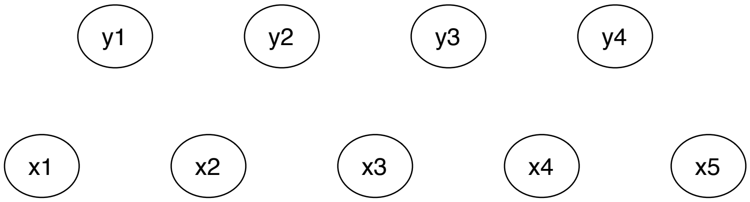
CPSC 540: Assignment 5

Yiwei Hou(84435156)
Xiaomeng Ju(86475150)
Tingting Yu(74439118)

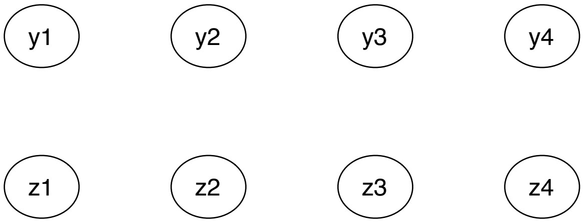
1 Undirected Graphical Models

1.1 Conditional UGM

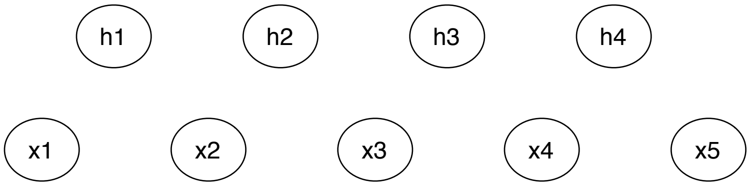
1. The UGM conditional on all the z and h values is given below. It wouldn't be expensive to compute univariate marginals as the variables are independent of each other conditional on z and h.



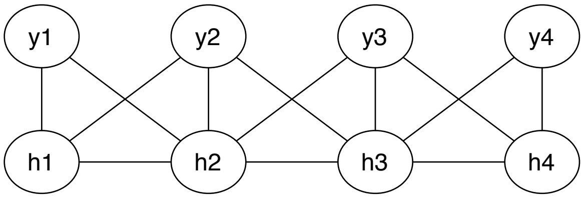
2. The UGM conditional on all the x and h values is given below. It wouldn't be expensive to compute univariate marginals as the variables are independent of each other conditional on x and h.



3. The UGM conditional on all the z and y values is given below. It wouldn't be expensive to compute univariate marginals as the variables are independent of each other conditional on z and y.



4. The UGM conditional on all the x and z values is given below. It would be quite expensive to compute univariate marginals as the variables are closely connected with each other.



1.2 Fitting a UGM to PINs

1. Because there are 10 possible values (states) for each pin number (node). Four nodes are not connected. With parameter tieing, each node has the same potentials. For each pin number (node) in our UGM model, we need to model the potential for all 10 states. However, scaling the potentials by constant doesn't change the distribution. So the last potential (for instance, $\phi(10)$) can be scaled to always be one and the log value is zero. Thus we only need 9 parameters for the whole UGM model.

2.

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z} \prod_{i=1}^{4} \exp(\theta_{x_i}),$$

where $x_i \in \{0, 1, 2, \cdots, 9\}$, Z is a constant, and θ is a vector of length 10 with the last element being 0. θ_{x_i} is the $(x_i + 1)$-th entry of θ.

3. By using tied parameters and a zero adjacency matrix, the four nodes have identical estimated potentials and the decoding will always return four identical number. So the model should allow different nodes to have different potentials for the same states. Also, the pin numbers may be associated with each other. The model should allow for dependence structure between the nodes and take into account pairwise potentials as well.

4. By setting tied=0, we allow each node to have distinct probability distribution of the states. So there are 9 parameters per node and 36 parameters in total.

The decoding now is 1 2 3 4 and the marginal probability of the first number being 1 is 0.3854. So the first two issues are fixed.

However, the model returns $P(x_4 = 4|x_1 = 1, x_2 = 2, x_3 = 3) = 0.1894$, which is still much lower than the probability observed in the data (more than 90%). So the third issue is not fixed.

5. By using chain-structured dependency, the decoding is 1 2 3 4, the marginal probability of the first number being 1 is 0.3854 and $P(x_4 = 4|x_1 = 1, x_2 = 2, x_3 = 3)$ is increased to 0.5820. So the first two issues are fixed and the third issue is somewhat fixed.

6. By using a completely-connected graph, the decoding is 1 2 3 4, the marginal probability of the first number being 1 is 0.3854 and $P(x_4 = 4|x_1 = 1, x_2 = 2, x_3 = 3)$ is further increased to 0.7843. So the three issues are well fixed.

7. Adding higher-order potentials will achieve more accurate conditional probabilities such as $P(x_4 = 4, x_1 = 1|x_2 = 2, x_3 = 3)$. The disadvantage of higher-order potentials would be that the inference becomes more expensive due to extra parameters and dependencies in the model.

2 Bayesian Inference

2.1 Conjugate Priors

1. The posterior distribution $p(\theta|y, \alpha)$

$$\begin{aligned} p(\theta|y,\alpha) &\propto p(y|\theta,\alpha)p(\theta|\alpha) \\ &= \theta_1^{I(y=1)}\theta_2^{I(y=2)}\theta_3^{I(y=3)}\frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)}\theta_1^{\alpha_1-1}\theta_2^{\alpha_2-1}\theta_3^{\alpha_3-1} \\ &\propto \theta_1^{\alpha_1-1+I(y=1)}\theta_2^{\alpha_2-1+I(y=2)}\theta_3^{\alpha_3-1+I(y=3)} \end{aligned}$$

Therefore, $p(\theta|y,\alpha)$ is specified as a Dirichlet distribution: $D(I(y=1)+\alpha_1, I(y=2)+\alpha_2, I(y=3)+\alpha_3)$.

2. For the simplicity of notation, Let $D(\alpha) = \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)}$ and $D(\alpha^+) = \frac{\Gamma(I(y=1)+I(y=2)+I(y=3)+\alpha_1+\alpha_2+\alpha_3)}{\Gamma(I(y=1)+\alpha_1)\Gamma(I(y=2)+\alpha_2)\Gamma(I(y=3)+\alpha_3)}$

$$\begin{aligned} p(y|\alpha) &= \int p(y,\theta|\alpha)d\theta \\ &= \int p(y|\theta,\alpha)p(\theta|\alpha)d\theta \\ &= \int D(\alpha)\theta_1^{\alpha_1-1+I(y=1)}\theta_2^{\alpha_2-1+I(y=2)}\theta_3^{\alpha_3-1+I(y=3)}d\theta \\ &= \frac{D(\alpha)}{D(\alpha^+)}\int D(\alpha^+)\theta_1^{\alpha_1-1+I(y=1)}\theta_2^{\alpha_2-1+I(y=2)}\theta_3^{\alpha_3-1+I(y=3)}d\theta \\ &= \frac{D(\alpha)}{D(\alpha^+)} \\ &= \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)}\frac{\Gamma(I(y=1)+\alpha_1)\Gamma(I(y=2)+\alpha_2)\Gamma(I(y=3)+\alpha_3)}{\Gamma(I(y=1)+I(y=2)+I(y=3)+\alpha_1+\alpha_2+\alpha_3)} \\ &= \frac{\Gamma(I(y=1)+\alpha_1)\Gamma(I(y=2)+\alpha_2)\Gamma(I(y=3)+\alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)(\alpha_1+\alpha_2+\alpha_3)} \end{aligned}$$

3.

$$\begin{aligned} E_{\theta|y,\alpha}[\theta_1] &= \int \theta_1 p(\theta|y,\alpha)d\theta \\ &= D(\alpha^+)\int \theta_1^{\alpha_1+I(y=1)}\theta_2^{\alpha_2-1+I(y=2)}\theta_3^{\alpha_3-1+I(y=3)}d\theta \\ &= D(\alpha^+)\frac{\Gamma(\alpha_1+I(y=1)+1)\Gamma(I(y=2)+\alpha_2)\Gamma(I(y=3)+\alpha_3)}{\Gamma(\alpha_1+\alpha_2+\alpha_3+2)} \\ &= \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3+1)\Gamma(\alpha_1+I(y=1)+1)\Gamma(I(y=2)+\alpha_2)\Gamma(I(y=3)+\alpha_3)}{\Gamma(I(y=1)+\alpha_1)\Gamma(I(y=2)+\alpha_2)\Gamma(I(y=3)+\alpha_3)\Gamma(\alpha_1+\alpha_2+\alpha_3+2)} \\ &= \frac{\alpha_1+I(y=1)}{\alpha_1+\alpha_2+\alpha_3+1} \end{aligned}$$

The derivations for θ_2 and θ_3 are similar to the one shown above. Therefore,

$$E_{\theta|y,\alpha}[\theta_1] = \frac{\alpha_i+I(y=i)}{\alpha_1+\alpha_2+\alpha_3+1}$$

4.

$$\begin{aligned} p(\hat{y}|y,\theta) &= \int p(\hat{y},\theta|y,\alpha)d\theta \\ &= \int p(\hat{y}|\theta)p(\theta|y,\alpha)d\theta \\ &= \int \theta_1^{I(\hat{y}=1)}\theta_2^{I(\hat{y}=2)}\theta_3^{I(\hat{y}=3)} D(\alpha^+)\theta_1^{\alpha_1-1+I(y=1)}\theta_2^{\alpha_2-1+I(y=2)}\theta_1^{\alpha_3-1+I(y=3)}d\theta \\ &= D(\alpha^+)\int \theta_1^{\alpha_1-1+I(y=1)+I(\hat{y}=1)}\theta_2^{\alpha_2-1+I(y=2)+I(\hat{y}=2)}\theta_1^{\alpha_3-1+I(y=3)+I(\hat{y}=3)}d\theta \end{aligned}$$

Let $\hat{D}(\alpha) = \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3+2)}{\Gamma(\alpha_1+I(y=1)+I(\hat{y}=1))\Gamma(\alpha_2+I(y=2)+I(\hat{y}=2))\Gamma(\alpha_3+I(y=3)+I(\hat{y}=3))}$

$$p(\hat{y}|y,\theta) = \frac{D(\alpha^+)}{\hat{D}(\alpha)}$$

2.2 Empirical Bayes

The optimal degree of polynomial is 3, α is 0.1, and λ is 10^{-4}.

```
function [model] = leastsSquaresEmpiricalBaysis(x,y)

degree_candidate = 1:10;
sigma_candidate = 10.^(-15:15)
lambda_candidate = 10.^(-15:15);
marginal_cur = Inf;

NLML = zeros(length(degree_candidate),length(sigma_candidate),length(lambda_candidate));
for i = 1:length(degree_candidate)
    for j = 1:length(sigma_candidate)
        for k = 1:length(lambda_candidate)
            [degree_candidate(i), sigma_candidate(j), lambda_candidate(k)];
            tmp = eval_marginal(x,y,degree_candidate(i), sigma_candidate(j), lambda_candidate(k));
            NLML(i,j,k) = tmp;
        end
    end
end

model.NLML = NLML;
end

function [tmp] = eval_marginal(x,y,degree, sigma, lambda)
Xpoly = polyBasis(x,degree);
C = 1/sigma^2*eye(size(x,1)) + 1/lambda* Xpoly*Xpoly';
v = C\y;
tmp = logdet(C,Inf) + y'*v;
end

model = leastsSquaresEmpiricalBaysis(x,y);
a = find(model.NLML==min(model.NLML(:)));
[I1,I2,I3] = ind2sub(size(model.NLML),a);
degree_candidate(I1)
sigma_candidate(I2)
lambda_candidate(I3)
```

3 Literature Survey

1. Background

Given a sample of observed entries in a matrix M, the task of completing unknown entries is known as the matrix completion, or matrix approximation problem. Matrix completion is widely used in recommendation system, image processing, dimension deduction and etc. In recommendation system, the matrix corresponds to users' ratings of a selection of items (e.g. movies) with the entry M_{ij} being the rating by users i for item j. The goal is to predict the rating for an unobserved entry in M when only a sample of M_{ij} are observed. In general, matrix completion is under-specified as there are uncountably infinite number of matrices that agree with the observed entries of M. A common assumption is that M is a low-rank matrix, which means $M \in \mathbb{R}^{n_1 \times n_2}$ can be well approximated by a rank r matrix $\hat{M} = UV^T$, where $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, and $r << min(n_1, n_2)$. Based on many real-life data sets, this assumption is realistic. Theoretically, under this assumption and certain assumptions on the entries of the matrix, locations and proportion of unobserved entries, the true underlying matrix can be accurately recovered. See [1, 2, 3].

2. Global Low-rank Matrix Approximation

A number of approaches have been proposed for matrix completion. For a matrix M, let $\Omega \subset \{1, 2, \cdots, n_1\} \times \{1, 2, \cdots, n_2\}$ denote the indices of observed entries. The key idea is to consider the following rank minimization problem:

$$\begin{aligned}&\text{minimize rank}(Z)\\&\text{subject to } \textstyle\sum_{i,j\in\Omega}(M_{ij} - Z_{ij})^2 \leq \delta,\end{aligned}$$

for user defined $\delta \geq 0$ and some $Z \in \mathbb{R}^{n_1 \times n_2}$. However, this optimization problem is NP hard. Greedy selection approach has been proposed to tackle this problem, see [4, 5]. Another approach is to relax the rank term into a trace or nuclear norm:

$$\begin{aligned}&\text{minimize } ||Z||_*\\&\text{subject to } \textstyle\sum_{i,j\in\Omega}(M_{ij} - Z_{ij})^2 \leq \delta,\end{aligned}$$

where the nuclear norm $|| \cdot ||_*$ is the sum of the singular values of a matrix. The above problem is a convex problem and can be formulated in Lagrange form: min $\frac{1}{2}\sum_{i,j\in\Omega}(M_{ij} - Z_{ij})^2 + \lambda||Z||_*$, where λ and δ have one-to-one projection. In [6] the nuclear norm penalized objective is approximated by writing the penalty in terms of the minimum Frobenius norm factorization, and solving it using a parallel projected incremental gradient method. The resulting problem can be considered as a generalization of Maximal-Margin Matrix Factorization (MMMF) [7]. Note that the optimization of the Lagrange form can be considered a generalization of the $L1$ regularized least squares problem which is addressed in [8]. As with $L1$ and $L0$ linear regression, minimizing the nuclear norm can outperform the rank minimized solution, as supported empirically in [9].

In [10] the authors propose a Singular Value Thresholding (SVT) algorithm for the optimization of nuclear norm equation in which $\delta = 0$. In [11] the authors use an accelerated proximal gradient algorithm which gives an ϵ-accurate solution. A nuclear norm minimization subject to linear and second order cone constraints is solved in [12], with an application to recommendation on a large movie rating dataset. The soft impute algorithm of [9] is inspired by SVT, it does not require a step size parameter. Instead, soft impute is controlled using the regularization parameter λ, and using warm restarts one can compute the complete regularization path for model selection. The algorithm is shown to be competitive to SVT and MMMF and scalable to relatively large datasets.

3. Local Low-rank Matrix Approximation

The intuition behind the local low-rank assumption is as follows: the entire rating matrix is not low-rank but sub-matrices restricted to certain types of similar users and items (for example, children users viewing cartoon movies) are. The papers summarized below solve the local low-rank matrix completion using different approaches.

Lee et al. (2013) proposed a novel approach that approximates a matrix with weighted sum of low-rank matrices. Each of these matrices is a mapping of the original matrix that preserves a particular region of the matrix [13]. They have showed that their local-rank modeling is significantly more accurate than global low-rank modeling in the context of recommendation systems and allows for parallel implementation in large scale problems. However, the low-rank matrix approximation may still exhibit poor scalability as the mapping of the original matrix is in the same size as the original matrix, so the matrix factorization on the large matrix may be highly computaitionally complex.

Chen et al. (2015) proposed a weighted and ensemble matrix approximation method that achieves both high accuracy and high scalability for CF-based recommendation [14]. The idea is to first adopt co-clustering methods to partition the large user-item rating matrix into a set of smaller submatrices [15, 16, 17] and then propose a weighting strategy based on the intuition that submatrices containing more frequent samples of certain user/item/rating tend to make more reliable rating predictions for these specific user/item/rating. In specific, the method consists of two important components: (i) a weighting strategy that is computed based on the rating distribution in each submatrix and applied to approximate a single matrix containing those submatrices; and (ii) an ensemble strategy that leverages user-specific and item-specific rating distributions to combine the approximation matrices of multiple sets of co-clustering results.

4. Online Matrix Completion

So far, we have discussed several (local or global) low-rank matrix approximation methods for off-line data. In practice, entries of the matrix of interest are observed in a sequential order and recomputing the matrix decompositions, such as singular value decomposition (SVD), for matrix approximation at each new time point is very tedious and computationally expensive when the matrix is in a large scale. So the extension of existing matrix completion methods to the sequential prediction context is in demand for Big Data, and yet little addressed in the literature.

Dhanjal et al. (2014) proposed an online version of the Soft Impute algorithm that can efficiently evaluate the SVD using randomized SVD [18]. This improvement allows to bypass the bottleneck in the algorithm which consists in the use of the SVD of a large matrix at each iteration. Additionally, under a sparse and potentially high-rank change, they showed that the matrix completion can be conducted in an online setting by using previous solutions.

References

[1] E. J. Candés, B. Recht.(2009) Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717-772.

[2] E. Candés and T. Tao. (2010) The power of convex relaxation: Near-optimal matrix completion. Information Theory, IEEE Transactions on, 56(5):2053-2080.

[3] R. H. Keshavan, A. Montanari, and S. Oh. (2010) Matrix completion from a few entries. Information Theory, IEEE Transactions on, 56(6):2980-2998.

[4] K. Lee and Y. Bresler. Admira. (2010) Atomic decomposition for minimum rank approximation. Information Theory, IEEE Transactions on, 56(9):4402-4416.

[5] . Shalev-Shwartz, A. Gonen, and O. Shamir.(2011) Large-scale convex minimization with a lowrank constraint. arXiv preprint arXiv:1106.1622.

[6] B. Recht, C. Ré, and S. Wright.(2011) Parallel stochastic gradient algorithms for large-scale matrix completion. Optimization Online.

[7] N. Srebro, J. D. Rennie, and T. Jaakkola. (2005) Maximum-margin matrix factorization. Advances in neural information processing systems, 17(5):1329-1336.

[8] R. Tibshirani.(1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267-288.

[9] . Mazumder, T. Hastie, and R. Tibshirani.(2010) Spectral regularization algorithms for learning large incomplete matrices. The Journal of Machine Learning Research, 11:2287-2322.

[10] J.-F. Cai, E. J. Candé, and Z. Shen. (2010) A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956-1982.

[11] K.-C. Toh and S. Yun.(2010) An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pacific Journal of Optimization, 6(615-640):15.

[12] Y.-J. Liu, D. Sun, and K.-C. Toh.(2012) An implementable proximal point algorithmic framework for nuclear norm minimization. Mathematical Programming, 133(1-2):399-436.

[13] Lee, J., Kim, S., Lebanon, G., & Singer, Y. (2013). Local Low-Rank Matrix Approximation. ICML (2), 28, 82-90.

[14] Chen, C., Li, D., Zhao, Y., Lv, Q., & Shang, L. (2015, August). Wemarec: Accurate and scalable recommendation through weighted and ensemble matrix approximation. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval (pp. 303-312). ACM.

[15] George, T., & Merugu, S. (2005, November). A scalable collaborative filtering framework based on co-clustering. In Data Mining, Fifth IEEE international conference on (pp. 4-pp). IEEE.

[16] Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2007). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. Journal of Machine Learning Research, 8(Aug), 1919-1986.

[17] Xu, B., Bu, J., Chen, C., & Cai, D. (2012, April). An exploration of improving collaborative recommender systems via user-item subgroups. In Proceedings of the 21st international conference on World Wide Web (pp. 21-30). ACM.

[18] Dhanjal, C., Gaudel, R., & Clmenon, S. (2014, April). Online matrix completion through nuclear norm regularisation. In Proceedings of the 2014 SIAM International Conference on Data Mining (pp. 623-631). Society for Industrial and Applied Mathematics.