Errata for Probabilistic Graphical Models for Computer Vision

1 Chapter 1 Corrections

1. Page 1: ~~a RV~~ ⇒ a random variable
2. Page 7: ~~Bayesian estimation~~ ⇒ maximum a posteriori estimation

2 Chapter 2 Corrections

1. Page 13: An approximation to the chain rule is the pseudolikelihood **rule**, which states that

$$p(X_1, X_2, ..., X_N) \approx \prod_{n=1}^{N} p(X_n|X_{-n}) \tag{1}$$

   The rule plays an important in developing efficient PGM inference methods.
2. Page 22: ~~Maximum joint likelihood~~ ⇒ Maximum joint likelihood estimation
3. Page 23: Discriminative learning, however, cannot handle incomplete inputs and cannot obtain the marginal distribution of the inputs.
4. Page 23: ~~maximize~~ ⇒ maximizes
5. Page 23: ~~likelihood~~ ⇒ loglikelihood
6. Page 23: ~~Bayesian estimation~~ ⇒ Maximum A Posterior (MAP) estimation
7. Page 23: ~~Bayesian~~ ⇒ MAP
8. Page 23: ~~Bayesian~~ ⇒ MAP
9. Page 24: ~~Bayesian~~ ⇒ MAP
10. Page 24: ~~Bayesian~~ ⇒ MAP
11. Page 27: For standard multivariate distribution such as the multivariate Gaussian distribution, we can perform direct sampling through the re-parameterization trick. Readers may refer to Appendix 2.6.2 for details.

3 Chapter 3 Corrections

1. Page 42: While regression BN significantly reduces the number of parameters to specify the CPTs, it also introduces errors in modeling the distributions as the exact CPTs are now approximated by a small set of regression parameters. To reduce the approximation error, the linear regression can be approximated by non-linear regression functions such as an nth order polynomial regression function or even by a neural network.

2. Page 43: Eq. 3.15 $\Rightarrow$

$$\boldsymbol{x}_Q^* = \arg\max_{\boldsymbol{x}_Q} p(\boldsymbol{X}_Q = \boldsymbol{x}_Q | \boldsymbol{X}_E = \boldsymbol{x}_E) = \arg\max_{\boldsymbol{x}_Q} \sum_{\boldsymbol{x}_U \setminus \boldsymbol{x}_Q} p(\boldsymbol{X}_U = \boldsymbol{x}_U | \boldsymbol{X}_E = \boldsymbol{x}_E). \tag{2}$$

3. Page 43: ~~or max sum~~, as it involves finding the best configuration for all unknown variables. It is also called MPE inference.

4. Page 43: ~~depending on whether the maximum is taken with respect to the posterior probability or the log posterior probability.~~ $\Rightarrow$ Marginal MAP is also called max sum inference, as it involves finding the best configuration of only a subset of unknown variables. Summation is needed to marginalize out other unknown variables as shown in Eq. 3.15.

5. Page 46: ~~can~~

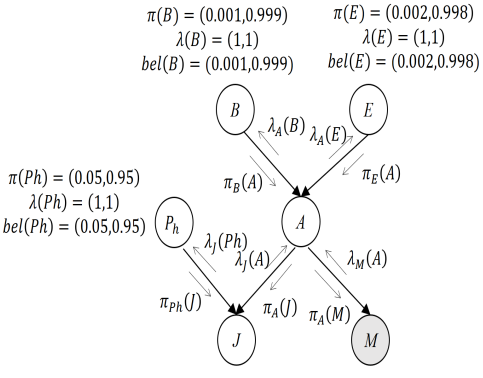
6. Page 50: The entries of CPDs involving the evidence node are set to zero if the value for the evidence node is different from the observed value and unchanged otherwise.

7. Page 50: This means belief propagation for each node can be performed in parallel and asynchronously.

8. Page 50: its total π and total λ messages ; and updates its total λ messages; ,and updates its total π messages.

9. Page 51: and updates its total λ messages

10. Page 51: Replace Figure 3.16 with the attached figure below



(B)

11. Page 51: **9**. Compute the final belief for each node and normalize.

12. Page 51: Refer to appendix 3.9.6.1 for additional examples of belief propagation, in particular for the case when the evidence nodes are non-boundary nodes.

13. Page 52: in Eq. (3.22), ~~is~~ ⇒ are

14. Page 52: An example of max-product inference can be found in Appendix 3.9.6.2.

15. Page 53: ~~Cluster~~ ⇒ To generate a valid junction tree, ~~running~~ ⇒ cluster

16. Page 54: (RIP) [1]. RIP states that the cliques should be ordered $(C_1, C_2, \ldots, C_k)$ so that for all $1 < j \leq k$, there is an $i < j$ such that $C_j \cap (C_1 \ldots C_{j-1}) \subset C_i$.

17. Page 58: Given a DAG, multiple valid junction trees may be constructed; they all yield the same inference results, but their computational complexity can vary. The optimal junction tree can lead to the most efficient inference.

18. Page 58: So far, we have discussed the exact methods for sum-product and max-product inferences. Sum-product (posterior probability) inference infers the posterior for one variable, while max-product inference (MAP) infers the most probable configuration for all unknown variable (max-product). In many real world applications such as presence of the latent variables, we may be interested in marginal MAP inference, i.e., inferring the best configuration for a subset of unknown variables. Marginal MAP inference is much more challenging than either sum-product or max-product as it involves both summation and multiplication, having complexity NP^{PP}-complete [1]. Various approximate solutions [2, 3] have been developed for marginal MAP inference.

19. Page 58: In addition, we can also perform approximate MAP inference using the Iterated conditional modes (ICM) method. Details about the ICM method may be found in Section 4.5.2.1.

20. Page 58: either sum-product or max-product

21. Page 60: remove the arrow symbol

22. Page 63: Original Metropolis–Hastings assumes symmetric proposal distribution. Later it was extended to non-symmetric proposal distribution by using the following equation to compute the acceptance probability.

$$p = \min(1, \frac{p'(x^t)q(x^{t-1}|x^t)}{p'(x^{t-1})q(x^t|x^{t-1})}) \tag{3}$$

    While simple and easy to implement, Metropolis–Hastings method does not scale up well as it's proposal distribution typically follows a random walk and is hence slow. To speed up, Hamiltonian Monte Carlo was introduced. The key idea is to use Hamiltonian dynamics instead of a random walk to propose a new sample. Details about the Hamiltonian Monte Carlo method may be found in Appendix 3.9.5.

[1] the RIP property is needed in order for the tree to satisfy the clustering intersection property

23. Page 68: ~~$\boldsymbol{\alpha}$~~ $\Rightarrow \boldsymbol{\alpha}_n$

24. Page 69: posterior

25. Page 69: Eq. 3.54 $\prod_{j=1}^{J} \Rightarrow \prod_{j=1}^{J_n}$

26. Page 69: Eq. 3.54 $p^{I(\pi^m(x_n)=j)}$

27. Page 69-70: $\prod_{j=1}^{J}\prod_{k=1}^{K} \Rightarrow \prod_{j=1}^{J_n}\prod_{k=1}^{K_n}$

28. Page 70: BN $\Rightarrow$ posterior probability, posterior

29. Page 70: In summary, point-based inference performs inference using a set of model parameters learned from the training data via a parameter maximization process. In contrast, Bayesian inference performs inference directly from the training data using all parameters via a parameter marginalization process. They are fundamentally different. Bayesian inference is robust to over-fitting, imbalanced, and insufficient data. But Bayesian inference is computationally expensive as it requires integration over the parameter space and hence does not scale up well.

30. Page 70: ~~p~~ $\Rightarrow q$, Such uncertain evidence is called soft or virtual evidence.

31. Page 70: ~~can perform expectation inference~~ $\Rightarrow$ treat the uncertain evidence as soft evidence and then employ Jeffrey's update rule [4] to perform the expected inference,

32. Page 70: Jeffrey's rule cannot be used to combine multiple uncertain evidences as it's result depends on the combination order of the uncertain evidences. To address this issue, we may treat the uncertain evidence as virtual evidence. Further information about inference under uncertain evidence may be found in Appendix 3.9.4.

33. Page 71: ~~Bayesian estimation (BE)~~ $\Rightarrow$ Maximum Posterior Probability (MAP)

34. Page 73: ~~Bayesian~~ $\Rightarrow$ MAP

35. Page 75: ~~Bayesian~~ $\Rightarrow$ MAP in four places

36. Page 77: In practice, $N_{ijk}+\alpha_{ijk} > 1$ and α_{ijk} are chosen as follows: $\alpha_{ijk} = M' \times p(x_i = k, \pi(x_i) = j)$, where M' is called equivalent sample size or prior strength and it can be tuned. $p(x_i = k, \pi(x_i) = j)$ is approximated by $\frac{1}{|x_i| \times |\pi(x_i)|}$, where $|\cdot|$ represents the cardinality of its argument. More details on selecting the Dirichlet prior can be found in [5].

37. Page 78: ~~Bayesian~~ $\Rightarrow$ MAP

38. Page 78: Finally, we briefly discuss Bayesian parameter learning. Different from MAP parameter learning, which uses the mode of the parameter posterior as the estimated parameters, Bayesian parameter learning uses the mean or

expectation as the estimated parameters, that is, $\theta^* = E_{p(\theta|\mathcal{D})}(\theta)$. For discrete BN with Dirichlet prior, the estimated parameters are

$$\theta^*_{njk} = \frac{N_{njk} + \alpha_{njk}}{\sum_{k=1}^{K} N_{njk} + \sum_{k=1}^{K} \alpha_{njk}}. \tag{4}$$

The estimated parameters θ^*_{njk} are very similar to those obtained by the MAP estimate in Eq. 3.81, with the only difference being -1 and $-K$ are removed respectively from the numerator and denominator.

39. Page 78: ~~Bayesian~~ $\Rightarrow$ MAP

40. Page 79: ~~Bayesian~~ $\Rightarrow$ maximum posterior probability or MAP

41. Page 79: ~~Exactly~~ $\Rightarrow$ For discrete BNs, Heckerman et al [6] shows that if we assume $p(\boldsymbol{\theta}|\mathcal{G})$ follows Dirichlet distribution and $P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta})$ follows multinormal distribution, Eq. (3.91) can be solved analytically with a closed solution. In general, exactly

42. Page 81: which renders the penalty term independent of the sample size, BD score is impractical as it is hard to fully specify the Dirichlet parameters,~~M~~ $\Rightarrow M^{'}$, where $M^{'}$ is called the equivalent sample size.

43. Page 81: $\alpha_{njk} = \frac{M'}{|x_n| \times |\pi(x_n)|}$, where $M^{'}$ is a tuning parameter for each node.

44. Page 81: because α_{njk} are the same for all j and k values of node n.

45. Page 82: The first term is the structure prior, which is different from the parameter prior in the BIC score.

46. Page 82: For a graph with N nodes, the number of possible directed graphs is $3^{\frac{N\times(N-1)}{2}}$. For example, the total number of directed graphs is 3^{45} for a graph with 10 nodes.

47. Page 83: Footnote: One way to impose the DAG constraint is to employ the Notear's constraint [7], which imposes the constraint on the weighted adjacent matrix that encodes the BN structure.

48. Page 83: ~~Another way~~ $\Rightarrow$ Other heuristics search methods ,

49. Page 83: ~~using the simulated annealing method~~ $\Rightarrow$ include random restart, TABU, and the simulated annealing method.

50. Page 84: ~~L^1~~ $\Rightarrow \ell_1$.

51. Page 84:
Instead of performing BN structure learning in discrete space, a recent work by Zheng et al [7] reformulates the original combinatorial optimization into a continuous constrained optimization, allowing to employ non-linear optimization methods to perform BN structure learning. While avoiding the combinatorial search, the method encounters difficulties associated with non-linear non-convex optimization. In addition, the method is slow in convergence and requires a thresholding operation to produce the final structure.

52. Page 84: ~~data~~ $\Rightarrow$ variables

53. Page 84: The necessary and sufficient condition for the presence of a link between variables X_i and X_j is: $X_i \not\perp X_j|S, \forall S \subset \mathbf{X} \setminus X_i \setminus X_j$. The necessary and sufficient condition for the absence of a link between variables X_i and X_j is $X_i \perp X_j|S, \exists S \subset \mathbf{X} \setminus X_i \setminus X_j$.

54. Page 86: ~~Bayesian~~ $\Rightarrow$ the MAP

55. Page 89: $w_{m,c} \times I((\mathbf{y}^m, \mathbf{z}^m) = njk)$

56. Page 89: $w_{m,c} = p(\mathbf{z}^m = c|\mathbf{y}^m, \boldsymbol{\theta}^{t-1})$, where $\mathbf{z}^m = c$ means $\mathbf{z}^m$ takes it's cth configuration. $I((\mathbf{y}^m, \mathbf{z}^m) = njk)$ is the indicator function that equals to 1 when it's argument is true, i.e., node n takes njk configuration, with $x_n = k$ and its parents taking on the jth configuration.

57. Page 90: ~~stochastic~~ $\Rightarrow$ Monte Carlo

58. Page 90: ~~stochastic EM~~ $\Rightarrow$ Monte Carlo EM [8]

59. Page 91: ~~stochastic~~ $\Rightarrow$ Monte Carlo

60. Page 91-93: ~~Bayesian~~ $\Rightarrow$ Maximum Posterior Probability, ~~Bayesian~~ $\Rightarrow$ MAP

61. Page 94: add [ ]

62. Page 96: $\mathcal{G}^0$ captures $p(\mathbf{X}^0)$, i.e., the joint distribution of $\mathbf{X}^0$, while $\overrightarrow{\mathcal{G}}$ captures the conditional joint distributions of $\overrightarrow{\mathbf{X}}^t$, i.e., $p(\overrightarrow{\mathbf{X}}^t|\overrightarrow{\mathbf{X}}^{t-1})$.

63. Page 97: ~~all~~ $\Rightarrow$ the $\overrightarrow{\mathbf{X}}^t$, ~~non-stationary~~ $\Rightarrow$ time variant ,via the DBN chain rule

64. Page 98: ~~Bayesian~~ $\Rightarrow$ MAP estimation,

65. Page 98: ~~$\mathbf{D} = \{S_1, S_2, ..., S_M\}$~~ $\Rightarrow \mathbf{D} = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_M\}$, ~~a~~ $\Rightarrow$ the m th

66. Page 98: ~~$\mathbf{S}_m = \{S_{m,0}, S_{m,1}, ..., S_{m,t_m}\}$~~ $\Rightarrow \mathbf{S}_m = \{\mathbf{X}^{m,0}, \mathbf{X}^{m,1}, ..., \mathbf{X}^{m,t_m}\}$

67. Page 98: after applying DBN chain rule

68. Page 98: ~~$\{S_{m,0}\}$~~ $\Rightarrow \{\mathbf{X}^{0,m}\}$,~~$\{S_{m,t-1}, S_{m,t}\}$~~ $\Rightarrow \{\mathbf{X}^{m,t-1}, \mathbf{X}^{m,t}\}$, ~~$\{S_{m,t-1}, S_{m,t}\}$~~ $\Rightarrow \{\mathbf{X}^{m,t-1}, \mathbf{X}^{m,t}\}$

69. Page 99: Specifically, for the EM method, the E-step must estimate the expected counts for the prior and transition network jointly. In the M-step however, the parameters and structures for the prior and transition network can be computed separately from the expected counts.

70. Page 99: The former is called observation independence assumption, while the latter is called state independence assumption, probability, Particle filtering results from replacing the integral in Eq. 3.131 with the average of samples acquired from sampling x^{t-1} from the filtering probability at time $t-1$.

71. Page 100: Besides the recursive methods discussed above, we can also employ the direct method. The direct method simply unrolls the DBN to T time slices to form an expanded BN and then performs DBN inferences using the exact or approximate BN inference methods. Exact methods include the junction tree method and the approximate methods include MCMC sampling and variatiaonl methods.

72. Page 100: ~~the transition network or the unrolled DBN network.~~ $\Rightarrow$ both the direct and recursive methods.

73. Page 102: $, Y^{t-1}, Y^{t-2}, .., \ p(X^0)p(Y^0|X^0)\prod_{t=1}^{T} p(Y^t|X^t)p(X^t|X^{t-1})$, i.e., the most likely state sequence,

74. Page 102: Like the DBN inference, HMM inference methods include the direct approach and the recursive approach. The direct approach unrolls the HMM into an expended BN of T time slices. Exact or approximate BN inference methods can then be applied to perform both likelihood and MAP inferences. For example, max-product belief propagation can be used to perform HMM decoding inference.

75. Page 102: ~~both types of inference~~ $\Rightarrow$ direct methods, based on recursive computation , HMM's underlying state and observation independence assumptions to recursively perform the inference, ~~efficiently~~ $\Rightarrow$ recursively

76. Page 104: ~~$p(\mathbf{y}^{1:T})$~~ $\Rightarrow p(\mathbf{y}^{1:T}|\mathbf{\Theta})$, repeat 3 times.

77. Page 105: Appendix 3.9.7 introduces a standard EM algorithm for HMM learning.

78. Page 106: ~~x^0~~ $\Rightarrow x_m^0$, ~~y^0~~ $\Rightarrow y_m^0$, ~~$\mathbf{x}$~~ $\Rightarrow \mathbf{x}_m$

79. Page 113: and $\mathbf{\Theta}$ are latent variables.

80. Page 113: ~~Bayes rule~~ $\Rightarrow$ conditional chain rule, ~~decomposed~~ $\Rightarrow$ factorized, the product of

81. Page 113: , i.e. $p(\mathbf{X}, \mathbf{\Theta}|\boldsymbol{\alpha}) = p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta}|\boldsymbol{\alpha})$

82. Page 114: $p(\mathbf{X}, \mathbf{Y}, \boldsymbol{\theta}|\boldsymbol{\alpha}) = p(\mathbf{X}, \mathbf{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})$, due to the presence of the latent variables $\boldsymbol{\theta}$.

83. Page 114: ~~$\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^M\}$~~ $\Rightarrow \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N, \mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^N\}$

84. Page 117: ~~$p(\theta|\mathcal{D})$~~ $\Rightarrow p(\theta|\mathcal{D}, \boldsymbol{\alpha}^{s_a})$

85. Page 117: Empirical Bayes inference is an approximation to the full Bayesian inference. It's a good approximation when the posterior probability of $\boldsymbol{\theta}$ is sharply peaked. When training data is insufficient or imbalanced, full Bayesian inference produces better inference results.

86. Page 118: that captures $p(X, Y)$, that captures $p(X, Z_1, .., Z_n, Y)$, where $Z_1, Z_2.., Z_n$ are latent variables.

87. Page 118: They can be treated as hidden BNs, with latent variables $Z_1, Z_2, ..., Z_n$, and can

88. Page 120: computationally

89. Page 120: Compared with the deterministic deep models such as Convolutional Neural Networks (CNNs), deep probabilistic graphical models have the following advantages: 1) for classification/regression tasks, they can not only produce a prediction but can also produce a probability distribution in a principle manner such that the prediction uncertainty/confidence can be quantified. CNNs cannot effectively capture their prediction uncertainty. Recent developments in Bayesian neural networks are aimed at addressing this problem with CNNs. Second, as generative models, they can perform both discriminative task (e.g. classification) and generative tasks such as data generation, while CNNs can only perform classifications, and the Generative Adversarial Networks (GANs) are used to perform data generation. Thirdly, they can perform both data generation and classification under incomplete input data, which both CNNs/GANs cannot. The main limitation with deep probabilistic graphical models are their computational complexity in both learning and inference, which prevents them from scaling up to large data/models.

90. Page 121: ~~using the outputs from the previous layer as inputs, gradient ascent~~ $\Rightarrow$ the direct, in plate notation, w_{ij}, the j-th word in i-th document,

91. Page 122: The model captures $p(X, Z^1, Z^2, \Theta^1, \Theta^2 | \alpha^1, \alpha^2)$, where $Z^1, Z^2, \Theta^1, \Theta^2, \alpha^1$, and α^2 are all latent variables.

92. Page 122: ~~z~~ $\Rightarrow z_{ij}$, the latent topic that word w_{ij} belongs to, $\theta \Rightarrow \theta_i$, the parameters that specify the topic distribution for document i

93. Page 122: The model captures the joint distribution of w_{ij}, z_{ij}, θ_i, and following the BN chain rule, $p(w_{ij}, z_{ij}, \theta_i | \alpha, \beta) = p(w_{ij}|z_{ij}, \beta)p(z_{ij}|\theta_i)p(\theta_i|\alpha)$, where θ_i and z_{ij} are latent variables.

94. Page 122: It may also be used to infer the most likely topic distribution for a given document i, i.e., $\theta_i^* = \arg\max_{\theta_i} p(\theta_i | \mathbf{w}, \alpha, \beta)$, based on which document i may be classified into one of the document categories

4 Chapter 4 Corrections

1. Page 133: ~~is~~ $\Rightarrow$ are

2. Page 134: This property and it's complement can be employed to perform constraint-based MN structure learning.

3. Page 135: ~~$\phi(x_i)$~~ $\Rightarrow \phi_i(x_i)$

4. Page 136: ~~the conditional random field of $\mathbf{X}$ given $\mathbf{Y}$, quantified by the conditional distribution $p(\mathbf{X}|\mathbf{Y})$~~ $\Rightarrow$ joint distribution of $\mathbf{X}$ and $\mathbf{Y}$, i.e., $p(\mathbf{X}, \mathbf{Y})$

5. Page 137: ~~$p(x_1, x_2, ..., x_N | y_1, y_2, .., y_N)$~~ $\Rightarrow p(x_1, x_2, ..., x_N, y_1, y_2, .., y_N)$

6. Page 139: ~~ℓ_2~~ $\Rightarrow \ell_1$

7. Page 141: that captures $p(\mathbf{X}|\mathbf{Y})$

8. Page 141: that captures $p(\mathbf{X}, \mathbf{Y})$

9. Page 142: ~~Although both CRF and label-observation MRFs capture the conditional label distributions,~~ $\Rightarrow$ Because label-observation MRFs capture the joint distribution of labels and observations, while CRFs capture the conditional distribution of labels given observations,

10. Page 143: Instead of pairwise energy function, high order energy function can be used to define N-wise (N>2) energy functions.

11. Page 145: ~~m_{ji}~~ $\Rightarrow m_{ji}(x_i)$, ~~m_{kj}~~ $\Rightarrow m_{kj}(x_j)$, ~~X_i~~ $\Rightarrow x_i$, ~~m_{ji}~~ $\Rightarrow m_{ji}(x_i)$

12. Page 145: ~~follow the same trace back procedure as the MAP-variable elimination in Algorithm 3.2~~

13. Page 145: Different from BN inference, which is NP hard in general, inference for pairwise MRF with BP is linear to the number of edges and quadratic to the number of states for each variable. Inference for high order MRF, however, becomes NP hard. Examples of belief propagation for MRF can be found in Appendix 4.8.1.

14. Page 146: ~~then follow the trace back procedure to~~

15. Page 146: using it's maximum marginal probability.

16. Page 146: In summary, the message passing equations for both posterior and MAP inferences in junction tree are essentially the same as those for MN inferences, with the only difference being that both messages passing and belief updating are with respect to the cluster nodes instead of individual variable nodes as for MNs.

    Finally, like ancestral sampling for BN posterior inference, Monte Carlo (MC) sampling can also apply to MN to perform exact posterior inference. But MC sampling for MN is not as simple as BN since we do not have a topological order to follow during sampling. We can randomly select one node as the root node and sample it's unary. This is then followed by sampling other nodes that are adjacent to the node using their pairwise functions. This can continue until we have samples for all nodes.

17. Page 146: Find a set of X labels to swap using a min cut/max flow algorithm from network theory such that the flow from a source node s to a sink node t is maximized. Initial s and t are manually identified.

18. Page 147: ~~conditional~~ $\Rightarrow$ pseudo-likelihood

19. Page 149: ~~,~~$\mathbf{y}$, efficiently via $p(x_i|N_{x_i}) = \frac{\exp(-\alpha_i E_i(x_i) - \sum_{x_j \in N_{x_i}} w_{ij} E(x_i, x_j|))}{\sum_{x_i} \exp(-\alpha_i E_i(x_i) - \sum_{x_j \in N_{x_i}} w_{ij} E(x_i, x_j))}$

20. Page 149-150: ~~The only difference is replacing the sum operation with the maximum operation in computing the message each node sends to its neighbors. After convergence, we can then follow the trace back procedure discussed for the MAP variable elimination algorithm to identify theMAP assignment for each node.~~

21. Page 157: ~~Bayesian~~ ⇒ Maximum A Posteriori (MAP), ~~Bayesian~~ ⇒ MAP

22. Page 158: ~~Bayesian~~ ⇒ MAP

23. Page 159: the direct method that, ~~maximizing~~ ⇒ the EM method that maximizes, ~~gradient~~ ⇒ direct

24. Page 160: Unlike the BIC score for BN, the BIC score for MN is not decomposible due to the partition function. Hence, the structure for each node cannot be learned separately.

25. Page 160: The likelihood term in the BIC score can be approximated by the pseudo-likelihood, yielding the pseudo-BIC score, which can be computed more efficiently.

26. Page 161: ~~The~~ ⇒ But because of the non-decomposible score function, approximately

27. Page 161: Hence, a better approach may be hybrid, whereby a local method is used to learn an initial structure and a global method can be used to refine the initial structure.

28. Page 161: The basic idea is to assume the structure is fully connected initially and then begin to prune the links through learning, η

29. Page 161: This assumption holds if we assume the MN follows Gibbs distribution, with a log-linear pairwise potential function.

30. Page : Finally, we can also employ the constraint based approach by performing independence tests. Following the pairwise independence property, the necessary and sufficient condition for the absence of a link between two nodes is that they are independent, given all other nodes. Alternatively, we can use the complement of the pairwise independence property to ascertain the presence of a link between the two nodes, i.e., there exists a link between two nodes if and only if they are dependent, given any subset (including empty set) of the remaining nodes.

31. Page 162: In fact, unlike the BN parameter learning, which is decomposable and has closed form solutions, there is no closed form solution to MN parameter learning and learning is not decomposable.

32. Page 179: ~~casual~~ ⇒ causal

33. Page 262: ~~casual~~ ⇒ causal