Lambert Matrix Factorization

Arto Klami, Jarkko Lagus, and Joseph Sakaya

University of Helsinki, Department of Computer Science, Finland
{arto.klami,jarkko.lagus,joseph.sakaya}@cs.helsinki.fi

Abstract. Many data generating processes result in skewed data, which should be modeled by distributions that can capture the skewness. In this work we adopt the flexible family of Lambert W distributions that combine arbitrary standard distribution with specific nonlinear transformation to incorporate skewness. We describe how Lambert W distributions can be used in probabilistic programs by providing stable gradient-based inference, and demonstrate their use in matrix factorization. In particular, we focus in modeling logarithmically transformed count data. We analyze the weighted squared loss used by state-of-the-art word embedding models to learn interpretable representations from word co-occurrences and show that a generative model capturing the essential properties of those models can be built using Lambert W distributions.

Keywords: Skewed data · Matrix factorization · Lambert distribution

1 Introduction

Real-valued data is often modeled with probabilistic models relying on normal likelihood, which captures simple additive noise well. Many realistic data generating processes, however, correspond to noise with wider tails. If the noise is still symmetric there are well-understood likelihoods to choose from. For example, student-t, Cauchy and stable distributions can be used for modeling heavy-tailed noise, and the machine learning community has used them for providing *robust* variants of various analysis methods [2,26].

When only one of the tails is heavy, the distribution is called *skewed* – a distribution with heavy left tail has negative skew and a distribution with heavy right tail has positive skew. Such data occurs in many applications: income and wealth are highly skewed, stock market returns are negatively skewed, and many monotonic transformations of symmetric noise result in skewness. There are, however, way fewer tools for modeling skewed data. The skew-normal distribution and its extensions to student-t [4] and Cauchy [3] are the main alternatives, but they are computationally cumbersome and limited in expressing skewness.

We build on the work of Goerg [11], who proposed a flexible family of skewed distributions by combining arbitrary continuous base distribution $F_Z(z)$ with forward transformation $g(z) = ze^{\gamma z}$. He coined it the Lambert $W \times F_Z$ distribution because the corresponding backward transformation $g^{-1}(x)$ uses the Lambert W function [8]. Here γ controls the skewness independently of the other

properties of the distribution, which makes the family a promising solution for general-purpose modeling of skewed data.

Maximum likelihood estimation of Lambert $W \times F_Z$ distribution is possible with an iterated generalized method of moments [11]. We are, however, interested in using the distribution as a building block in more complex systems, such as mixtures, matrix factorizations or arbitrary probabilistic programs, which requires more general inference algorithms. The construction using differentiable transformation of standard density is in principle directly applicable for gradient-based learning, including also posterior inference strategies such as variational approximations [28] and Hamiltonian Monte Carlo [6]. Unfortunately, the mapping $g(z)$ in [11] is not bijective because the range of the Lambert W function is limited, which causes computational issues. We show that an equivalent probability density can be obtained by using a bijective transformation that deviates from W only in regions with small probability, and we provide a computationally efficient distribution that can be used for modeling arbitrary skewed data.

We are particularly interested in the family because it provides a natural basis for fully generative interpretation of representation learning techniques used for estimating word embeddings from co-occurrences of words, such as GloVe [23] and Swivel [25]. Embeddings capturing semantic and syntactic relationships between words can be extracted by modeling logarithmic co-occurrence counts with a linear model. The state-of-the-art models do this by minimizing a weighted squared error between the model and the data, weighting the individual elements by the observed data itself. This allows the model to focus more on modeling co-occurrences that are considered more reliable.

Generative alternatives have been proposed for skip-gram models that learn embeddings based on individual sentences [27,5], but generative variants for the models operating directly on the co-occurrence counts of the whole corpus are missing. Some attempts have been made, but none of these manage to replicate the properties obtained by weighting the least squares loss-based model on the data itself. For example, Vilnis and McCallum [29] model the variance by summing individual terms for each row and column and hence cannot give higher uncertainty for uncommon co-occurrences, Li et al. [19] use empirical Bayes priors centered at the observed data failing to produce a proper generative model, and Jameel and Schockaert [16] model the variances by explicitly analyzing the residuals and do not provide a closed-form model.

We show that the weighted least squares error used by GloVe [23] and Swivel [25] can be interpreted as a negative log likelihood of a distribution that has a number of interesting properties. It is locally Gaussian, it is left skewed, and its variance and mean are negatively correlated. We show that the Lambert $W \times F$ family of distributions allows replicating these properties when we use as the base distribution F a normal distribution controlled by a single parameter μ influencing both the mean and variance. Building on this, we propose a simple probabilistic model that can be used for extracting interpretable embeddings from count data. It combines the above likelihood with an additional latent variable indicating presence or absence of co-occurrence, with probability that

is controlled by the same parameter μ that controls the location and scale of the data generating distribution.

We demonstrate the Lambert matrix factorization technique first in standard matrix factorization application with skewed noise to illustrate that the gradient-based learning scheme works as intended. We then proceed to showcase the algorithm in modeling log-transformed count data, demonstrating also that it achieves accuracy comparable to its closest non-probabilistic comparison method Swivel [25] on standard evaluation metrics for word embeddings.

2 Lambert $W \times F$ Distribution

The Lambert $W \times F$ distribution family [11] is defined through an input/output system driven by some standard distribution $F_Z(z)$ with zero mean and unit variance. Latent inputs z drawn from F_Z are transformed with skewing function

$$x = g(z) = ze^{\gamma z}, \tag{1}$$

where γ is a parameter controlling the skewness. While the specific choice of the transformation may seem arbitrary, is has several desirable properties that are difficult to achieve at the same time: it is easy to differentiate, reduces to identity for $\gamma = 0$, and allows easy change of skew direction ($\gamma > 0$ and $\gamma < 0$ correspond to right and left skewness, respectively). It hence provides an interesting opportunity for probabilistic programming since it allows skewing any base distribution by a learnable quantity.

The probability density of x can be expressed in terms of the base density $p_Z(z)$ using change of variables as

$$p_X(x|\gamma) = p_Z(g^{-1}(x)) \left| \frac{dg^{-1}(x)}{dx} \right|, \tag{2}$$

where the absolute value of the Jacobian of the inverse transformation $z = g^{-1}(x)$ accounts for the change of volume under the nonlinear transformation. The inverse of (1) is

$$z = g^{-1}(x) = W_0(\gamma x)/\gamma, \tag{3}$$

where $W_0(\cdot)$ is the principal branch of the Lambert W function that gives rise to the name of the distribution family. The W function is illustrated in Figure 2 (top left), showing its two real branches $W_0(x)$ and $W_{-1}(x)$. The function has numerous applications in differential calculus, quantum mechanics, fluid dynamics etc. Consequently, its properties have been studied in detail [8].

The derivatives of W (for $x \neq -1/e$) can be computed using

$$\frac{dW(x)}{dx} = \frac{1}{x + e^{W(x)}} \tag{4}$$

and consequently the density in (2) becomes

$$p_Z(W(\gamma x)/\gamma) \frac{1}{\gamma x + e^{W(\gamma x)}}.$$

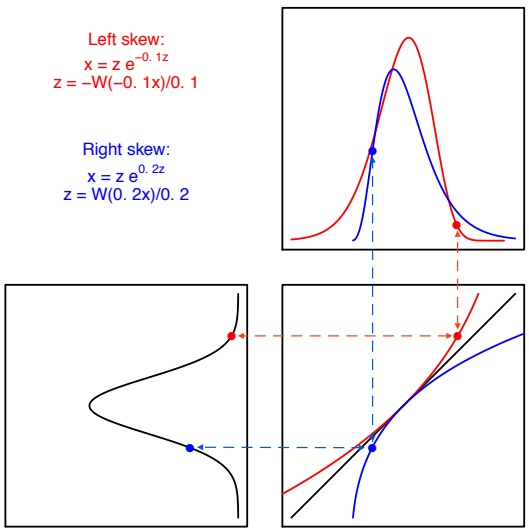

Fig. 1. Illustration of the Lambert $W \times F$ distribution. Draws from a standard distribution (bottom left), here $\mathcal{N}(0, 1)$, are transformed to draws from Lambert $W \times F$ distribution (top right) by $g(z) = ze^{\gamma z}$, where γ controls skewness. For $\gamma < 0$ the transformation is convex and the resulting distribution is left skew (red), whereas for $\gamma > 0$ the transformation is concave and the distribution is right skew (blue).

We illustrate the density and the construction based on a standard distribution (here normal) in Figure 1. For practical modeling purposes the skewed distribution would then be transformed with further location-scale transformation

$$y = \sigma x + \mu, \tag{5}$$

which results in the final density

$$p_Y(\mu, \sigma, \gamma) = \frac{p_Z(\frac{1}{\gamma} W(\frac{\gamma(y-\mu)}{\sigma}))}{\gamma(y-\mu) + \sigma e^{W(\frac{\gamma(y-\mu)}{\sigma})}}.$$

We denote this density with $\mathcal{W}_F(\mu, \sigma, \gamma)$, where F tells the base distribution.

Goerg [11] provides extensive analysis of the family, showing, for example, closed-form expressions for various moments of the distribution when $F = \mathcal{N}$, and that for wide range of γ around zero the parameter is related to the actual skewness of the distribution as $\text{skew}(X) = 6\gamma$. This implies that relatively small values of γ are likely to be most useful.

2.1 Practical Implementation for Probabilistic Programming

Using Lambert $W \times F$ distributions in probabilistic programs is in principle straightforward. Since the transformation is differentiable, we can construct a transformed distribution by combining any base distribution with a chain of two transformations: First we skew the draws using (1) and then perform the location-scale transformation (5). The backwards transformations and their derivatives required for evaluating the likelihood are known, and sampling from the distribution is easy. However, for a practical implementation for probabilistic programs we need to pay attention to some details.

Computation of W The Lambert W function does not have a closed-form expression and hence requires iterative computation, typically carried out by the Halley's method. An initial guess $w \approx W(x)$ is iteratively refined using

$$w_{j+1} = w_j - \frac{w_j e^{w_j} - x}{e^{w_j}(w_j + 1) - \frac{(w_j+2)(w_j e^{w_j} - x)}{2w_j+2}},$$

where roughly 10 iterations seem to suffice in most cases. While this reasonably efficient procedure is available in most scientific computing languages, using an iterative algorithm for every likelihood evaluation is not very desirable. However, $W(x)$ is a scalar function that in practical applications needs to be evaluated only for a relatively narrow range of inputs. Hence it is easy to pre-compute and use tabulated values during learning, converting a trivial pre-computation effort into fast evaluation with practically arbitrarily high precision. The derivatives of $W(x)$ can be computed using the same pre-computation table due to (4).

Finite support The support of the Lambert $W \times F$ distribution described in [11] is finite. For $\gamma < 0$ the support is $[-\infty, -\frac{1}{\gamma e})$ and for $\gamma > 0$ it is $(-\frac{1}{\gamma e}, \infty]$. This imposes strict limits for the set of possible μ, σ and γ values compatible with given data y. This limitation is a property of the distribution itself and will hold for reasonable parameter values that describe the data well. However, during learning it poses unnecessary computational difficulties since naive unconstrained optimization over the parameters would often result in zero probability. To avoid this we would need to conduct optimization over a constraint set determined by interplay between μ, σ and γ.

Instead of constrained optimization we modify the density itself to have small probability for data outside the support of the original formulation, by altering the forward and backward transformations in a way that allows computation of the log-likelihood for arbitrary values. This solution has the advantage that it not only removes need for constrained optimization but it also makes the distribution more robust for outliers, allowing individual data points to fall outside the support because of additional noise components not taken into account by the model.

Our modified formulation uses a piece-wise extension of $W(x)$ defined as

$$W_e(x) = \begin{cases} W(x) & \text{if } x \geq d \\ W(d) + \frac{x-d}{x+e^{W(d)}} & \text{if } x < d \end{cases}$$

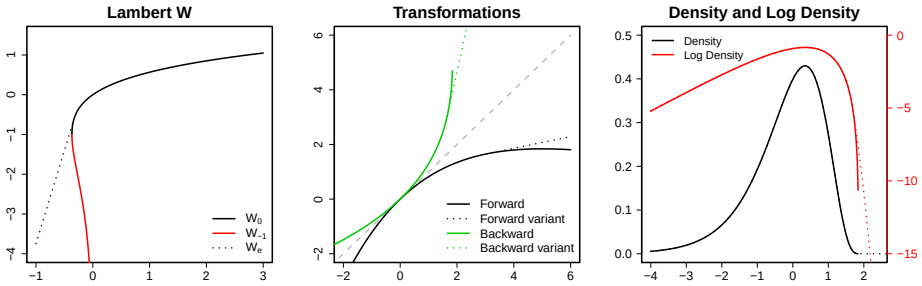

Fig. 2. Illustration of our modified Lambert W function (left) that replaces the W_{-1} branch with linear continuation of W_0, and its implication for the forward and backwards transformations (middle) for rather large skewness $\gamma = -0.2$. The forward transformations start deviating only at $z = 3$. When using $N(0, 1)$ as the base distribution only 0.1% of samples fall into that region. The corresponding probability densities (right) are consequently nearly identical, the only difference being that the modified version has support over the whole real line but with extremely low probability for $x > 2$ excluded altogether by the original formulation.

with suitable choice of d that is between $-e^{-1}$ and 0 – in our experiments we used $d = -0.9e^{-1}$ but other values close to $-e^{-1}$ result in similar behavior. This modification replaces the secondary branch $W_{-1}(x)$ (and small part of the main branch) with linear extension that matches the main branch $W_0(x)$ and its derivative at d. While this extension clearly breaks down characteristic properties of the Lambert W function, making it inapplicable in most other uses of the function, the Lambert $W \times F$ family of distributions is mostly concerned about values for which $W_e(x) = W(x)$.

The resulting transformations are illustrated in Figure 2 to demonstrate how they only influence the very low-probability tail of the distribution, giving finite but very small probabilities for samples that would be impossible under standard Lambert $W \times F$ distribution, enabling gradient-based learning.

On skewness With $\gamma = 0$ the backward transformation (3) involves division by zero. Any practical implementation should hence implement the non-skewed scenario as a special case. When learning skewness we also want to avoid extreme values for improved stability. We re-parameterize $\gamma = \gamma_m \tanh \hat{\gamma}$ so that $\hat{\gamma}$ can take any real value while limiting $|\gamma| < \gamma_m$. We use $\gamma_m = 0.2$ in our experiments, corresponding to roughly maximum skewness of 1.2.

2.2 Lambert Matrix Factorization

Probabilistic matrix factorization refers to the model

$$x|\theta, \beta, \phi \sim p(x|\theta^T\beta, \phi), \qquad \theta \sim p(\theta), \quad \beta \sim p(\beta),$$

where a bilinear term $\theta^T\beta$ models the mean of the generating distribution and ϕ refers to all other parameters of the distribution, such as the variance in case of normal likelihood. This formulation covers practical models proposed for recommender engines [21], probabilistic interpretation of PCA [15] and CCA [17], group factor analysis [18], and non-negative matrix factorization [22]. All these models differ merely in the choice of priors and likelihoods.

Lambert $W \times F$ distributions as presented here are designed to be used with arbitrary probabilistic models, and hence are applicable for matrix factorization as well. We use $\mathcal{W}_F(\theta^T\beta, \sigma, \gamma)$ as the likelihood, modeling the mean still with a bilinear term. We can then either treat $\phi = \{\sigma, \gamma\}$ as global parameters to modulate the scale and skewness of the noise, or assume more complicated structure for them as well. In fact, we will later make also σ a function of $\theta^T\beta$.

3 Modeling Dyadic Data

As a practical application of the Lambert $W \times F$ distribution in matrix factorization, we consider the problem of learning interpretable distributed representations for dyadic data, observed as N triplets (i_n, j_n, C_n) interpreted as object i occurring together with item j with associated scalar count c for the number of such co-occurrences. The classical example concerns learning word embeddings based on observing that word j occurs c times in the context of word i, providing for each word a vectorial representation θ_i. Another example considers media consumption, where user i has, for example, listened to the song j for c times and high counts are interpreted as expressing interest for the song.

Such data can be thought of having been generated by drawing N independent observations from a joint density $p(i, j)$ over pairs of i and j, corresponding to each count C_{ij} following a Binomial distribution $\text{Binom}(N, p_{ij})$. Since N is here typically very large and each p_{ij} is small since they need to sum up to one over IJ entries, the distribution can accurately be approximated also by $\text{Poisson}(\lambda_{ij})$ where $\lambda_{ij} = Np_{ij}$. We will later use both interpretations when deriving specific elements of our proposed generative model.

To extract meaningful vectorial representations from such data, we should model ratios of co-occurrence probabilities with linear models. In practice this is typically achieved by modeling point-wise mutual information (or its empirical estimates) with

$$\theta_i^T\beta_j \approx \log \frac{p(j,i)}{p(i)p(j)} \approx \log \frac{C_{ij}N}{C_iC_j}.$$

This can equivalently be written as matrix factorization with row and column biases for the logarithmic counts themselves, letting the bias terms learn the negative logarithmic counts of the marginals [23]:

$$\theta_i^T\beta_j + a_i + b_j - \log N \approx \log C_{ij}.$$

The state of the art techniques solve the above problem by minimizing the squared error between the approximation and the logarithmic counts. Importantly, they weight the individual terms of the loss by a term that increases

roughly linearly as a function of C_{ij}, the actual count, and hence exponentially as a function of $\log C_{ij}$ that is being modeled. Putting more effort in modeling large counts is intuitively reasonable, but it makes generative interpretations of these models difficult.

3.1 Generative Embedding Model for Dyadic Data

We propose to model such dyadic data with the following probabilistic program. It assumes bilinear form with bias terms for a location parameter μ_{ij} and uses the same parameter for controlling three aspects of the data: the probability of observing a co-occurrence, the expected logarithmic count, and the variance.

The first stage is achieved by drawing a binary variable h_{ij} from a Bernoulli distribution with parameter $f(\mu_{ij}) = 1 - e^{-e^{\mu_{ij}}}$ (See Section 3.2). For $h_{ij} = 1$ the model then generates logarithmically transformed observed data $y_{ij} = \log C_{ij}$ from a Lambert $W \times F$ distribution with mean μ_{ij} and $\sigma_{ij}(\mu_{ij})$. Finally, the observed count is transformed through an exponential linear unit to guarantee positive output (since $\log C \geq 0$ for $C > 0$).

The full model is

$$\begin{aligned} \mu_{ij} &= \theta_i^T \beta_j + a_i + b_j - \log N \\ h_{ij} &\sim \text{Bernoulli}(f(\mu_{ij})) \\ y_{ij}|h_{ij} = 0 &= -\infty \\ y_{ij}|h_{ij} = 1 &= \begin{cases} \bar{y}_{ij} & \text{if } \bar{y}_{ij} \geq 0 \\ \alpha(e^{\bar{y}_{ij}} - 1) & \text{if } \bar{y}_{ij} < 0 \end{cases}, \text{ where} \\ \bar{y}_{ij} &\sim \mathcal{W}_N(\mu_{ij}, \sigma(\mu_{ij}), \gamma), \end{aligned} \tag{6}$$

with suitable priors assigned for the parameters. In practice the constant $\log N$ can be incorporated as a part of either bias term. The model includes several non-obvious elements that are critical in modeling log-transformed counts, listed below and motivated in detail in the following subsections.

1. A single location parameter μ controls a) the probability $f(\mu)$ of observing a co-occurrence, b) the expected logarithmic count μ of co-occurrences, and c) the variance $\sigma(\mu_{ij})$ of the logarithmic counts
2. The likelihood function is left-skewed, belonging to the Lambert $W \times F$ family and using normal distribution as the base distribution
3. The variance $\sigma(\mu_{ij})^2$ is maximal around $\mu = 1$ and decreases exponentially towards both infinities

3.2 Missing Co-occurrences

Many dyadic data sets are very sparse, so that most co-occurrences are missing. Various solutions to this challenge have been proposed: GloVe [23] ignored the missing co-occurrences completely by setting the corresponding weight in loss function to zero, likelihood evaluations in hierarchical Poisson factorization scale

linearly in the observed co-occurrences [12] but only for models that are linear in the actual count space, and Swivel [25] models missing co-occurrences based on an alternative loss that penalizes for over-estimating the point-wise mutual information and uses parallel computation for fast evaluation.

We replace these choices with a Bernoulli model that determines whether a particular co-occurrence is seen, so that the probability of generating a zero comes directly from the assumed process of Binomial draws for each pair. By approximating $\text{Binom}(N, p_{ij})$ with $\text{Poisson}(\lambda_{ij})$ we immediately see that $p(C_{ij} = 0) = e^{-\lambda_{ij}}$. Since we model the expected logarithmic count by μ_{ij} we note that $\lambda_{ij} = e^{\mu_{ij}}$ and hence $p(C_{ij} = 0) = e^{-e^{\mu_{ij}}}$ and $f(\mu_{ij}) = p(C_{ij} = 1) = 1 - e^{-e^{\mu_{ij}}}$. In other words, the underlying generative assumption directly implies that the same parameter controlling the expected value should control also the probability of success in the Bernoulli model.

The log-likelihood of this model can be contrasted with the loss in Swivel [25] assumed for measuring the error for the pairs that do not co-occur. They use a logistic loss between the model and data that corresponds to having observed a single co-occurrence, in an attempt to allow some leeway for the model. The loss Swivel uses, here written for logarithmic counts instead of point-wise mutual information for easier comparison, is $\log(1 + e^{\mu_{ij}})$. Our model, in turn, results in negative logarithmic likelihood $-\log p(C_{ij} = 0) = e^{\mu_{ij}}$, and hence it penalizes more heavily for observing no co-occurrences for large μ.

3.3 Likelihood

Let us next take a look at the likelihood function, which we assume to be Lambert $W \times N$ with left-skewness indicated by $\gamma < 0$. Furthermore, we assume the scale $\sigma(\mu)$ of the distribution is controlled by the same parameter μ that controls the mean μ. This has close connection with the weighted least squares error [23,25],

$$-\frac{1}{2}\sum_{i,j} h(e^{y_{ij}})\|y_{ij} - \mu_{ij}\|^2,$$

where y_{ij} is the logarithmic count and $h(\cdot)$ is a function that grows according to the observed count itself. The loss is superficially similar to the negative log-likelihood of normal distribution, merely replacing the constant precision τ with one parameterized by the observed data y.

Since the weight depends on the observed data the loss does not match an actual normal distribution. However, if we assume it corresponds to negative log-likelihood of some distribution then we can interpret the loss function directly as a log-likelihood by normalizing it suitably. Alternatively, we can search for equiprobability contours of the loss function around any given μ by solving for

$$\frac{1}{2}\tau(x)(x - \mu)^2 = c$$

for some constant $c > 0$. Plugging in $\tau(x) = e^{\alpha x}$ results in two solutions, one at each side of μ:

$$x = \mu + \frac{2}{\alpha} W\left(\frac{a\sqrt{c}}{\sqrt{2}}\sqrt{e^{-\alpha\mu}}\right) \quad \text{and} \quad x = \mu + \frac{2}{\alpha} W\left(-\frac{a\sqrt{c}}{\sqrt{2}}\sqrt{e^{-\alpha\mu}}\right).$$

For small c, via linearization of the Lambert W, these correspond to

$$x = \mu \pm \frac{\sqrt{2}}{\sqrt{e^{\alpha\mu}}}\sqrt{c}$$

which matches the equiprobability contours of normal distribution with precision $\tau = \sqrt{e^{\alpha\mu}}$. For larger values the solution on the left side is further away from μ than the one on the right, corresponding to negative skewness. Furthermore, the separation of the distances is controlled by the Lambert W function that is used as the backward transformation in the Lambert $W \times F$ family of distributions.

The derivation shows that the likelihoods induced by GloVe (which uses $h(y_{ij}) = e^{3/4 y_{ij}}$) and Swivel ($h(y_{ij}) = 0.1 + 0.25 e^{1/2 y_{ij}}$) correspond to negatively skewed likelihoods where the variance and mean are controlled by the same parameter. The derivation does not imply direct correspondence with the Lambert $W \times F$ family since the $W(x)$ function is here used in slightly different fashion. Nevertheless, empirical comparison with the GloVe and Swivel losses normalized into likelihoods and the Lambert W family with suitable skewness reveal that the family is able to replicate the behavior accurately (Figure 3). Note that GloVe and Swivel losses only determine relative precisions, not absolute ones; for the illustration we re-scaled them to match the overall scaling of our model.

3.4 Variance Scaling

The final missing piece concerns the function $\sigma(\mu)$ used for scaling the variance. As mentioned above, GloVe and Swivel assume here simple exponential formulas for the precision, $e^{3/4y}$ and $0.1 + 0.25e^{1/2y}$ respectively. To derive a justified variance scaling we start with the series expansion

$$\mathrm{Var}[\log C] \approx \frac{1}{\mathbb{E}[C]^2}\mathrm{Var}[C],$$

which holds for large values. Since $C \propto e^{\mu}$, we see the variance of $y = \log C$ should indeed decrease exponentially with the mean, matching the general intuition of GloVe and Swivel.

Many of the observations, however, are for C in single digits, for which the approximation does not hold. Whenever $C = 0$ the variance becomes infinite, but our generative model takes care of these instances with the h variable. Hence, what we need is a formula for the variance of $\log C$ when $C > 0$, applicable for small C. We are not aware of simple analytic expressions for this, but we can easily analyze the empirical behavior of large collection of samples drawn from the Binomial distribution assumed to generate the data. Figure 4 shows that for

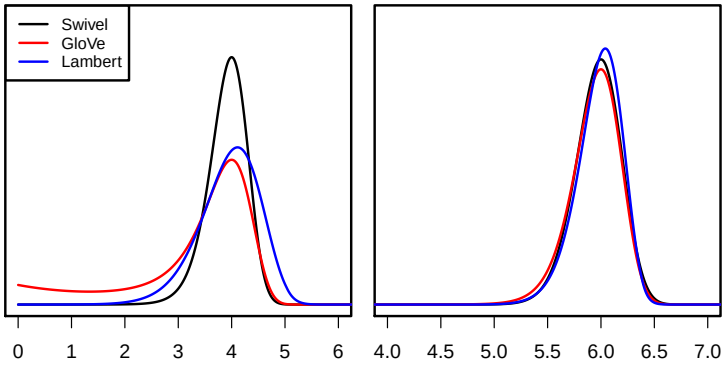

Fig. 3. Comparison of the densities induced by GloVe and Swivel and the explicit density modeled by Lambert W distribution in our model, using $\gamma = -0.1$. The x-axis corresponds to logarithmic count. For small expected counts (left) GloVe induces likelihood that is even more left-skewed (due to pushing precision to zero for very small values), but for expected log-counts around 6 (right) the distributions are almost identical for all three methods.

large μ the variance follows the analytic expression $e^{-\mu}$ and for large negative μ it follows e^{μ}. The two regions smoothly transition to each other in the middle, with maximum value of 0.318 reached at $\mu = 1$. This can be approximated by an average of e^{μ} and $e^{-\mu}$, resulting in hyperbolic cosine (for precision) or hyperbolic secant (for variance). If we further match the maximal variance at $\mu = 1$, we get

$$\sigma(\mu) = \sqrt{\frac{0.318}{\cosh(\mu - 1)}}$$

that is almost exact replicate of the empirical variance except that for large μ the inputs are off by one. We allow for this slight overestimation of variance to avoid excessively narrow distributions for very large counts.

3.5 Computation

Computing the likelihood requires evaluating the loss for all co-occurrences, including ones that are not observed in the data. For efficient maximum likelihood estimation we adopt the parallel computation strategy presented in Swivel [25]. The data is split into shards, $k \times k$ submatrices, that roughly maintain the frequency characteristics of the original data, giving rise to efficient mini-batch training. Further speedups can be obtained by performing asynchronous updates in parallel so that each worker fetches and stores parameters on a central parameter server in a lock-free fashion [9]. Because the Lambert function $W(x)$ and its derivative are required for only a narrow range of inputs, they can be pre-computed and tabulated for fast evaluation.

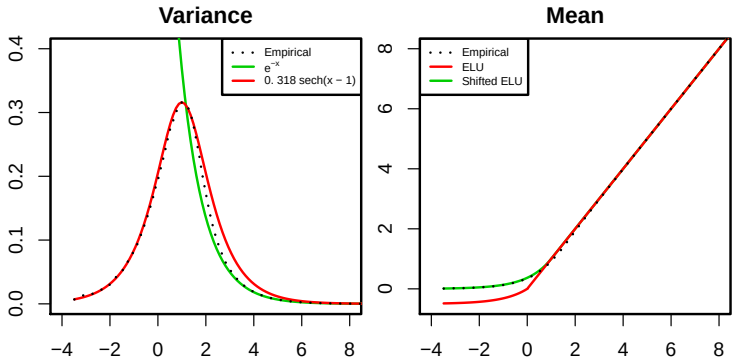

Fig. 4. Logarithm of data drawn from Binomial model gives raise to specific non-linear mapping from the mean parameter to variance and mean of the distribution. The left plot depicts the variance of $\log C$ for $C > 0$ and we see the empirical variance (dotted black line) is well captured by the hyperbolic secant function (red line). For positive parameters also the analytic expression (green line) is a good match. The right plot depicts the mean of C for $C > 0$. The empirical curve (dotted black line) can be matched by re-scaled and shifted exponential linear unit (ELU), but in practice standard ELU (red line) that gives negative values for negative inputs works better since it does not map the zero observations to minus infinity in the parameter space.

3.6 Side remark: Why not model counts directly?

Given the assumed generative story of $C \sim \text{Binomial}(N, p)$, a natural alternative would be to model the counts directly with a Poisson matrix factorization model [1] that passes $\theta^T\beta$ via the exponential link function. Such model would achieve similar variance scaling and can be implemented using the exponential family embeddings framework [24]. Another natural alternative would be to use negative-binomial likelihood [30,31] that Hashmimoto et al. [13] used to reproduce the objective function of GloVe [23]. We focus on modeling logarithmic counts with Lambert distributions because of the added flexibility: Our model allows tuning the skewness and variance scaling properties as desired, instead of being forced to adopt the exact choices Poisson and negative binomial distributions induce that may be suboptimal for the data.

4 Experiments

4.1 Lambert Matrix Factorization for Skewed Data

As a sanity check we run the model on artificial data sampled from the model itself. Given randomly generated θ and β, we sample data from the process

$$z \sim \mathcal{N}(0, 1), \qquad x = \theta^T\beta + ze^{\gamma z},$$

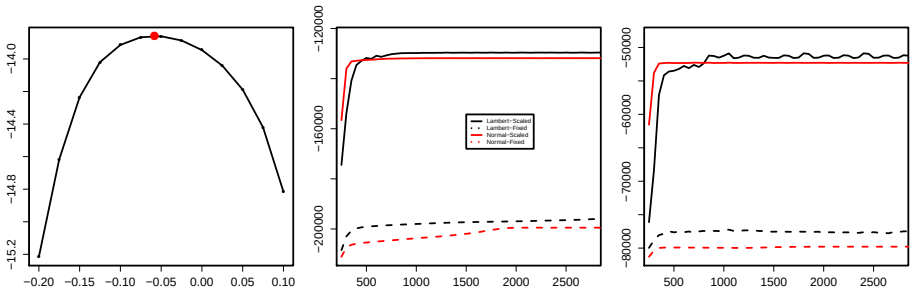

Fig. 5. *Left:* Demonstration of Lambert W for artificial data generated from the model with $\gamma = -0.05$. The plot shows logarithmic likelihood for a range of skew parameters, and the red dot indicates the solution obtained when learning γ as well. The model is shown to be robust for small variations of skewness, but the optimal value results in highest likelihood and can be found with gradient-based learning algorithms. *Center and right*: Progress of training (center) and test (right log-likelihoods as function of the iteration. The full model with skewed distribution and variance scaling (black solid line) outperforms alternatives that are either forced to use symmetric normal likelihood (red lines) or constant variance for all entries (dashed lines).

using $\gamma = -0.05$ to produce slightly left skewed data. We then perform gradient-based learning for θ and β using fixed γ to evaluate the log-likelihood of the observed data under various choices of skewness. The first plot in Figure 5 illustrates how the likelihood is maximized with the right value, and this is verified by performing inference over γ; we find $\hat{\gamma} = -0.058$ as the learnt parameter.

4.2 Representation Learning for Co-occurrence Data

Next we evaluate the importance of individual elements of our model in generative description of dyadic data. We sample a total of $N = 400,000$ tokens for $I = 1000$ and $J = 200$ with low-rank (10 factors) linear structure for p_{ij}. We then model the resulting dyadic data $\{C_{ij}, i, j\}$ using variants of the model (6). In particular, we compare the full model against alternatives that omit one or more of the elements considered crucial parts of the generative model. We try both replacing the Lambert distribution with symmetric normal distribution and replacing the $\sigma(\mu)$ function with constant σ to remove the variance scaling.

We train the models on data where only 70% of the co-occurrences – missing or not – are observed and the rest are left as test data, to demonstrate the added benefit of working with generative models. Figure 5 plots the training and test log-likelihoods for the variants mentioned above, showing how the algorithm converges rapidly. Both scaling the variances and using left-skewed distributions are found to be important – each element alone improves the likelihoods and the full model combining both is the best.

4.3 Word Representations

Table 1. Performance of the proposed Lambert matrix factorization and the comparison method Swivel in standard word embedding tasks. Higher is better for all metrics.

Metric	Swivel	LMF
Word Similarity, Finkelstein et al. [10]	0.704	**0.714**
Word Relatedness, Finkelstein et al. [10]	0.578	**0.608**
MEN dataset, Bruni et al. [7]	0.674	**0.680**
Stanford Rare Word, Luong et al. [20]	**0.403**	0.397
SimLex-999, Hill et al. [14]	0.291	**0.332**

Finally, we compare the proposed model against its closest non-generative comparison Swivel [25] on a word embedding task. Swivel uses two losses, one for positive co-occurrences and one for the missing ones, whereas our model performs maximum likelihood inference for the model (6) and hence combines both elements in a single loss. For efficient computation for large data both models use the sharding technique [25]. We use a one gigabyte snapshot of the Wikipedia dump data with vocabulary size of 50K and shard size of 1024 to infer embeddings of 300 dimensions. We measure performance using standard evaluation metrics, presenting the results in Table 1. The proposed method matches the accuracy of Swivel in all comparisons, showing that our probabilistic program matches the key properties of word embedding models.

5 Discussion

Even though almost no noise distribution is truly symmetric, the tools available for handling skewed distributions are relatively limited. We extended the work of Goerg [11] to create an easy-to-use skewed family of distributions that can be plugged in to various probabilistic programming environments. Besides use as a likelihood, it can also be used as part of a variational approximation for model parameters due to its reparameterization property [28].

We demonstrated the flexibility of the likelihood family by using it as a building block in purely generative model for learning embeddings from dyadic data. We analyzed the loss functions proposed for this task by GloVe [23] and Swivel [25] and showed that their squared errors weighted by the observed data itself can be interpreted as left-skewed distributions where the skewing is performed with the Lambert W function.

Our main goal was to introduce the probability density and its computational facilities, and to provide a proof-of-concept application. The proposed model roughly matches the accuracy of Swivel in various word embedding evaluation tasks, but we did not seek to provide maximally accurate embeddings. Instead, we see the most likely use cases for our solution in more complex models that

build on the same principles but include more hierarchy instead of just learning simple vectorial representation, for example in form of time-evolving embeddings [5] and mixtures for polysemous embeddings [27]. We also note that even though we here performed maximum a posteriori analysis for easier comparison, the machinery presented here would directly allow full posterior analysis as well.

Acknowledgements

The project was supported by Academy of Finland (grants 266969 and 313125) and Tekes (Scalable Probabilistic Analytics).

References

1. Ailem, M., Role, F., Nadif, M.: Sparse poisson latent block model for document clustering. IEEE Transactions on Knowledge and Data Engineering **29**(7), 1563–1576 (2017)
2. Archambeau, C., Delannay, N., Verleysen, M.: Robust probabilistic projections. In: Proc. of the 23rd International Conference on Machine Learning. pp. 33–40 (2006)
3. Arnold, B., Beaver, R.J.: The skew-Cauchy distribution. Statistics & Probability Letters **49**, 285–290 (2000)
4. Azzalini, A., Capitanio, A.: Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t distribution. Journal of Royal Statistical Society Series B **65**, 367–389 (2003)
5. Bamler, R., Mandt, S.: Dynamic word embeddings. In: Proc. of the 34th International Conference on Machine Learning (2017)
6. Betancourt, M.: A conceptual introduction to Hamiltonian Monte Carlo. Tech. rep., arXiv:1701.02434 (2017)
7. Bruni, E., Boleda, G., Baroni, M., Tran, N.K.: Distributional semantics in technicolor. In: Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1. pp. 136–145. ACL '12, Association for Computational Linguistics (2012)
8. Corless, R., Gonnet, G., Hare, D., Jeffrey, D., Knuth, D.: On the Lambert W function. Advances in Computational Mathematics **5**(1), 329–359 (1993)
9. Dean, J., Corrado, G.S., Monga, R., Chen, K., Devin, M., Le, Q.V., Mao, M.Z., Ranzato, M., Senior, A., Tucker, P., Yang, K., Ng, A.Y.: Large scale distributed deep networks. In: Advances in Neural Information Processing Systems 25, pp. 1223–1231 (2012)
10. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: The concept revisited. In: Proc. of the 10th International Conference on World Wide Web. pp. 406–414. ACM (2001)
11. Goerg, G.M.: Lambert W random variables – a new family of generalized skewed distributions with applications to risk estimation. The Annals of Applies Statistics **5**(3), 2197–2230 (2011)
12. Gopalan, P., Hofman, J.M., Blei, D.M.: Scalable recommendation with hierarchical poisson factorization. In: Proc. of the 31st Conference on Uncertainty in Artificial Intelligence. pp. 326–335 (2015)
13. Hashimoto, T.B., Alvarez-Melis, D., Jaakkola, T.S.: Word, graph and manifold embedding from markov processes. CoRR **abs/1509.05808** (2015)

14. Hill, F., Reichart, R., Korhonen, A.: Simlex-999: Evaluating semantic models with genuine similarity estimation. Computational Linguistics **41**, 665–695 (2015)
15. Ilin, A., Raiko, T.: Practical approaches to principal component analysis in the presence of missing data. Journal of Machine Learning Research **11**, 1957–2000 (2010)
16. Jameel, S., Schockaert, S.: D-GloVe: A feasible least squares model for estimating word embedding densities. In: Proc. of the 26th International Conference on Computational Linguistics. pp. 1849–1860 (2016)
17. Klami, A., Virtanen, S., Kaski, S.: Bayesian canonical correlation analysis. Journal of Machine Learning Research **14**, 965–1003 (2013)
18. Klami, A., Virtanen, S., Leppäaho, E., Kaski, S.: Group factor analysis. IEEE Transactions on Neural Networks and Learning Systems **26**(9), 2136–2147 (2015)
19. Li, S., Zhu, J., Miao, C.: A generative word embedding model and its low rank positive semidefinite solution. In: Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1599–1609 (2015)
20. Luong, T., Socher, R., Manning, C.D.: Better word representations with recursive neural networks for morphology. In: Proc. of the 17th Conference on Computational Natural Language Learning, CoNLL 2013. pp. 104–113 (2013)
21. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in neural information processing systems. pp. 1257–1264 (2008)
22. Paisley, J., Blei, D., Jordan, M.: Handbook of Mixed Membership Models and Their Applications, chap. Bayesian nonnegative matrix factorization with stochastic variational inference. Chapman and Hall (2014)
23. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014)
24. Rudolph, M., Ruiz, F., Mandt, S., Blei, D.: Exponential family embeddings. In: Advances in Neural Information Processing Systems. pp. 478–486 (2016)
25. Shazeer, N., Doherty, R., Evans, C., Waterson, C.: Swivel: Improving embeddings by noticing what's missing. arXiv:1602.02215 (2016)
26. Teimouri, M., Rezakhah, S., Mohammdpour, A.: Robust mixture modelling using sub-Gaussian alpha-stable distribution. Tech. rep., arXiv:1701.06749 (2017)
27. Tian, F., Dai, H., Bian, J., Gao, B.: A probabilistic model for learning multi-prototype word embeddings. In: Proc. of the 25th International Conference on Computational Linguistics, pp. 151–160 (2014)
28. Titsias, M., Lázaro-Gredilla, M.: Doubly stochastic variational bayes for non-conjugate inference. In: Proc. of International Conference on Machine Learning (ICML) (2014)
29. Vilnis, L., McCallum, A.: Word representations via Gaussian embeddings. In: Proc. of International Conference on Learning Representations (2015)
30. Zhou, M., Carin, L.: Negative binomial process count and mixture modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(2), 307–320 (2015)
31. Zhou, M., Hannah, L., Dunson, D., Carin, L.: Beta-negative binomial process and Poisson factor analysis. Proc. of Artificial Intelligence and Statistics pp. 1462–1471 (2012)