The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Core Dependency Networks

Alejandro Molina
alejandro.molina@tu-dortmund.de
CS Department
TU Dortmund, Germany

Alexander Munteanu
alexander.munteanu@tu-dortmund.de
CS Department
TU Dortmund, Germany

Kristian Kersting
kersting@cs.tu-darmstadt.de
CS Department and
Centre for Cognitive Science
TU Darmstadt, Germany

Abstract

Many applications infer the structure of a probabilistic graphical model from data to elucidate the relationships between variables. But how can we train graphical models on a massive data set? In this paper, we show how to construct coresets—compressed data sets which can be used as proxy for the original data and have provably bounded worst case error—for Gaussian dependency networks (DNs), i.e., cyclic directed graphical models over Gaussians, where the parents of each variable are its Markov blanket. Specifically, we prove that Gaussian DNs admit coresets of size independent of the size of the data set. Unfortunately, this does not extend to DNs over members of the exponential family in general. As we will prove, Poisson DNs do not admit small coresets. Despite this worst-case result, we will provide an argument why our coreset construction for DNs can still work well in practice on count data. To corroborate our theoretical results, we empirically evaluated the resulting Core DNs on real data sets. The results demonstrate significant gains over no or naive sub-sampling, even in the case of count data.

Artificial intelligence and machine learning have achieved considerable successes in recent years, and an ever-growing number of disciplines rely on them. Data is now ubiquitous, and there is great value in understanding the data, e.g., building probabilistic graphical models to elucidate the relationships between variables. In the big data era, however, scalability has become crucial for any useful machine learning approach. In this paper, we consider the problem of training graphical models, in particular, Dependency Networks (Heckerman et al. 2000), on massive data sets. They are cyclic directed graphical models, where the parents of each variable are its Markov blanket and have been proven successful in various tasks, such as collaborative filtering (Heckerman et al. 2000), phylogenetic analysis (Carlson et al. 2008), genetic analysis (Dobra 2009; Phatak et al. 2010), network inference from sequencing data (Allen and Liu 2013), and traffic as well as topic modeling (Hadiji et al. 2015).

Specifically, we show that Dependency Networks over Gaussians—arguably one of the most prominent type of distribution in statistical machine learning—admit coresets of size independent of the size of the data set. Coresets are weighted subsets of the data, which guarantee that models fitting them will also provide a good fit for the original data set, and have been studied before for clustering (Badoiu, Har-Peled, and Indyk 2002; Feldman, Faulkner, and Krause 2011; Feldman, Schmidt, and Sohler 2013; Lucic, Bachem, and Krause 2016), classification (Har-Peled, Roth, and Zimak 2007; Har-Peled 2015; Reddi, Póczos, and Smola 2015), regression (Drineas, Mahoney, and Muthukrishnan 2006; 2008; Dasgupta et al. 2009; Geppert et al. 2017), and the smallest enclosing ball problem (Badoiu and Clarkson 2003; 2008; Feldman, Munteanu, and Sohler 2014; Agarwal and Sharathkumar 2015); we refer to (Phillips 2017) for a recent extensive literature overview. Our contribution continues this line of research and generalizes the use of coresets to probabilistic graphical modeling.

Unfortunately, this coreset result does not extend to Dependency Networks over members of the exponential family in general. We prove that Dependency Networks over Poisson random variables (Allen and Liu 2013; Hadiji et al. 2015) do not admit (sublinear size) coresets: every single input point is important for the model and needs to appear in the coreset.This is unfortunate when modeling count data—the primary target of Poisson distributions—which is at the center of many scientific endeavors such as citation counts, number of web page hits, counts of procedures in medicine, etc. Therefore, despite our worst-case result, we will provide an argument why our coreset construction for Dependency Networks can still work well in practice on count data. To corroborate our theoretical results, we empirically evaluated the resulting Core Dependency Networks (CDNs) on several real data sets and demonstrate significant gains over no or naive sub-sampling, even for count data.

We proceed as follows. We review Dependency Networks (DNs), prove that Gaussian DNs admit sublinear size coresets, and discuss the possibility to generalize this result to count data. Before concluding, we present empirical results.

Dependency Networks

Most of the existing AI and machine learning literature on graphical models is dedicated to binary, multinomial, or certain classes of continuous (e.g. Gaussian) random variables. Undirected models, aka *Markov Random Fields* (MRFs), such as Ising (binary random variables) and Potts (multinomial random variables) models have found a lot of applica-

tions in various fields such as robotics, computer vision, and statistical physics, among others. Whereas MRFs allow for cycles in the structures, directed models aka *Bayesian Networks* (BNs) required acyclic directed relationships among the random variables.

Dependency Networks (DNs)—the focus of the present paper—combine concepts from directed and undirected worlds and are due to Heckerman et al. (2000). Specifically, like BNs, DNs have directed arcs, but they allow for networks with cycles and bi-directional arcs, akin to MRFs. This makes DNs quite appealing for many applications because we can build multivariate models from univariate distributions (Allen and Liu 2013; Yang et al. 2015; Hadiji et al. 2015), while still permitting efficient structure learning using local estimators or gradient tree boosting. If the data are fully observed, learning is done locally on the level of the conditional probability distributions for each variable mixing directed and undirected as needed. Based on these local distributions, samples from the joint distribution are obtained via Gibbs sampling. Indeed, the Gibbs sampling neglects the question of a consistent joint probability distribution and instead makes only use of local distributions.

Formally, let $X = (X^{(1)}, \ldots, X^{(d)})$ denote a random vector and x its instantiation. A *Dependency Network* (DN) on X is a pair $(\mathcal{G}, \Psi)$ where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed, possibly cyclic, graph where each node in $\mathcal{V} = [d] = \{1, \ldots, d\}$ corresponds to the random variable $X^{(i)}$. In the set of directed edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \setminus \{(i,i) \,|\, i \in [d]\}$, each edge models a dependency between variables, i.e., if there is no edge between i and j then the variables $X^{(i)}$ and $X^{(j)}$ are conditionally independent given the other variables $X^{\setminus i,j}$ indexed by $[d] \setminus \{i,j\}$ in the network. We refer to the nodes that have an edge pointing to $X^{(i)}$ as its parents, denoted by $\mathbf{pa}_i = \{X^{(j)} \,|\, (j,i) \in \mathcal{E}\}$. $\Psi = \{p_i \,|\, i \in [d]\}$ is a set of conditional probability distributions associated with each variable $X^{(i)} \sim p_i$, where

$$p_i = p(x^{(i)} | \, \mathbf{pa}_i) = p(x^{(i)} | \, x^{\setminus i}) \; .$$

As example of such a local model, consider Poisson conditional probability distributions as illustrated in Fig. 1 (left):

$$p(x^{(i)} | \, \mathbf{pa}_i) = \frac{\lambda_i(x^{\setminus i})^{x^{(i)}}}{x^{(i)}!} e^{-\lambda_i(x^{\setminus i})} \; .$$

Here, $\lambda_i(x^{\setminus i})$ highlights the fact that the mean can have a functional form that is dependent on $X^{(i)}$'s parents. Often, we will refer to it simply as λ_i. The construction of the local conditional probability distribution is similar to the (multinomial) Bayesian network case. However, in the case of DNs, the graph is not necessarily acyclic and $p(x^{(i)} | \, x^{\setminus i})$ typically has an infinite range, and hence cannot be represented using a finite table of probability values. Finally, the full joint distribution is simply defined as the product of local distributions:

$$p(\mathbf{x}) = \prod\nolimits_{i \in [d]} p(x^{(i)} | \, x^{\setminus i}) \; ,$$

also called pseudo likelihood. For the Poisson case, this reads

$$p(\mathbf{x}) = \prod\nolimits_{i \in [d]} \frac{\lambda_i^{x^{(i)}}}{x^{(i)}!} e^{-\lambda_i} \; .$$

Note, however, that doing so does not guarantee the existence of a consistent joint distribution, i.e., a joint distribution of which they are the conditionals. Bengio et al. (2014), however, have recently proven the existence of a consistent distribution per given evidence, which does not have to be known in closed form, as long as an unordered Gibbs sampler converges.

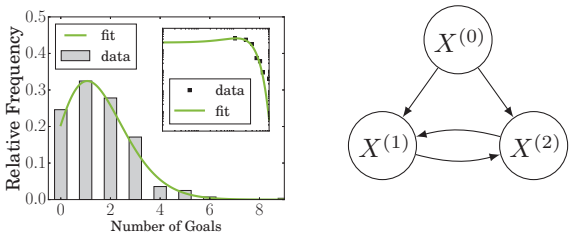

Figure 1: Illustration of Dependency Networks (DNs) using Poissons. **(left)** The number of goals scored in soccer games follows a Poisson distribution. The plot shows the distribution of home goals in the season 2012/13 of the German Bundesliga by the home team. The home team scored on average $\lambda = 1.59$ goals per game. **(right)** Example structure of a Poisson DN. The conditional distribution of each count variable given its neighbors is a Poisson distribution. Similar to a Bayesian network a Poisson DN is directed. However, it also contains cycles. (Best viewed in color)

Core Dependency Networks

As argued, learning Dependency Networks (DNs) amounts to determining the conditional probability distributions from a given set of n training instances $x_i \in \mathbb{R}^d$ representing the rows of the data matrix $X \in \mathbb{R}^{n \times d}$ over d variables. Assuming that $p(x^{(i)} | \, \mathbf{pa}_i)$ is parameterized as a generalized linear model (GLM) (McCullagh and Nelder 1989), this amounts to estimating the parameters $\gamma^{(i)}$ of the GLM associated with each variable $X^{(i)}$, since this completely determines the local distributions, but $p(x^{(i)} | \, \mathbf{pa}_i)$ will possibly depend on all other variables in the network, and these dependencies define the structure of the network. This view of training DNs as fitting d GLMs to the data allows us to develop *Core Dependency Networks* (CDNs): Sample a coreset and train a DN over certain members of the GLM family on the sampled coreset.

A *coreset* is a (possibly) weighted and usually considerably smaller subset of the input data that approximates a given objective function for all candidate solutions (Badoiu, Har-Peled, and Indyk 2002):

Definition 1 (ε-coreset)**.** *Let X be a set of points from a universe U and let Γ be a set of candidate solutions. Let $f : U \times \Gamma \to \mathbb{R}^{\geq 0}$ be a non-negative measurable function. Then a set $C \subset X$ is an ε-coreset of X for f, if*

$$\forall \gamma \in \Gamma : |f(X, \gamma) - f(C, \gamma)| \leq \varepsilon \cdot f(X, \gamma).$$

We now introduce the formal framework that we need towards the design of coresets for learning dependency networks. A very useful structural property for ℓ_2 based objective (or loss) functions is the concept of an ε-subspace embedding (Drineas, Mahoney, and Muthukrishnan 2006),2008.

Definition 2 (ε-subspace embedding)**.** *An ε-subspace embedding for the columnspace of X is a matrix S such that*

$$\forall \gamma \in \mathbb{R}^d : (1-\varepsilon)\|X\gamma\|^2 \leq \|SX\gamma\|^2 \leq (1+\varepsilon)\|X\gamma\|^2$$

We can construct a sampling matrix S which forms an ε-subspace embedding with constant probabilty in the following way: Let U be any orthonormal basis for the columnspace of X. This basis can be obtained from the singular value decomposition (SVD) $X = U\Sigma V^T$ of the data matrix. Now let $\rho = \operatorname{rank}(U) = \operatorname{rank}(X)$ and define the *leverage scores* $l_i = \|U_i\|^2/\|U\|_F^2 = \|U_i\|^2/\rho$ for $i \in [n]$. Now we fix a sampling size parameter $k = O(\rho \log(\rho/\varepsilon)/\varepsilon^2)$, sample the input points one-by-one with probability $q_i = \min\{1, k \cdot l_i\}$ and reweight their contribution to the loss function by $w_i = 1/q_i$. Note that, for the sum of squares loss, this corresponds to defining a diagonal (sampling) matrix S by $S_{ii} = 1/\sqrt{q_i}$ with probability q_i and $S_{ii} = 0$ otherwise. Also note, that the expected number of samples is $k = O(\rho \log(\rho/\varepsilon)/\varepsilon^2)$, which also holds with constant probability by Markov's inequality. Moreover, to give an intuition why this works, note that for any fixed $\gamma \in \mathbb{R}^d$, we have

$$\mathbb{E}\left[\|SX\gamma\|^2\right] = \sum \left(\frac{x_i\gamma}{\sqrt{q_i}}\right)^2 q_i = \sum (x_i\gamma)^2 = \|X\gamma\|^2.$$

The significantly stronger property of forming an ε-subspace embedding, according to Definition 2, follows from a matrix approximation bound given in (Rudelson and Vershynin 2007; Drineas, Mahoney, and Muthukrishnan 2008).

Lemma 3. *Let X be an input matrix with $\operatorname{rank}(X) = \rho$. Let S be a sampling matrix constructed as stated above with sampling size parameter $k = O(\rho \log(\rho/\varepsilon)/\varepsilon^2)$. Then S forms an ε-subspace embedding for the columnspace of X with constant probability.*

Proof. Let $X = U\Sigma V^T$ be the SVD of X. By Theorem 7 in (Drineas, Mahoney, and Muthukrishnan 2008) there exists an absolute constant $C > 1$ such that

$$\begin{aligned}\mathbb{E}\left[\|U^TS^TSU - U^TU\|\right] &\leq C\sqrt{\frac{\log k}{k}}\|U\|_F\|U\| \\ &\leq C\sqrt{\frac{\log k}{k}}\sqrt{\rho} \;\leq\; \varepsilon,\end{aligned}$$

where we used the fact that $\|U\|_F = \sqrt{\rho}$ and $\|U\| = 1$ by orthonormality of U. The last inequality holds by choice of $k = D\rho\log(\rho/\varepsilon)/\varepsilon^2$ for a large enough absolute constant $D > 1$ such that $\frac{1+\log D}{D} < \frac{1}{4C^2}$, since

$$\frac{\log k}{k} = \frac{\log(D\rho\log(\rho/\varepsilon)/\varepsilon^2)}{D\rho\log(\rho/\varepsilon)/\varepsilon^2} \leq \frac{2\varepsilon^2\log(D\rho\log(\rho/\varepsilon)/\varepsilon)}{D\rho\log(\rho/\varepsilon)}$$
$$\leq \frac{4\varepsilon^2(\log(\rho/\varepsilon)+\log D)}{D\rho\log(\rho/\varepsilon)} \leq \frac{4\varepsilon^2}{\rho}\left(\frac{1+\log D}{D}\right) < \frac{\varepsilon^2}{C^2\rho}\,.$$

By an application of Markov's inequality and rescaling ε, we can assume with constant probability

$$\|U^TS^TSU - U^TU\| \leq \varepsilon. \tag{1}$$

We show that this implies the ε-subspace embedding property. To this end, fix $\gamma \in \mathbb{R}^d$.

$$\begin{aligned}&|\,\|SX\gamma\|^2 - \|X\gamma\|^2\,| \\ =\;& \|\gamma^TX^TS^TSX\gamma - \gamma^TX^TX\gamma\| \\ =\;& \|\gamma^TV\Sigma U^TS^TSU\Sigma V^T\gamma - \gamma^TV\Sigma U^TU\Sigma V^T\gamma\| \\ =\;& \|\gamma^TV\Sigma\left(U^TS^TSU - U^TU\right)\Sigma V^T\gamma\| \\ \leq\;& \|U^TS^TSU - U^TU\| \cdot \|\Sigma V^T\gamma\|^2 \\ \leq\;& \|U^TS^TSU - U^TU\| \cdot \|X\gamma\|^2 \;\leq\; \varepsilon\|X\gamma\|^2,\end{aligned}$$

The first inequality follows by submultiplicativity, and the second from rotational invariance of the spectral norm. Finally we conclude the proof by Inequality (1). □

The question arises whether we can do better than $O(\rho\log(\rho/\varepsilon)/\varepsilon^2)$. One can show by reduction from the coupon collectors theorem that there is a lower bound of $\Omega(\rho\log\rho)$ matching the upper bound up to its dependency on ε. The hard instance is a $d^m \times d, m \in \mathbb{N}$ orthonormal matrix in which the scaled canonical basis $\mathbb{I}_d/\sqrt{d^{m-1}}$ is stacked d^{m-1} times. The leverage scores are all equal to $1/d^m$, implying a uniform sampling distribution with probability $1/d$ for each basis vector. Any rank $\rho = d$ preserving sample must comprise at least one of them. This is exactly the coupon collectors theorem with d coupons which has a lower bound of $\Omega(d\log d)$ (Motwani and Raghavan 1995). The fact that the sampling is without replacement does not change this since the reduction holds for arbitrarily large m creating sufficient multiple copies of each element to simulate the sampling with replacement (Tropp 2011).

Now we know that with constant probability over the randomness of the construction algorithm, S satisfies the ε-subspace embedding property for a given input matrix X. This is the crucial structural property to show that actually SX is a coreset for Gaussian linear regression models and dependency networks. Consider $(\mathcal{G}, \Psi)$, a Gaussian dependency network (GDN), i.e., a collection of Gaussian linear regression models

$$\Psi = \{p_i(X^{(i)}|X^{\setminus i}, \gamma^{(i)}) = \mathcal{N}(X^{\setminus i}\gamma^{(i)}, \sigma^2) \mid i \in [d]\}$$

on an arbitrary digraph structure $\mathcal{G}$ (Heckerman et al. 2000). The logarithm of the *(pseudo-)likelihood* (Besag 1975) of the above model is given by

$$\ln \mathcal{L}(\Psi) = \ln \prod p_i = \sum \ln p_i.$$

A maximum likelihood estimate can be obtained by maximizing this function with respect to $\gamma = (\gamma^{(1)}, \ldots, \gamma^{(d)})$ which is equivalent to minimizing the GDN loss function

$$f_G(X, \gamma) = \sum \|X^{\setminus i}\gamma^{(i)} - X^{(i)}\|^2.$$

Theorem 4. *Given S, an ε-subspace embedding for the columnspace of X as constructed above, SX is an ε-coreset of X for the GDN loss function.*

Proof. Fix an arbitrary $\gamma = (\gamma^{(1)}, \ldots, \gamma^{(d)}) \in \mathbb{R}^{d(d-1)}$. Consider the affine map $\Phi : \mathbb{R}^{d-1} \times [d] \to \mathbb{R}^d$, defined by $\Phi(\gamma^{(i)}) = \mathbb{I}_d^{\setminus i}\gamma^{(i)} - e_i$. Clearly Φ extends its argument from $d-1$ to d dimensions by inserting a -1 entry at position i and leaving the other entries in their original order. Let $\beta^{(i)} = \Phi(\gamma^{(i)}) \in \mathbb{R}^d$. Note that for each $i \in [d]$ we have

$$X\beta^{(i)} = X\Phi(\gamma^{(i)}) = X^{\setminus i}\gamma^{(i)} - X^{(i)}, \tag{2}$$

and each $\beta^{(i)}$ is a vector in $\mathbb{R}^d$. Thus, the triangle inequality and the universal quantifier in Definition 2 guarantee that

$$\begin{aligned} & |\sum \|SX\beta^{(i)}\|^2 - \sum \|X\beta^{(i)}\|^2| \\ = \; & |\sum (\|SX\beta^{(i)}\|^2 - \|X\beta^{(i)}\|^2)| \\ \leq \; & \sum |\|SX\beta^{(i)}\|^2 - \|X\beta^{(i)}\|^2| \\ \leq \; & \sum \varepsilon \|X\beta^{(i)}\|^2 \;=\; \varepsilon \sum \|X\beta^{(i)}\|^2. \end{aligned}$$

The claim follows by substituting Identity (2). □

It is noteworthy that computing one single coreset for the columnspace of X is sufficient, rather than computing d coresets for the d different subspaces spanned by $X^{\setminus i}$.

From Theorem 4 it is straightforward to show that the minimizer found for the coreset is a good approximation of the minimizer for the original data.

Corollary 5. *Given an ε-coreset C of X for the GDN loss function, let $\tilde{\gamma} \in \operatorname{argmin}_{\gamma \in \mathbb{R}^{d(d-1)}} f_G(C, \gamma)$. Then it holds that*

$$f_G(X, \tilde{\gamma}) \leq (1 + 4\varepsilon) \min_{\gamma \in \mathbb{R}^{d(d-1)}} f_G(X, \gamma).$$

Proof. Let $\gamma^* \in \operatorname{argmin}_{\gamma \in \mathbb{R}^{d(d-1)}} f_G(X, \gamma)$. Then

$$\begin{aligned} f_G(X, \tilde{\gamma}) \;&\leq\; \frac{1}{1-\varepsilon} f_G(C, \tilde{\gamma}) \;\leq\; \frac{1}{1-\varepsilon} f_G(C, \gamma^*) \\ &\leq\; \frac{1+\varepsilon}{1-\varepsilon} f_G(X, \gamma^*) \;\leq\; (1+4\varepsilon) f_G(X, \gamma^*). \end{aligned}$$

The first and third inequalities are direct applications of the coreset property, the second holds by optimality of $\tilde{\gamma}$ for the coreset, and the last follows from $\varepsilon < \frac{1}{2}$. □

Moreover, the coreset does not affect inference within GDNs. Recently, it was shown for (Bayesian) Gaussian linear regression models that the entire multivariate normal distribution over the parameter space is approximately preserved by ε-subspace embeddings (Geppert et al. 2017), which generalizes the above. This implies that the coreset yields a useful pointwise approximation in Markov Chain Monte Carlo inference via random walks like the pseudo-Gibbs sampler in (Heckerman et al. 2000).

Negative Result on Coresets for Poisson DNs

Unfortunately, there is no (sublinear size) coreset for the simpler problem of Poisson regression, which implies the result for Poisson DNs. We show this formally by reduction from the communication complexity problem known as *indexing*.

Recall that the negative log-likelihood for Poisson regression is (McCullagh and Nelder 1989; Winkelmann 2008)

$$\ell(\gamma) := \ell(\gamma | X, Y) = \sum \exp(x_i\gamma) - y_i \cdot x_i\gamma + \ln(y_i!).$$

Theorem 6. *Let Σ_D be a data structure for $D = [X, Y]$ that approximates likelihood queries $\Sigma_D(\gamma)$ for Poisson regression, such that*

$$\forall \gamma \in \mathbb{R}^d : \eta^{-1} \cdot \ell(\gamma|D) \leq \Sigma_D(\gamma) \leq \eta \cdot \ell(\gamma|D).$$

If $\eta < \frac{\exp(\frac{n}{4})}{2n^2}$ then Σ_D requires $\Omega(n)$ bits of storage.

Proof. We reduce from the indexing problem which is known to have $\Omega(n)$ one-way randomized communication complexity (Jayram, Kumar, and Sivakumar 2008). Alice is given a vector $b \in \{0,1\}^n$. She produces for every i with $b_i = 1$ the points $x_i = (r \cdot \omega^i, -1) \in \mathbb{R}^3$, where $\omega^i, i \in \{0, \ldots, n-1\}$ denote the n^{th} unit roots in the plane, i.e., the vertices of a regular n-polygon of radius $r = n/(1 - \cos(\frac{2\pi}{n})) \leq n^3$ in canonical order. The corresponding counts are set to $y_i = 1$. She builds and sends Σ_D of size $s(n)$ to Bob, whose task is to guess the bit b_j. He chooses to query $\gamma = (\omega^j, r \cdot \cos(\frac{2\pi}{n})) \in \mathbb{R}^3$. Note that this affine hyperplane separates $r \cdot \omega^j$ from the other scaled unit roots since it passes exactly through $r \cdot \omega^{(j-1) \bmod n}$ and $r \cdot \omega^{(j+1) \bmod n}$. Also, all points are within distance $2r$ from each other by construction and consequently from the hyperplane. Thus, $-2r \leq x_i\gamma \leq 0$ for all $i \neq j$.

If $b_j = 0$, then x_j does not exist and the cost is at most

$$\begin{aligned} \ell(\gamma) \;&=\; \sum \exp(x_i\gamma) - y_i \cdot x_i\gamma + \ln(y_i!) \\ &\leq\; \sum 1 + 2r + 1 \leq 2n + 2nr \leq 4n^4 \,. \end{aligned}$$

If $b_j = 1$ then x_j is in the expensive halfspace and at distance exactly

$$\begin{aligned} x_j\gamma \;&=\; (r\omega^j)^T\omega^j - r \cdot \cos\left(\frac{2\pi}{n}\right) \\ &=\; r \cdot \left(1 - \cos\left(\frac{2\pi}{n}\right)\right) \;=\; n \end{aligned}$$

So the cost is bounded below by $\ell(\gamma) \geq \exp(n) - n + 1 \geq \exp(\frac{n}{2})$.

Given $\eta < \frac{\exp(\frac{n}{4})}{2n^2}$, Bob can distinguish these two cases based on the data structure only, by deciding whether $\Sigma_D(\gamma)$ is strictly smaller or larger than $\exp(\frac{n}{4}) \cdot 2n^2$. Consequently $s(n) = \Omega(n)$, since this solves the indexing problem. □

Note that the bound is given in bit complexity, but restricting the data structure to a sampling-based coreset and assuming every data point can be expressed in $O(d \log n)$ bits, this means we still have a lower bound of $k = \Omega(\frac{n}{\log n})$ samples.

Corollary 7. *Every sampling based coreset for Poisson regression with approximation factor $\eta < \frac{\exp(\frac{n}{4})}{2n^2}$ as in Theorem 6 requires at least $k = \Omega(\frac{n}{\log n})$ samples.*

At this point, it seems very likely that a similar argument can be used to rule out any $o(n)$-space constant approximation algorithm. This remains an open problem for now.

Why Core DNs for Count Data can still work

In the Gaussian setting, the loss is measured in squared Euclidean distance and the number of important points, i.e., having significantly large leverage scores, is bounded essentially by $O(d)$. This is implicit in the original early works (Drineas, Mahoney, and Muthukrishnan 2008) and has been explicitly formalized later (Langberg and Schulman 2010; Clarkson and Woodruff 2013). It is crucial to understand that this is an inherent property of the norm function, and thus holds for arbitrary data. For the Poisson GLM, in contrast, we have shown that its loss function does not come with such properties from scratch. We constructed a worst case scenario, where basically every single input point is important for the model and needs to appear in the coreset. Usually, this is not the case with statistical models, where the data is assumed to be generated i.i.d. from some generating distribution that fits the model assumptions. Consider for instance a data reduction for Gaussian linear regression via leverage score sampling vs. uniform sampling. It was shown that the leverage scores are quite uniform. In the presence of more and more outliers generated by the heavier tails of t-distributions, the leverage scores increasingly outperform uniform sampling (Ma, Mahoney, and Yu 2015).

The Poisson model

$$y_i \sim \mathrm{Poi}(\lambda_i),\ \lambda_i = \exp(x_i\gamma). \tag{3}$$

suffers from its inherent limitation on equidispersed data since $\mathbb{E}[y_i|x_i] = \mathbb{V}[y_i|x_i] = \exp(x_i\gamma)$. Count data, however, is often overdispersed especially for large counts. This is due to unobserved variables or problem specific heterogeneity and contagion-effects. The log-normal Poisson model is known to be inferior for data which specifically follows the Poisson model, but turns out to be more powerful in modeling the effects that can not be captured by the simple Poisson model. We review the log-normal Poisson model for count data (Winkelmann 2008)

$$\begin{aligned} y_i &\sim \mathrm{Poi}(\lambda_i), \\ \lambda_i &= \exp(x_i\gamma)u_i = \exp(x_i\gamma + v_i), \\ v_i = \ln u_i &\sim \mathcal{N}(\mu, \sigma). \end{aligned}$$

A natural choice for the parameters of the log-normal distribution is $\mu = -\frac{\sigma^2}{2}$ in which case we have

$$\begin{aligned} \mathbb{E}[y_i|x_i] &= \exp(x_i\gamma + \mu + \sigma^2/2) = \exp(x_i\gamma), \\ \mathbb{V}[y_i|x_i] &= \mathbb{E}[y_i|x_i] + (\exp(\sigma^2) - 1)\mathbb{E}[y_i|x_i]^2. \end{aligned}$$

It follows that $\mathbb{V}[y_i|x_i] = \exp(x_i\gamma) + \Omega(\exp(x_i\gamma)^2) > \exp(x_i\gamma)$, where a constant σ^2 that is independent of x_i, controls the amount of overdispersion. Taking the limit for $\sigma \to 0$ we arrive at the simple model (3), since the distribution of $v_i = \ln u_i$ tends to δ_0, the deterministic Dirac delta distribution which puts all mass on 0. The inference might aim for the log-normal Poisson model directly as in (Zhou et al. 2012), or it can be performed by (pseudo-)maximum likelihood estimation of the simple Poisson model. The latter provides a consistent estimator as long as the log-linear mean function is correctly specified, even if higher moments do not possess the limitations inherent in the simple Poisson model (Winkelmann 2008).

Preserving the log-linear mean function in a Poisson model is crucial towards consistency of the estimator. Moreover, modeling counts in a log-normal model gives us intuition why leverage score sampling can capture the underlying linear model accurately: In the log-normal Poisson model, u follows a log-normal distribution. It thus holds for $\ln\lambda = X\gamma + \ln u = X\gamma + v$, that

$$v \sim \mathcal{N}\left(-\frac{\sigma^2}{2}\cdot\mathbb{1}, \sigma^2\mathbb{I}_n\right)$$

by independence of the observations, which implies

$$\ln\lambda \sim \mathcal{N}\left(X\gamma - \frac{\sigma^2}{2}\cdot\mathbb{1}, \sigma^2\mathbb{I}_n\right).$$

Omitting the bias $\mu = -\frac{\sigma^2}{2}$ in each intercept term (which can be cast into X), we notice that this yields again an ordinary least squares problem $\|X\gamma - \ln(\lambda)\|^2$ defined in the column space of X.

There is still a missing piece in our argumentation. In the previous section, we have used that the coreset construction is an ε-subspace embedding for the columnspace of the whole data set including the dependent variable, i.e., for $[X, \ln(\lambda)]$. We face two problems. First, λ is only implicitly given in the data, but is not explicitly available. Second, λ is a vector derived from $X^{\setminus i}$ in our setting and might be different for any of the d instances. Fortunately, it was shown via more complicated arguments (Drineas, Mahoney, and Muthukrishnan 2008), that it is sufficient for a good approximation if the sampling is done obliviously to the dependent variable. The intuition comes from the fact that the loss of any point in the subspace can be expressed via the projection of $\ln(\lambda)$ onto the subspace spanned by X, and the residual of its projection. A good approximation of the subspace implicitly approximates the projection of any fixed vector, which is then applied to the residual vector of the orthogonal projection. This solves the first problem since it is only necessary to have a subspace embedding for X. The second issue can be addressed by increasing the sample size by a factor of $O(\log d)$ for boosting the error probability to $O(1/d)$ and taking a union bound.

Empirical Illustration

Our intention here is to corroborate our theoretical results by investigating the following questions empirically:

(Q1) How does the performance of CDNs compare to DNs with access to the full training data set and to a uniform sample from the training data set? And, how does the empirical error behave according to the sample sizes?

(Q2) Do coresets affect the structure recovered by the DN?

To this aim, we implemented (C)DNs in Python[1]. All experiments ran on a Linux machine (56 cores, 4 GPUs, and 512GB RAM). All DNs were trained using Iteratively reweighted least squares (IRWLS), however, coresets do not depend on the learning algorithm used.

[1] https://github.com/alejandromolinaml/CoreDNs

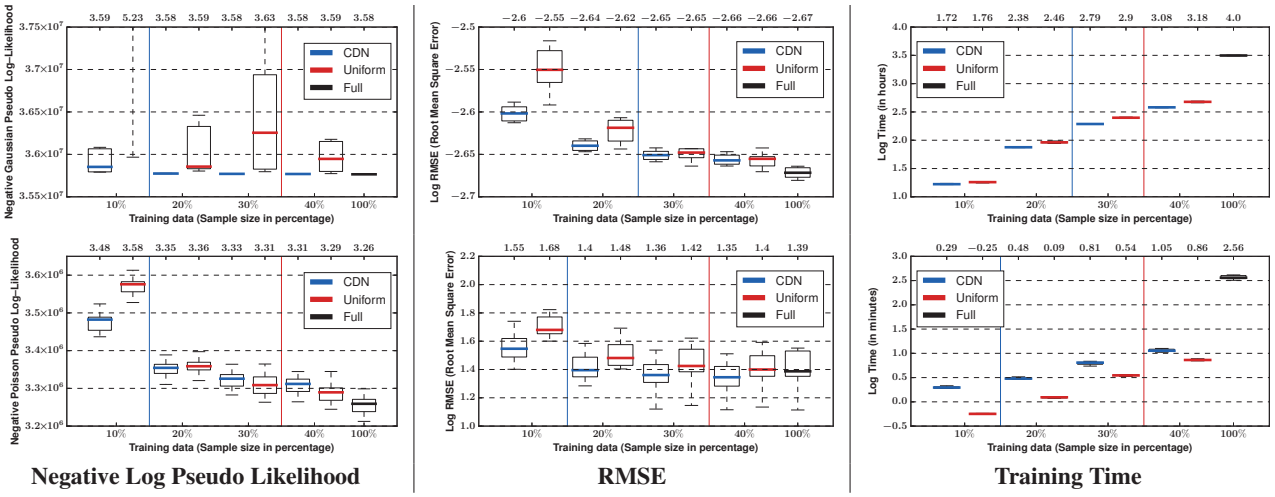

Figure 2: **(Q1)** Performance (the lower, the better) of Gaussian CDNs on MNIST (upper row) and Poisson CNDs on the traffic dataset (lower row) 10-fold cross-validated. Shown are the negative log pseudo-likelihood (left), the squared error loss (middle, in log-space) as well as the training time (right, in log-space) on the y-axis for different proportions of the data sampled (x axis). Please note the jump in the x-axis after 40%. As one can see, CDNs (blue) quickly approach the predictive performance of the full dataset (Full, black). Uniform sampling (Uniform, red) does not perform as well as CDNs. Moreover, CDNs can be orders of magnitude faster than DNs on the full dataset and scale similar to uniform sampling. This is also supported by the vertical lines. They denote the mean performances (the more to the left, the better) on the top axes. (Best viewed in color)

Benchmarks on MNIST and Traffic Data (Q1): We considered two datasets. We used the MNIST[2] data set of handwritten labeled digits. We employed the training set consisting of 55000 images, each with 784 pixels, for a total of 43,120,000 measurements, and trained Gaussian DNs on it. The second dataset contains traffic count measurements on selected roads around the city of Cologne in Germany (Ide et al. 2015). It consists of 7994 time-stamped measurements taken by 184 sensors for a total of 1,470,896 measurements, and we trained Poisson DNs on it. For each dataset, we performed ten fold cross-validation for training a full DN (Full) using all the data, leverage score sampling coresets (CDNs), and uniform samples (Uniform), for different sample sizes. We then compared the predictions made by all the DNs and the time taken to train them. Although the traffic dataset is easy to approximate by larger uniform sampling; due to the regularities in daily traffic patterns (commuting people cause peaks in the morning and evening, little traffic at night, more traffic at daytime). The challenging task is to be good at small sample sizes, where CPDNs are superior. It can also be seen that CPDNs are better in predictive performance (RMSE). For the predictions on the MNIST dataset, we clipped the values to the range [0,1] for all the DNs. For the Traffic dataset, we computed the predictions $\lfloor x \rfloor$ of every measurement x rounded to the largest integer less than or equal to x.

Fig. 2 summarizes the results. As one can see, CDNs outperform DNs trained on full data and are orders of magnitude faster. Compared to uniform sampling, coresets are competitive. As seen on the traffic dataset, CDNs can have more predictive power than the "optimal" model using the full data. This is in line with Mahoney (2011), who observed that coresets implicitly introduce regularization and lead to more robust output. Table 1 summarizes the empirical relative errors $|f(X, \tilde{\gamma}) - f(X, \gamma^*)|/f(X, \gamma^*)$ between (C/U)DNs $\tilde{\gamma}$ and DNs γ^* trained on all the data. CDNs clearly recover the original model, at a fraction of training data. Overall, this answers **(Q1)** affirmatively.

Sample	MNIST		Traffic	
portion	GCDN	GUDN	PCDN	PUDN
10%	**18.03%**	11162.01%	**6.81%**	9.6%
20%	**0.57%**	13.86%	**2.9%**	3.17%
30%	**0.01%**	13.33%	2.04%	**1.68%**
40%	**0.01%**	2.3%	1.59%	**0.99%**

Table 1: **(Q1)** Comparison of the empirical relative error (the lower, the better). Best results per dataset are bold. Both Gaussian (GCDNs) and Poisson (PCDNs) CDNs recover the model well, with a fraction of the training data. Uniformly sampled DNs (UDNs) lag behind as the sample size drops.

Relationship Elucidation (Q2): We investigated the performance of CDNs when recovering the graph structure of word interactions from a text corpus. For this purpose, we used the NIPS[3] bag-of-words dataset. It contains 1,500 documents with a vocabulary above 12k words. We considered the 100 most frequent words. Fig. 3 illustrates the results qualitatively. It shows three CDNs of sampling sizes 40%, 70% and 100% for Gaussians (top) after a $\log(x + 1)$ trans-

[2]http://yann.lecun.com/exdb/mnist/

[3]https://archive.ics.uci.edu/ml/datasets/bag+of+words

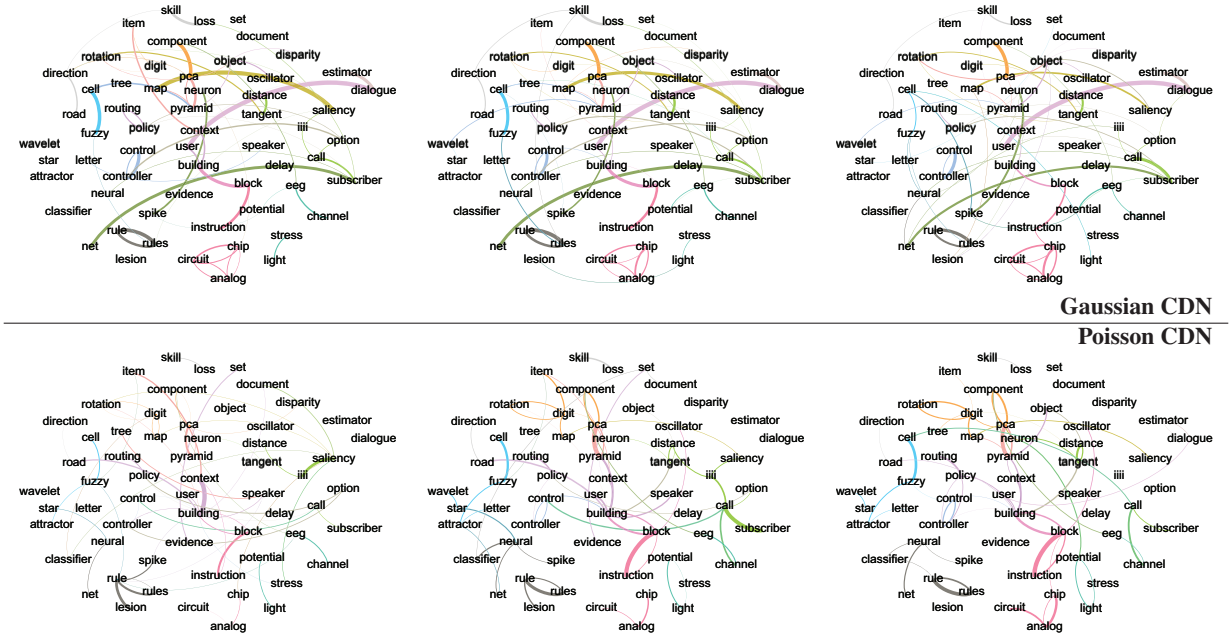

Figure 3: **(Q2)** Elucidating the relationships between random variables. Shown are the (positive) dependency structures of Gaussian (top) and Poisson (bottom) CDNs on NIPS and different learning sampling sizes: using 40% (Left) , 70% (Middle) and 100% (Right). The edges show the 70 top thresholded positive coefficients of the GLMs. The colors of the edges represent modularity. As one can see, CDNs elucidate relationships among the words that make semantical sense and approach the structure learned using the full dataset. For a quantitative assessment, see Tab. 2. (Best viewed in color)

formation and for Poissons (bottom): CDNs capture well the gist of the NIPS corpus. Table 2 confirms this quantitatively. It shows the Frobenius norms between the DNs: CDNs capture the gist better than naive, i.e., uniform sampling. This answers **(Q2)** affirmatively.

To summarize our empirical results, the answers to questions **(Q1)** and **(Q2)** show the benefits of CDNs.

Conclusions

Inspired by the question of how we can train graphical models on massive datasets, we have studied coresets for estimating Dependency networks (DNs). We present the first rigorous guarantees for obtaining compressed ε-approximations of Gaussian DNs for large data sets. We proved worst-case impossibility results on coresets for Poisson DNs. A review of log-normal Poisson modeling of counts provided deep insights into why our coreset construction still performs well for count data in practice. Our experimental results demonstrate the resulting Core Dependency Networks (CDNs) can achieve significant gains over no or naive sub-sampling, even in the case of count data, making it possible to learn models on much larger datasets using the same hardware.

CDNs provide several interesting avenues for future work. The conditional independence assumption opens the door to explore hybrid multivariate models, where each variable can potentially come from a different GLM family or link function, on massive data sets. This can further be used to hint at independencies among variables in the multivariate setting, making them useful in other large data applications.

Sample	UDN		CDN	
portion	Gaussian	Poisson	Gaussian	Poisson
40%	9.0676	6.4042	**3.9135**	**0.6497**
70%	4.8487	1.6262	**2.6327**	**0.3821**

Table 2: **(Q2)** Frobenius norm of the difference of the adjacency matrices (the lower, the better) recovered by DNs trained on the full data and trained on a uniform subsample (UDN) resp. coresets (CDNs) of the training data. The best results per statistical type (Gaussian/Poisson) are bold. CDNs recover the structure better than UDNs.

Our results may pave the way to establish coresets for deep models using the close connection between dependency networks and deep generative stochastic networks (Bengio et al. 2014), sum-product networks (Poon and Domingos 2011; Molina, Natarajan, and Kersting 2017), as well as other statistical models that build multivariate distributions from univariate ones (Yang et al. 2015).

Acknowledgements: The authors would like to thank the anonymous reviewers for their feedback and acknowledge the support by the German Science Foundation (DFG) Collaborative Research Center SFB 876 Providing Information by Resource-Constrained Analysis, projects B4 and C4. KK acknowledges the support by the Centre for Cognitive Science at the TU Darmstadt.

References

Agarwal, P. K., and Sharathkumar, R. 2015. Streaming algorithms for extent problems in high dimensions. *Algorithmica* 72(1):83–98.

Allen, G. I., and Liu, Z. 2013. A local poisson graphical model for inferring networks from sequencing data. *IEEE Transactions on Nanobioscience* 12(3):189–198.

Badoiu, M., and Clarkson, K. L. 2003. Smaller core-sets for balls. In *Proc. of SODA*, 801–802.

Badoiu, M., and Clarkson, K. L. 2008. Optimal core-sets for balls. *Computational Geometry* 40(1):14–22.

Badoiu, M.; Har-Peled, S.; and Indyk, P. 2002. Approximate clustering via core-sets. In *Proceedings of STOC*, 250–257.

Bengio, Y.; Laufer, E.; Alain, G.; and Yosinski, J. 2014. Deep generative stochastic networks trainable by backprop. In *Proc. of ICML*, 226–234.

Besag, J. 1975. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society, Series D* 24(3):179–195.

Carlson, J. M.; Brumme, Z. L.; Rousseau, C. M.; Brumme, C. J.; Matthews, P.; Kadie, C. M.; Mullins, J. I.; Walker, B. D.; Harrigan, P. R.; Goulder, P. J. R.; and Heckerman, D. 2008. Phylogenetic dependency networks: Inferring patterns of CTL escape and codon covariation in HIV-1 gag. *PLoS Computational Biology* 4(11).

Clarkson, K. L., and Woodruff, D. P. 2013. Low rank approximation and regression in input sparsity time. In *Proc. of STOC*, 81–90.

Dasgupta, A.; Drineas, P.; Harb, B.; Kumar, R.; and Mahoney, M. W. 2009. Sampling algorithms and coresets for ℓ_p regression. *SIAM Journal on Computing* 38(5):2060–2078.

Dobra, A. 2009. Variable selection and dependency networks for genomewide data. *Biostatistics* 10(4):621–639.

Drineas, P.; Mahoney, M. W.; and Muthukrishnan, S. 2006. Sampling algorithms for ℓ_2 regression and applications. In *Proc. of SODA*, 1127–1136.

Drineas, P.; Mahoney, M. W.; and Muthukrishnan, S. 2008. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications* 30(2):844–881.

Feldman, D.; Faulkner, M.; and Krause, A. 2011. Scalable training of mixture models via coresets. In *Proc. of NIPS*.

Feldman, D.; Munteanu, A.; and Sohler, C. 2014. Smallest enclosing ball for probabilistic data. In *Proc. of SOCG*, 214–223.

Feldman, D.; Schmidt, M.; and Sohler, C. 2013. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In *Proc. of SODA*, 1434–1453.

Geppert, L. N.; Ickstadt, K.; Munteanu, A.; Quedenfeld, J.; and Sohler, C. 2017. Random projections for Bayesian regression. *Statistics and Computing* 27(1):79–101.

Hadiji, F.; Molina, A.; Natarajan, S.; and Kersting, K. 2015. Poisson dependency networks: Gradient boosted models for multivariate count data. *MLJ* 100(2-3):477–507.

Har-Peled, S.; Roth, D.; and Zimak, D. 2007. Maximum margin coresets for active and noise tolerant learning. In *Proc. of IJCAI*, 836–841.

Har-Peled, S. 2015. A simple algorithm for maximum margin classification, revisited. *arXiv* 1507.01563.

Heckerman, D.; Chickering, D.; Meek, C.; Rounthwaite, R.; and Kadie, C. 2000. Dependency networks for density estimation, collaborative filtering, and data visualization. *Journal of Machine Learning Research* 1:49–76.

Ide, C.; Hadiji, F.; Habel, L.; Molina, A.; Zaksek, T.; Schreckenberg, M.; Kersting, K.; and Wietfeld, C. 2015. LTE connectivity and vehicular traffic prediction based on machine learning approaches. In *Proc. of IEEE VTC Fall*.

Jayram, T. S.; Kumar, R.; and Sivakumar, D. 2008. The one-way communication complexity of Hamming distance. *Theory of Computing* 4(1):129–135.

Langberg, M., and Schulman, L. J. 2010. Universal epsilon-approximators for integrals. In *Proc. of SODA*.

Lucic, M.; Bachem, O.; and Krause, A. 2016. Strong coresets for hard and soft bregman clustering with applications to exponential family mixtures. In *Proc. of AISTATS*, 1–9.

Ma, P.; Mahoney, M. W.; and Yu, B. 2015. A statistical perspective on algorithmic leveraging. *JMLR* 16:861–911.

Mahoney, M. W. 2011. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning* 3(2):123–224.

McCullagh, P., and Nelder, J. 1989. *Generalized Linear Models*. Chapman and Hall.

Molina, A.; Natarajan, S.; and Kersting, K. 2017. Poisson sum-product networks: A deep architecture for tractable multivariate poisson distributions. In *Proc. of AAAI*.

Motwani, R., and Raghavan, P. 1995. *Randomized Algorithms*. Cambridge Univ. Press.

Phatak, A.; Kiiveri, H. T.; Clemmensen, L. H.; and Wilson, W. J. 2010. NetRaVE: constructing dependency networks using sparse linear regression. *Bioinformatics* 26(12):1576–1577.

Phillips, J. M. 2017. Coresets and sketches. In *Handbook of Discrete and Computational Geometry*.

Poon, H., and Domingos, P. 2011. Sum-Product Networks: A New Deep Architecture. *Proc. of UAI*.

Reddi, S. J.; Póczos, B.; and Smola, A. J. 2015. Communication efficient coresets for empirical loss minimization. In *Proc. of UAI*, 752–761.

Rudelson, M., and Vershynin, R. 2007. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM* 54(4):21.

Tropp, J. A. 2011. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis* 3(1-2):115–126.

Winkelmann, R. 2008. *Econometric Analysis of Count Data*. Springer, 5th edition.

Yang, E.; Ravikumar, P.; Allen, G. I.; and Liu, Z. 2015. On graphical models via univariate exponential family distributions. *JMLR* 16:3813–3847.

Zhou, M.; Li, L.; Dunson, D. B.; and Carin, L. 2012. Lognormal and gamma mixed negative binomial regression. In *Proceedings of ICML*.