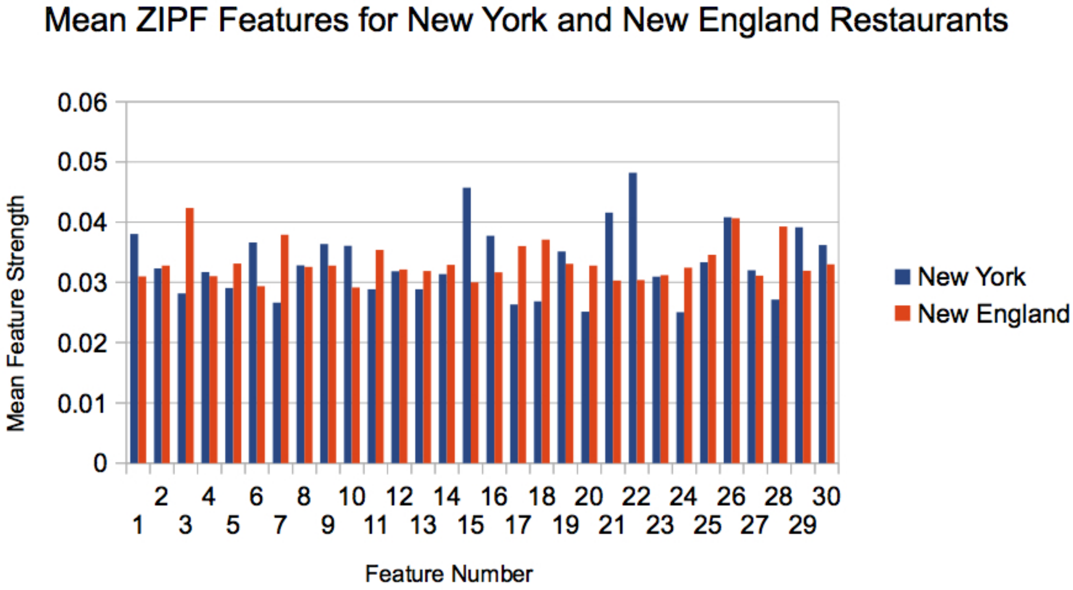

Figure 9: Mean Feature Strengths for ZIPF

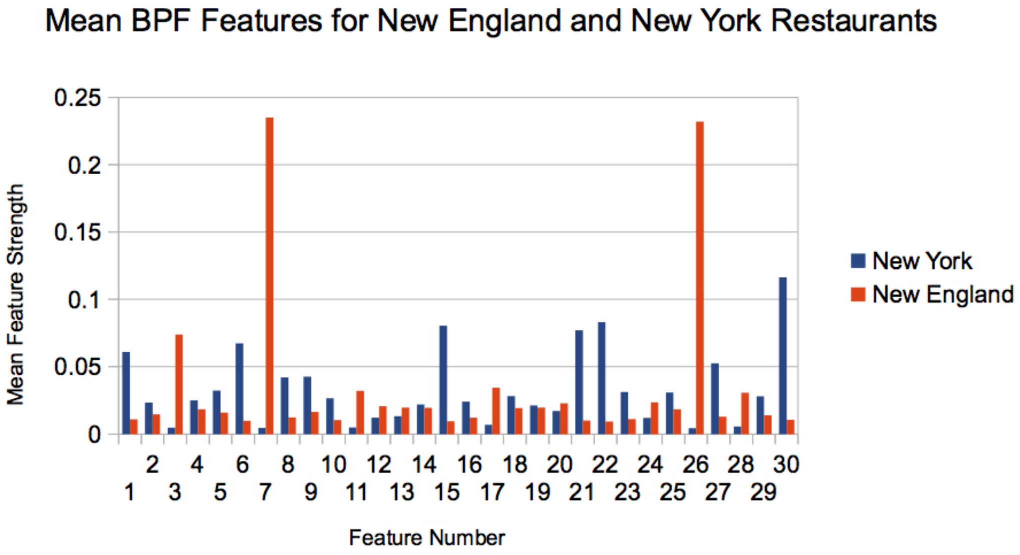

Figure 10: Mean Feature Strengths for BPF