The von Mises Graphical Model: Regularized Structure and Parameter Learning

Narges Razavian, Hetunandan Kamisetty, Christopher James Langmead

September 2011
CMU-CS-11-129
CMU-CB-11-101

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The von Mises distribution is a continuous probability distribution on the circle used in directional statistics. In this paper, we introduce the undirected von Mises Graphical model and present an algorithm for parameter and structure learning using L_1 regularization. We show that the learning algorithm is both consistent and statistically efficient. Additionally, we introduce a simple inference algorithm based on Gibbs sampling. We compare the von Mises Graphical Model (vGM) with a Gaussian Graphical Model (GGM) on both synthetic data and on data from protein structures, and demonstrate that the vGM achieves higher accuracy than the GGM.

This research was supported by NSF IIS-0905193. Corresponding Author: cjl@cs.cmu.edu

Keywords: Structure Learning, Regularization, von Mises, Probabilistic Graphical Models, Proteins

1 Introduction

The von Mises distribution is used in directional statistics for modeling angles [4] and other circularly distributed continuous random variables. It closely approximates the wrapped normal distribution [2], but has the advantage of being more tractable, mathematically [10]. Unfortunately, dealing with a large number of inter-dependent von Mises variables can be very challenging. Thus, motivated by the desire to model the bond and dihedral angles that determine the three dimensional structure of proteins and other molecules, we introduce the first algorithm for learning the dependency structure and parameters for large multivariate von Mises probabilistic graphical Models. Previous learning algorithms for von Mises distributions have either been limited to bivariate models [6, 5, 1], or else assume that the dependency structure is known [16, 17].

The paper is organized as follows. In Section 2 we review the univariate and bivariate von Mises distributions. Then, we introduce the multivariate von Mises distribution as an undirected graphical model in Section 3. We then define a Gibbs sampler for the model in Section 4, which we use later for drawing samples and performing inference. Section 5 presents our structure learning algorithm which employs L_1 regularization. We also prove that our algorithm is consistent. In Section 6, we compare and contrast the von Mises graphical model (vGM) to another popular continuous graphical model, the Gaussian graphical model (GGM), over synthetic and real angle data. We show that the vGM achieves higher accuracy than the GGM when each variable has high marginal variance.

2 Background

The wrapped normal distribution for an anglau variable, $\theta \in (-\pi, \pi]$ is defined as an infinite sum of the wrappings of a normal distribution around the unit circle:

$$f_{WN}(\theta; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} e^{\frac{-(\theta-\mu+2\pi k)^2}{2\sigma^2}},$$

where μ and σ are the mean and standard deviation of the unwrapped distribution, respectively. The von Mises distribution, which is also known as the circular normal distribution, has a more compact representation given by:

$$f_{VM}(\theta; \mu, \kappa) = \frac{e^{\kappa \cos(\theta-\mu)}}{2\pi I_0(\kappa)}$$

where $I_0(\kappa)$ is the modified Bessel function of order 0, and the parameters μ and $\frac{1}{\kappa}$ are analogous to μ and σ^2 (the mean and variance) in the normal distribution. κ is known as the *concentration* of the variable, and so high concentration implies low variance.

Unlike the wrapped normal distribution, the von Mises distribution belongs to the exponential family and can be extended to higher dimension. The bivariate von Mises distribution [9] over $\Theta = (\theta_1, \theta_2)$, can be defined as:

$$f(\Theta) = \frac{\exp\left\{\left[\sum_{i=1}^{2} \kappa_i \cos(\theta_i - \mu_i)\right] + \vec{K}_1(\Theta, \mu)\mathbf{M}\vec{K}_2(\Theta, \mu)^T\right\}}{Z_c(\mu_1, \mu_2, \kappa_1, \kappa_2, \mathbf{M})},$$

where μ_1 and μ_2 are the means of θ_1 and θ_2, respectively, κ_1 and κ_2 are their corresponding concentrations, $\vec{K}_1(\Theta, \mu) = [\cos(\theta_1 - \mu_1), \sin(\theta_1 - \mu_1)]$, $\vec{K}_2(\Theta) = [\cos(\theta_2 - \mu_2), \sin(\theta_2 - \mu_2)]$, $\mathbf{M}$ is a 2×2 matrix corresponding to their correlation, and $Z_c(\cdot)$ is the normalization constant.

Another way to define the bivariate von Mises, which has been the definition of choice in all previous work[5] [1], is as follows:

$$f(\Theta) = \frac{\exp\left\{\left[\sum_{i=1}^{2} \kappa_i \cos(\theta_i - \mu_i)\right] + \lambda g(\theta_1, \theta_2)\right\}}{Z_s(\mu_1, \mu_2, \kappa_1, \kappa_2, \lambda)},$$

where μ_1, μ_2, κ_1, and κ_2 are as previously defined, $g(\theta_1, \theta_2) = \sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)$, and λ is a measure of the dependence between θ_1 and θ_2. This formulation, known as the *sine variant*, is generally preferred because it only requires five parameters and is easily expandable to more than 2 variables, as will be demonstrated in the next section.

3 The von Mises Graphical Model (vGM)

Let $\Theta = (\theta_1, \theta_2, ..., \theta_p)$, where $\theta_i \in (-\pi, \pi]$. The multivariate von Mises distribution [9] with parameters $\vec{\mu}, \vec{\kappa}$, and Λ is given by:

$$f(\Theta) = \frac{\exp\left\{\vec{\kappa}^T C(\vec{\Theta}, \mu) + \frac{1}{2} S(\vec{\Theta}, \mu)\Lambda S(\vec{\Theta}, \mu)^T\right\}}{Z(\vec{\mu}, \vec{\kappa}, \Lambda)},$$

where $\vec{\mu} = [\mu_1, \mu_2, \cdots, \mu_p]$, $\vec{\kappa} = [\kappa_1, \kappa_2, \cdots, \kappa_p]$, $C(\vec{\Theta}, \mu) = [\cos(\theta_1 - \mu_1), \cos(\theta_2 - \mu_2), \cdots, \cos(\theta_p - \mu_p)]$, $S(\vec{\Theta}, \mu) = [\sin(\theta_1 - \mu_1), \sin(\theta_2 - \mu_2), \cdots, \sin(\theta_p - \mu_p)]$, Λ is a $p \times p$ matrix such that $\Lambda_{ii} = 0$, and $\Lambda_{ij} = \lambda_{ij} = \lambda_{ji}$, and $Z(\vec{\mu}, \vec{\kappa}, \Lambda)$ is the normalization constant.

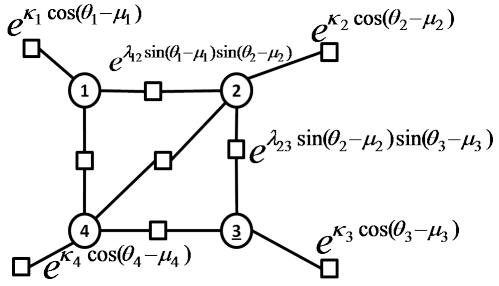


Figure 1: Factor Graph Representation for multivariate von Mises distribution. Each circular node is a variable, and the square nodes are factors.

It is known that the multivariate von Mises distribution can be closely approximated with a multivariate Gaussian distribution — provided that each of the variables has low variance (i.e.,

for large values of κ) [6]. This is significant because learning and inference can be performed analytically for multivariate Gaussian distributions. However, we will show in Section 6 that the Gaussian approximation introduces significant error when the variance is high (i.e., for small values of κ_i). We address this problem by encoding the multivariate von Mises distribution as a graphical model over von Mises-distributed random variables. Figure 1 shows the factor graph representation of the graphical mode for four variables. Under this representation the node factors are defined as $f_i = \kappa_i cos(\theta_i - \mu_i)$ and the edge factors are defined as $f_{ij} = \lambda_{ij} sin(\theta_i - \mu_i) sin(\theta_j - \mu_j)$. Like all factor graphs, the model encodes the joint distribution as the normalized product of all factors:

$$P(\Theta = \theta) = \frac{1}{Z} \prod_{a \in A} f_a(\theta_{ne(a)}),$$

where A is the set of factors and $\theta_{ne(a)}$ are the neighbors of f_a (factor a) in the factor graph.

4 Sampling for Inference

The evaluation of the joint von Mises distribution requires the calculations of the normalization constant, Z. Unfortunately, Z does not have a closed form solution in this case, and must therefore be calculated by inference. We have recently derived an Expectation-Propagation style algorithm for performing inference in the vGM [11]. In this paper, we will instead use a Gibbs sampler to perform approximate inference.

Gibbs sampling assumes that it is easy to sample from the univariate conditionals. Fortunately, as shown in [6] the univariate von Mises $conditionals$ are $univariate$ von Mises distributions themselves, and this makes Gibbs sampling a feasible option. In particular

$$f(\theta_p | \theta_1, \theta_2, \ldots \theta_{p-1}) \propto e^{\kappa_p cos(\theta_p - \mu_p) + \sum_{j=1}^{p-1} \lambda_{jp} sin(\theta_j - \mu_j) sin(\theta_p - \mu_p)} = e^{\kappa^* cos(\theta_p - \mu^*)}$$

where

$$\kappa^* = \sqrt{\kappa_p^2 + (\sum_{j=1}^{p-1} \lambda_{jp} sin(\theta_j - \mu_j))^2} \tag{1}$$

$$\mu^* = \mu_p + \arctan(\frac{1}{\kappa_p} \sum_{j=1}^{p-1} \lambda_{jp} sin(\theta_j - \mu_j)) \tag{2}$$

This univariate conditional is sufficient for implementing a Gibbs sampler to generate samples from the vGM and perform inference.

5 Structure and Parameter Learning

We next consider the problem of learning the parameters of the model from data. Let $(\vec{\mu}, \vec{\kappa}, \Lambda)$ be the parameters of the vGM, as defined in Section 3. Given a set of $i.i.d.$ training samples,

$D = \{\Theta_1, \Theta_2, \cdots \Theta_n\}$, the likelihood function is:

$$\mathcal{L}(D|\vec{\mu}, \vec{\kappa}, \Lambda) = \prod_{i=1}^{n} \frac{e^{\vec{\kappa}\vec{C}_i(\Theta,\vec{\mu}) + \frac{1}{2}\vec{S}_i(\Theta,\vec{\mu})^T \Lambda \vec{S}_i(\Theta,\vec{\mu})}}{Z_p(\vec{\mu}, \vec{\kappa}, \Lambda)}$$

where $\vec{C}(\Theta_i, \vec{\mu}) = [\cos(\theta_{i1} - \mu_1), \cdots \cos(\theta_{in} - \mu_p)]$, and $\vec{S}(\Theta_i, \vec{\mu}) = [\sin(\theta_{i1} - \mu_1), \cdots \sin(\theta_{in} - \mu_p)]$.

In theory, a maximum likelihood estimate MLE for the parameters can be obtained by maximizing the likelihood of the data. Unfortunately, computing the normalization for the vGM is intractable and so computing the MLE estimate is as well. We will therefore maximize the full $pseudo$-likelihood instead.

5.1 Full pseudo-likelihood for von Mises Graphical Model

The full pseudo likelihood for the multivariate von Mises is defined as:

$$\mathcal{PL}(\Theta|\vec{\mu}, \vec{\kappa}, \Lambda) = (2\pi)^{-pn} \prod_{i=1}^{n} \prod_{j=1}^{p} P_{vm}(\theta_{i,j}|\theta_{i,1}, ..., \theta_{i,j-1}, \theta_{i,j+1}...\theta_{i,p})$$

As discussed in section 4, each univariate conditional term for the vGM is itself a univariate von Mises distribution. Thus, the full pseudo likelihood can be re-written as:

$$\mathcal{PL}(\Theta|\vec{\mu}, \vec{\kappa}, \Lambda) = (2\pi)^{-pn} \prod_{j=1}^{p} \prod_{i=1}^{n} \frac{e^{\kappa_{\setminus \mathrm{J}}^{(i)} cos(\theta_{i,j} - \mu_{\setminus \mathrm{J}}^{(i)})}}{I_0(\kappa_{\setminus \mathrm{J}}^{(i)})},$$

$$\text{such that: } \mu_{\setminus \mathrm{J}}^{(i)} = \mu_j + tan^{-1}\left(\frac{\sum_{l \neq j} \lambda_{j,l} sin(\theta_{i,l} - \mu_l)}{\kappa_j}\right)$$

$$\kappa_{\setminus \mathrm{J}}^{(i)} = \sqrt{\kappa_j^2 + (\sum_{l \neq j} \lambda_{j,l} sin(\theta_{i,l} - \mu_l))^2}.$$

5.2 Consistency of the pseudo likelihood estimator

Dillon and Lebanon show that a maximum pseudo likelihood estimator is consistent provided that the mapping between conditional probabilities and joint probability is $injective$, i.e. the joint probability can be uniquely specified by the set of conditionals [3]. This property does hold true for von Mises.

Proof: Consider two conditionals with different parameters ($\vec{\kappa}_1^*$ and $\vec{\kappa}_2^*$, and $\vec{\mu}_1^*$ and $\vec{\mu}_2^*$), which have the same conditional distributions.

$$[I_0(\kappa_1^*)]^{-1} e^{\kappa_1^* cos(\theta - \mu_1^*)} = [I_0(\kappa_2^*)]^{-1} e^{\kappa_2^* cos(\theta - \mu_2^*)}$$

By taking the derivative of the two conditionals based on θ, and equating the two derivatives, and setting those equal, we get the system of equations:

$$\kappa_1^* cos(\theta - \mu_1^*) = \kappa_2^* cos(\theta - \mu_2^*)$$
$$\kappa_1^* sin(\theta - \mu_1^*) = \kappa_2^* sin(\theta - \mu_2^*)$$

From which we conclude κ_1^*=κ_2^*, and μ_1^*=μ_2^*, for all i and j values.

So far we have shown that the conditional probability equality results in equality of the hyper parameters, κ^*s and μ^*s. These parameters are defined in equations (1) and (2), so now we have to show individual parameters are equal as well. (i.e. for each i and j, $\kappa_{1i} = \kappa_{2i}$ and $\lambda_{1ij} = \lambda_{2ij}$.)

Because the equalities κ_1^*=κ_2^* are held true for any θ value, we can set $\theta_i = \mu_i^*$ in equation (1). This decision eliminates the Sin term, and directly results in $\kappa_{1i}^2 = \kappa_{2i}^2$. And since κ is positive by definition, we conclude that for all i, $\kappa_{1i} = \kappa_{2i}$.

On the other hand, we can also set $\theta_i = \mu_i^* + \frac{\pi}{2}$ in equation (2), which results in the following system of equations. For all i and j,

$$\sum_{l \neq j} \lambda_{1jl} = \sum_{l' \neq j} \lambda_{2jl'}$$

This system has only one solution, which is, for all i and j, $\lambda_{1ij} = \lambda_{2ij}$. And with this conclusion, we have shown that knowing the conditional distributions for von Mises is enough to specify the whole probability distribution, and consequently, the theorem discussed in [3] proves that the Full Pseudo Likelihood is a $consistent$ estimator for the vGM.

5.3 Structure learning for vGM

The study of structure learning problem has received considerable attention recently (e.g., [15, 8, 7, 13]). Structure learning algorithms based on L_1 regularization are particularly interesting because they exhibit consistency and high statistical efficiency (see [14] for a review). We use an algorithm introduced by Schmidt *et.al* [13] that solves the L_1-regularized maximum likelihood estimation optimization problem using gradient projection. Their algorithm can be applied to any differentiable continuous loss function, without any specific functional forms assumed. In particular, for $x = (x_1, x_2, ..., x_n)$ and loss function L, their algorithm minimizes functions of the form:

$$min_x f(x) \equiv L(x) + \rho \|x\|_1 \text{ , where} \|x\|_1 = \sum_{i=1}^{n} |x_i|$$

Here, ρ corresponds to regularization parameter. The L_1-Projection method reformulates this problem as a constrained optimization problem. Schmidt et. al. [13] rewrite the absolute value as a differentiable function:

$$|x| \approx \frac{1}{\alpha}[log(1 + e^{-\alpha x}) + log(1 + e^{\alpha x})]$$

As α goes to infinity, the approximation error goes to zero. They then perform projected gradient descent to reach the local optimum.

We use this method to learn the structure and parameters of the vGM. Specifically, we define the loss function L as the negative log of full pseudo likelihood, as defined in Section 5.1:

$$L(\Theta|\vec{\mu},\vec{\kappa},\Lambda) = -(np)log(2\pi) + \sum_{j=1}^{p}\sum_{i=1}^{n} -log(I_0(\kappa_{\backslash j}^{(i)})) + \kappa_{\backslash j}^{(i)} cos(\theta_{i,j} - \mu_{\backslash j}^{(i)}).$$

The sub-gradients of the loss function are calculated as follows. For each element of $\vec{\kappa}$, κ_R we have:

$$\frac{\partial log(\mathcal{L})}{\partial \kappa_R} = \kappa_R \sum_{i=1}^{n} \left(\frac{cos(\theta_{iR} - \mu_{\backslash R}^{(i)}) - A_0(\kappa_{\backslash R}^{(i)})}{\kappa_{\backslash R}^{(i)}} + \frac{sin(\theta_{iR} - \mu_{\backslash R}^{(i)}) * \sum_{l \neq R} \lambda_{Rl} sin(\theta_{il} - \mu_l)}{\kappa_{\backslash R}^{(i)}} \right)$$

Where $A_0(\kappa)$ is defined as $\frac{I_1(\kappa)}{I_0(\kappa)}$ as described in [9].

Taking derivative of the pseudo likelihood with respect to each element of Λ matrix, $\lambda_{R,S}$, is also:

$$\frac{\partial log(\mathcal{L})}{\partial \lambda_{R,S}} = \sum_{j=1}^{p}\sum_{i=1}^{n} \left(\frac{\partial \kappa_{\backslash j}^{(i)}}{\partial \lambda_{R,S}} [-A_0(\kappa_{\backslash j}^{(i)}) + cos(\theta_{i,j} - \mu_{\backslash j}^{(i)}] + \frac{\partial \mu_{\backslash j}^{(i)}}{\partial \lambda_{R,S}} \kappa_{\backslash j}^{(i)} sin(\theta_{i,j} - \mu_{\backslash j}^{(i)}) \right)$$

$$\text{such that, } \frac{\partial \kappa_{\backslash j}^{(i)}}{\partial \lambda_{R,S}} = \delta(R,J) * \frac{\sum_{l \neq j} \lambda_{j,l} sin(\theta_{i,l} - \mu l) * sin(\theta_{i,s} - \mu_s)}{\kappa_{\backslash j}^{(i)}}$$

$$\frac{\partial \mu_{\backslash j}^{(i)}}{\partial \lambda_{R,S}} = \delta(R,J) * \frac{sin(\theta_{i,s} - \mu_s)}{\kappa_j * (1 + [\frac{\sum_{l \neq j} \lambda_{j,l} * sin(\theta_{i,l} - \mu l)}{\kappa_j}]^2)}$$

These gradients are then used in the projected gradient method to solve the maximum pseudo likelihood estimation for the parameters of the von Mises graphical model.

6 Experiments

We evaluated our algorithm on synthetic and real protein data. The synthetic data was generated using the Gibbs sampler in section 4, and the real data come from a molecular dynamics (MD) simulation of the protein ubiquitin. We compare our model to the Gaussian Graphical Model (GGM) [12].

6.1 Parameter Learning on Synthetic Data

We generated random vGM graphs for different parameter configurations by systematically varying: (a) the number of nodes of graph from 8 to 128; (b) the density of edges of the graph from 0.1% to 100%; and (c) the von Mises parameters $\vec{\kappa}$ and Λ. For each parameter configuration, we generated 50 vGMs by randomly generating the elements of $\vec{\kappa}$ using a uniform distribution on $[0, S_\kappa]$. Here, S_κ ranged from 10^{-2} to 10^2. Elements of the Λ matrix were drawn from a Gaussian distribution $\mathcal{N}(0, S_\Lambda)$ where S_Λ ranged from 10^{-2} to 10^2. In these synthetic datasets, the mean values for the marginal distributions, $\vec{\mu}$, were held fixed at zero.

Evaluation Metric Our evaluation metric is the $cosine$ of the angle between the true and estimated model parameters ($\vec{\kappa}$ and Λ). The cosine of angle between vectors A and B is defined as $\frac{A^T B}{\|A\|_2 \|B\|_2}$ and values closer to one indicate higher similarity. We used this metric as an indicator of the quality of algorithm in learning not only the structure, but the strength of the links in the graph.

Model Selection The structure learning algorithm has one free parameter – the regularization penalty for adding edges. We selected the optimal value for this parameter by first randomly shuffling each column of the samples (columns correspond to variables), to remove all correlation between the variables. Then we learned a vGM for many values of regularization penalty on this shuffled data, and selected the lowest penalty that did not capture any dependencies on the data. The same procedure was used to find the penatly for the GGM.

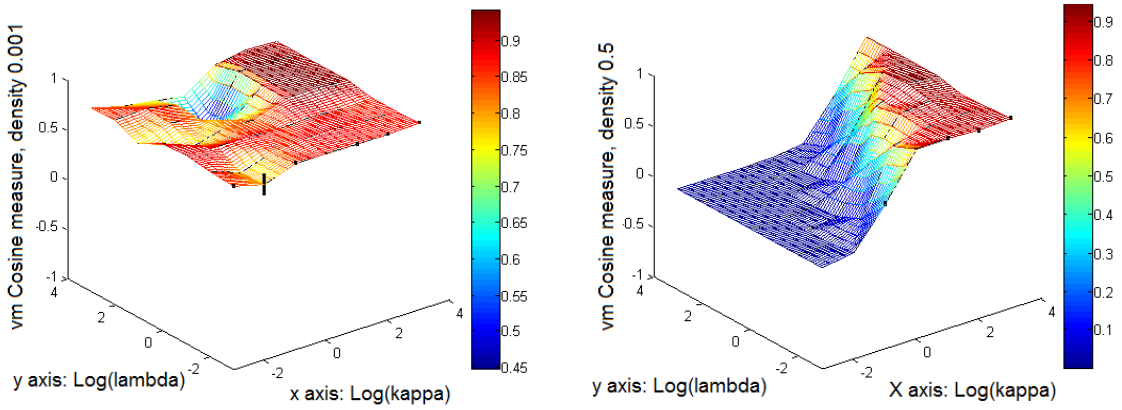
Results Figures 2(a) and 2(b) present surface plots depicting the cosine angles between the true and learned parameters for varying edge densities. In each figure, the x and y axes correspond to the log of S_κ and S_Λ (defined above), while the z axis and the color is the average cosine angle between the true and learned parameters, averaged over the 50 data sets of that configuration.

At very low edge density (Fig. 2(a)), the variables are mostly independent and the algorithm successfully learns the $\vec{\kappa}$ values over all combinations of the true values of $\vec{\kappa}$ and Λ. At 50% edge density (and also at higher densities, as we observed), the effect of the magnitude of the κ and Λ values becomes evident. In particular, accuracy is positively correlated with the magnitude of the κ's and inversely correlated with the magnitude of the λ's. Recall that κ is inversely related to the marginal variances, and that the elements of the Λ matrix correspond to the strength of the coupling/correlation between variables. Thus, accuracy of the learning algorithm decreases under high variance, and/or strong couplings.

Comparison to Gaussian Graphical Model As previously mentioned (Sec. 3), a vGM can be well-approximated with a GGM when the variables have low variance (i.e., high values of κ).

Under the condition where $(\theta_i - \mu_i)$ approaches zero (i.e., when the marginal variance of each variable is sufficiently small), a vGM can be approximated with a multivariate Gaussian distribution. (i.e. $f_{VMM}(\vec{\mu}, \vec{\kappa}, \Lambda) \propto f_{GGM}(\mu, \Sigma)$, where $(\Sigma^{-1})_{ii} = \kappa_i$, and$(\Sigma^{-1})_{ij} = -\Lambda_{ij}$).

We ran the GGM regularized learning algorithm [12] to determine if it has lower accuracy than the vGM when the variance of the variables is higher.



(a) (b)

Figure 2: Cosine angles between true and learnt parameters at (a) 0.1%(i.e. mostly independent variables), and (b) 50% edge density. Standard error bars are shown as black bars.

Figure 3 shows the average *difference* in the performance of the two learning algorithms for different parameter combinations (S_κ and S_Λ) at a fixed density of 50%. Each point in the plot is calculated by computing the cosine of the angle between the true and estimated model parameters obtained using the vGM algorithm minus the same quantity for the GGM algorithm. Thus, the peak in the contour plot corresponds to parameter combinations where vGM outperforms the GGM the most.

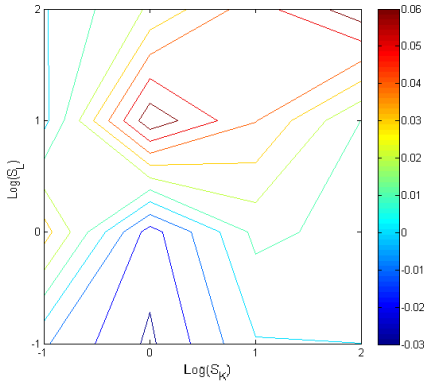


Figure 3: Performance of the vGM learning algorithm versus the GGM learning algorithm for different parameter combinations. See text for details.

Interestingly, the surface in Figure 3 is not monotonic. In the lower right corner, where the concentration is high (i.e., low variance) *and* the coupling between variables is low, the GGM's

performance is essentially the same as the vGM. This is expected. However, in the upper right corner, we see that the vGM can outperform the GGM when variance is low, provided that the average coupling is high. The peak in the plot occurs at approximately the point where $\log S_\kappa$ is zero and $\log S_\Lambda$ is one. Thus, the relative performance of the vGM increases as variance increases. But this is only true up to a point. At very low concentrations (i.e., high variances) the performance of both algorithms is about the same (left-most edge). Note that the GGM outperforms the vGM at approximately the point where $\log S_\kappa$ is zero and $\log S_\Lambda$ is negative one, but that the depth of the trough (≈ -0.03) isn't as deep as the hight of the peak (≈ 0.06). Thus, we conclude that the vGM performs as well, or better than the GMM over the majority of the parameter combinations we considered.

6.2 Parameter Learning Cross Validation on Protein Torsion Angle Data

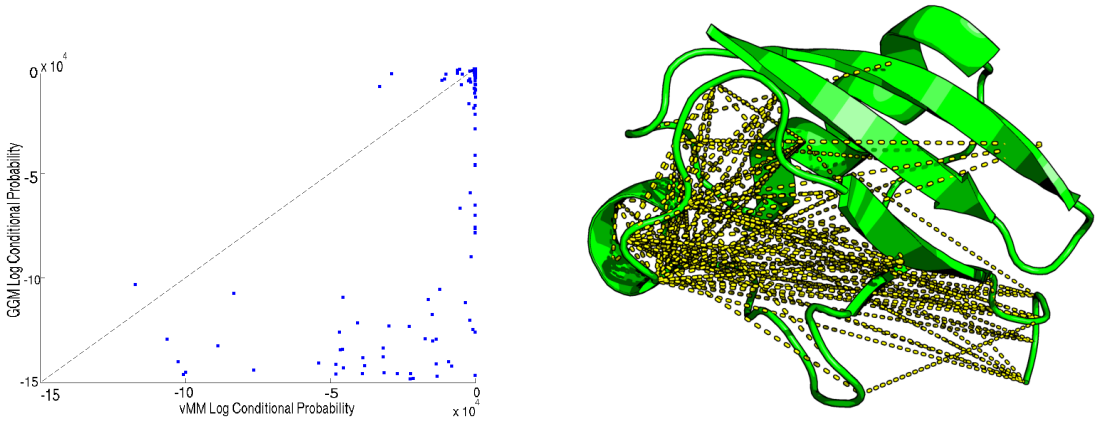
A protein is a linear chain of smaller molecules known as amino acids. The three dimensional structure of a protein can be defined in terms of the Cartesian coordinates of the constituent atoms or, equivalently (and with fewer parameters) in terms of a series of dihedral (aka torsion) angles. Each protein samples from an underlying distribution over configurations (known as the Boltzmann distribution) according to the laws of physics. Characterizing these distributions is very important for understanding the biological function of these complex molecules. Thus, a vGM is a suitable choice for modeling a protein's structure.

We applied our von Mises graphical model learning algorithm to learn a model of the joint distribution over a subset of the dihedral angles in the protein ubiquitin, which has 76 amino acids. The data set consisted of 15000 observations obtained via molecular dynamics simulation. Each observation consists of a vector of dihedral angles defining the structure of the protein.

We performed a 5-fold cross validation. and at each run, computed the probability of each variable conditioned on the rest of the observations, given the learned parameters. This conditional probability was calculated using the formula derived in section 4. We also learned a Gaussian Graphical Model on the same protein data, using the algorithm described in [12] and we performed the same cross validation procedure as with the vGM .

Figure 4(a) shows the log probability of each dihedral angle in the test set under the model learned on the training set. Each dot in the plot corresponds to one dihedral angle log probability. For a large number of dihedral angles, the log likelihood of the test set under the vGM is higher than their likelihood under GGM. There is also a large number of positions where both models perform comparably – these correspond to positions in the protein that are fairly constrained and do not fluctuate significantly. This echoes the intuition that when the angles are relatively constrained (i.e. have low variance), both von Mises and Gaussians have similar behavior. This result also demonstrates that overall the vGM is a better fit of directional data than the GMM when the variables have high variance.

Figure 4(b) overlays the edges on the the a depiction of the structure of ubiquitin. Notice that there are many edges between distant parts of the protein. While beyond the scope of this paper, we note that such long-range dependencies are consistent with the biological function of the protein which binds to other molecules via a gripping motion.



(a) The log conditional probability of each variable under the vGM and GGM models.

(b) The dependency links between the torsion angles of the Ubiquitin Protein backbone.

Figure 4: vGM and GMM performance on Ubiquitin protein data.

7 Conclusion and Future Work

In this paper we presented the first multivariate von Mises graphical model and introduced algorithms for sampling and structure learning. While clearly well suited to modeling networks of circular variables, the von Mises graphical model hasn't achieved more widespread use due to previously unsolved technical challenges associated with the mathematical form of the distribution. We have shown how to overcome those challenges by developing a maximum full pseudo likelihood estimator and then employing a recent gradient based algorithm for parameter and structure learning. Our algorithm inherits desirable properties of the pseudo likelihood and the learning algorithm, including consistency and high statistical efficiency.

We tested the quality of our estimator on a set of synthetic data created by the Von Mises sampler, and then compared our estimator to the regularized Gaussian Graphical Model estimator. We observed that the Von Mises model has a better accuracy compared to Gaussian Graphical Models across a fairly large range of parameter combinations. We also applied our model to the dihedral angles of the protein ubiquitin. Comparing the conditional probabilities of each variable conditioned on the rest of showed us that Von Mises is a better fit for the protein data, and can recover long distance dependencies between the movements of residues.

Finally, we note that we have recently derived the update equations for an Expectation-Propagation style inference algorithm for the vGM[11].

References

[1] Boomsma, Wouter, Mardia, Kanti V., Taylor, Charles C., Ferkinghoff-Borg, Jesper, Krogh, Anders, and Hamelryck, Thomas. A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences*, 105(26):8932–8937, July 2008.

[2] Ernst Breitenberger. Analogues of the normal distribution on the circle and the sphere. *Biometrika*, 50:81–82, 1963.

[3] Joshua Dillon and Guy Lebanon. Statistical and computational tradeoffs in stochastic composite likelihood. 2009.

[4] N.I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, 1993.

[5] Tim Harder, Wouter Boomsma, Martin Paluszewski, Jes Frellsen, Kristoffer E. Johansson, and Thomas Hamelryck. Beyond rotamers: a generative, probabilistic model of side chains in proteins. *BMC Bioinformatics*, 11:306, 2010.

[6] Gareth Heughes. *Gareth Hughes. Multivariate and time series models for circular data with applications to protein conformational angles*. PhD Thesis, Department of Statistics, University of Leeds, 2007.

[7] Holger Hofling and Robert Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10:883–906, April 2009.

[8] Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of markov networks using l_1-regularization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 817–824. MIT Press, Cambridge, MA, 2007.

[9] K. V. Mardia. Statistics of directional data. *J. Royal Statistical Society. Series B*, 37(3):349–393, 1975.

[10] K.V. Mardia and P.E. Jupp. *Directional statistics*. Wiley Chichester, 2000.

[11] Narges Sharif Razavian, Hetunandan Kamisetty, and Christopher James Langmead. The von mises graphical model: Expectation propagation for inference. Technical Report CMU-CS-11-130, Carnegie Mellon University, 2011.

[12] Mark Schmidt, Glenn Fung, and Rmer Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *In Proceedings of European Conference on Machine Learning*, pages 286–297, 2007.

[13] Mark Schmidt, Kevin Murphy, Glenn Fung, and Rmer Rosales. Structure learning in random fields for heart motion abnormality detection. In *CVPR*. IEEE Computer Society, 2008.

[14] JA Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

[15] Martin J. Wainwright, Pradeep Ravikumar, and John D. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1465–1472. MIT Press, Cambridge, MA, 2007.

[16] Richard S. Zemel, Christopher K. I. Williams, and Michael Mozer. Directional-unit boltzmann machines. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 172–179, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[17] Richard S. Zemel, Christopher K. I. Williams, and Michael C. Mozer. Lending direction to neural networks. *NEURAL NETWORKS*, 8:503–512, 1995.