APMA 1740/2610 2025: Homework 5

1. **Softmax: a tale of two loss functions.** There are two natural ways to measure the performance of a softmax classifier relative to the training data: the likelihood (corresponding to the cross-entropy loss) and the error rate. This problem explores the connections between the cross-entropy and error-rate loss functions in the context of the softmax model.

   Consider a classification problem with feature vectors $x \in \mathbb{R}^d$ and categories $y \in \{1, 2, \ldots, s\}$. Given training data (x_k, y_k), $k = 1 : n$, suppose that we adopt the softmax model, i.e.

$$r_c(x; \beta_1, \ldots, \beta_s) = \mathbb{P}\left(Y = c \middle| x; \beta_1, \ldots, \beta_s\right) = \frac{e^{\beta_c \cdot x}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x}} \quad \forall\, c = 1 : s$$

   To avoid over-parametrization, we define $\beta_1 \triangleq 0$.

   Consider the Bayes (minimum-error) classifier:

$$h(x; \beta_2, \ldots, \beta_s) = \text{argmax}_c \mathbb{P}\left(Y = c \middle| x; \beta_2, \ldots, \beta_s\right)$$

   Assume that the data is consistent, i.e. $x_k = x_l \Rightarrow y_k = y_l$, and that every category is represented, i.e. for every c there exists a k such that $y_k = c$.

   (a) Show that the error rate on the training data is zero (i.e. $h(x_k) = y_k$ for all $k = 1 : n$) if and only if, for every $k = 1 : n$,

$$(\beta_{y_k} - \beta_c) \cdot x_k > 0 \quad \text{for all } c \neq y_k$$

   ($\Longrightarrow$). By assumption,

$$\begin{aligned} h(x_k; \beta_2, \ldots, \beta_s) &= \arg\max_c \mathbb{P}(Y = c \mid x_k; \beta_2, \ldots, \beta_s) \\ &= \arg\max_c r_c(x_k; \beta_1, \ldots, \beta_s) \\ &= \arg\max_c \frac{e^{\beta_c \cdot x_k}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x_k}} = y_k \end{aligned}$$

   But this means that for any $c \neq y_k$,

$$\frac{e^{\beta_c \cdot x_k}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x_k}} < \frac{e^{\beta_{y_k} \cdot x_k}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x_k}} \implies \beta_c \cdot x_k < \beta_{y_k} \cdot x_k \implies (\beta_{y_k} - \beta_c) \cdot x_k > 0$$

   Note: we have $<$ instead of $\leq$ since $c \neq y_k$ by assumption and r_c is strictly monotonic in β_c.

   ($\Longleftarrow$). Suppose $\forall k = 1 : n$, $(\beta_{y_k} - \beta_c) \cdot x_k > 0$ for all $c \neq y_k$. Then, we have

$$\begin{aligned} \beta_{y_k} \cdot x_k &> \beta_c \cdot x_k \\ e^{\beta_{y_k} \cdot x_k} &> e^{\beta_c \cdot x_k} \\ \frac{e^{\beta_{y_k} \cdot x_k}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x_k}} &> \frac{e^{\beta_c \cdot x_k}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x_k}} \\ r_{y_k}(x_k; \beta_1, \ldots, \beta_s) &> r_c(x_k; \beta_1, \ldots, \beta_s) \qquad \forall y_k \neq c \end{aligned}$$

   hence, $y_k = \arg\max_c r_c(x_k; \beta_1, \ldots, \beta_s) = h(x_k; \beta_1, \ldots, \beta_s)$.

   (b) For every $c \in \{2, \ldots, s\}$, let $\beta_c(t)$ $t = 1, 2, \ldots$ be a sequence of vectors in $\mathbb{R}^d$, such that the likelihood

$$L = L(\{x_k, y_k\}, k = 1 : n; \beta_1(t), \ldots, \beta_s(t)) = \prod_{k=1}^{n} r_{y_k}(x_k; \beta_1(t), \ldots, \beta_s(t)) \to 1 \quad \text{as } t \to \infty$$

   Show that $\|\beta_c(t)\| \to \infty$ for every $c = 2 : s$.

By assumption,

$$L = \prod_{k=1}^{n} r_{y_k}(x_k; \beta_1(t), \ldots, \beta_s(t)) \to 1$$

$$r_{y_k}(x_k; \beta_1(t), \ldots, \beta_s(t)) \to 1$$

$$\frac{e^{\beta_{y_k}(t)\cdot x_k}}{1 + \sum_{\tilde{c}=2}^{s} e^{\beta_{\tilde{c}}(t)\cdot x_k}} \to 1$$

$$e^{\beta_{y_k}(t)\cdot x_k} \to 1 + \sum_{\tilde{c}=2}^{s} e^{\beta_{\tilde{c}}(t)\cdot x_k}$$

$$\sum_{\tilde{c}\neq y_k}^{s} e^{\beta_{\tilde{c}}(t)\cdot x_k} \to -1$$

However, for fixed x_k, $e^{\beta_{\tilde{c}}(t)\cdot x_k} \geq 0$ for all β_c and this is impossible. Hence, we must have that the sum does not converge, i.e. $\lim_{t\to\infty} \beta_{\tilde{c}}(t) \cdot x_k \geq -1$ (by p-series) for all possible values of x_k. In particular, this forces $\beta \to \infty$.

2. **Gradient descent and the maximum-margin classifier.** DNN classifiers are routinely trained with the number of parameters exceeding the number of samples by many orders of magnitude. The implicit function theorem indicates that any performance on the training set can be reproduced, exactly, by any of an infinite collection of parameter vectors—namely, the ones lying on a smooth high-dimensional manifold. Presumably, much of the area of the manifold is taken up by parameter vectors that would neither interpolate nor extrapolate to good out-of-sample performance. By what mechanisms do DNN classifiers avoid choosing poorly performing parameters?

   One suggestion, which is backed by some theoretical as well as experimental results, is that the training of certain loss functions via gradient descent can lead to models that extend smoothly beyond the training data, thereby producing a kind of implicit regularization.[1]

   The purpose of this problem is to explore a particularly well understood example of this behavior, following a result by Soudry et al., on "The Implicit Bias of Gradient Descent on Separable Data" (JMLR 19, 2018, 1-57). Soudry et al. have shown that gradient ascent of the log-likelihood function for the softmax model will converge to the maximum-margin classifier when trained on linearly separable data. The result is relevant since (i) we can always engineer an unambiguous training set to be linearly separable, and (ii) empirical results on DNN classifiers indicate that the highest layers will often feed the final softmax classifier with linearly separable data (e.g. Papyan et al., PNAS, V117, #40, 2020).

   Here, we will implement a particular example: Consider the two-category classification problem under the softmax (aka logit) model:

$$r_1(x;\beta) = \mathbb{P}\left(Y=1\middle|x;\beta\right) = \frac{e^{\beta\cdot x}}{e^{\beta\cdot x}+e^{-\beta\cdot x}} = \frac{1}{1+e^{-2\beta\cdot x}}$$
$$r_{-1}(x;\beta) = \mathbb{P}\left(Y=-1\middle|x;\beta\right) = \frac{e^{-\beta\cdot x}}{e^{\beta\cdot x}+e^{-\beta\cdot x}} = \frac{1}{1+e^{2\beta\cdot x}}$$

   where $x=(1,x_1,x_2)$, $y\in\{-1,1\}$, and $\beta=\beta=(\beta_1,\beta_2,\beta_3)$ are the desired parameters.[2] Although β has three components, the classifier is really just a function of two variables, $(x_1,x_2)\in\mathbb{R}^2$.

   Given β, the classifier is the simple MAP (maximum *a posteriori*) classifier—it chooses the most likely category:

$$h(x) = \begin{cases} 1 & \text{if } r_1(x) > r_{-1}(x) \\ -1 & \text{if } r_{-1}(x) > r_1(x) \end{cases}$$

   The training data, $\{(x(i),y(i))\}_{x=1:n}$, is assumed to be linearly separable. In other words, we assume that there exists a $\beta\in\mathbb{R}^3$ such that

$$y(i) = sign\left(\beta\cdot x(i)\right) = sign\big(\beta_1+\beta_2x_1(i)+\beta_3x_2(i)\big) \qquad \text{for all } i=1:n$$

   Let $\mathcal{N}_+=\{i=1:n\big|y(i)=1\}$ and $\mathcal{N}_-=\{i=1:n\big|y(i)=-1\}$. Then the likelihood L can be written as

$$L = L\left(\left\{\big(x(i),y(i)\big)\right\}_{x=1:n};\beta\right) = \prod_{i\in\mathcal{N}_+} r_1(x(i);\beta)\prod_{i\in\mathcal{N}_-} r_{-1}(x(i);\beta)$$

   As mentioned, if the data is linearly separable, then by the result of Sourdry et al., learning the maximum-likelihood classifier using gradient ascent of the log-likelihood should lead to the maximum-margin classifier.

   (a) Show that the gradient, ∇_β, of the log-likelihood, $\log(L\left(\left\{\big(x(i),y(i)\big)\right\}_{x=1:n};\beta\right))$, can be written as

$$\sum_{i\in\mathcal{N}_+}\frac{2}{1+e^{2\beta\cdot x(i)}}x(i) - \sum_{i\in\mathcal{N}_-}\frac{2}{1+e^{-2\beta\cdot x(i)}}x(i) \tag{1}$$

   (This representation is particularly robust to overflow and under flow of the exponentials.) And, therefore, discrete gradient ascent of $\frac{1}{n}\log(L\left(\left\{\big(x(i),y(i)\big)\right\}_{x=1:n};\beta\right))$, can be achieved using the recursion

$$\beta_t = \beta_{t-1} + \epsilon\frac{2}{n}\Big(\sum_{i\in\mathcal{N}_+}\frac{1}{1+e^{2\beta_{t-1}\cdot x(i)}}x(i) - \sum_{i\in\mathcal{N}_-}\frac{1}{1+e^{-2\beta_{t-1}\cdot x(i)}}x(i)\Big)$$

[1] Here are some good starter references on the connections between gradient descent and implicit regularization:

(i) Evidence from Facebook Research that stacking matrices (i.e. layers of fully-connected linear units) in the "bottleneck" of an autoencoder improves performance: (Jing et al.) arXiv:2010.00679v2 [cs.LG] 14 Oct 2020

(ii) A relatively clean special case, and a relatively complete analysis: (Chou et al.) arXiv:2011.13772v3 [cs.LG] 7 Apr 2021

(iii) Empirical and theoretical analysis of implicit regularization in deep *linear* networks: (Aurora et al.) arXiv:1905.13655v3 [cs.LG] 26 Oct 2019

[2] There are many equivalent formulations, but for the two-category problem this one is particularly convenient and also cleaner to code.

following some initialization $\beta_1 = \beta_o$.

$$\begin{aligned}
\nabla_\beta \log L(\{x_i, y_i\}_{1:n}; \beta) &= \nabla_\beta \left(\log \left[\prod_{i \in \mathcal{N}_+} r_1(x(i); \beta) \prod_{i \in \mathcal{N}_-} r_{-1}(x(i); \beta) \right] \right) \\
&= \nabla_\beta \left(\sum_{i \in \mathcal{N}_+} \log r_1(x(i); \beta) + \sum_{i \in \mathcal{N}_-} \log r_{-1}(x(i); \beta) \right) \\
&= \nabla_\beta \left(\sum_{i \in \mathcal{N}_+} \log \frac{1}{1 + e^{-2\beta \cdot x}} + \sum_{i \in \mathcal{N}_-} \log \frac{1}{1 + e^{2\beta \cdot x}} \right) \\
&= \sum_{i \in \mathcal{N}_+} \nabla_\beta \left(-\log(1 + e^{-2\beta \cdot x}) \right) + \sum_{i \in \mathcal{N}_-} \nabla_\beta \left(-\log(1 + e^{2\beta \cdot x}) \right) \\
&= \sum_{i \in \mathcal{N}_+} \frac{2}{1 + e^{2\beta \cdot x(i)}} x(i) - \sum_{i \in \mathcal{N}_-} \frac{2}{1 + e^{-2\beta \cdot x(i)}} x(i)
\end{aligned}$$

(b) (Do not submit) The data consists of two arrays, one ('XSampPos') contains the locations in $\mathbb{R}^2$ of the feature vectors labelled $y = +1$, and the other ('XSampNeg') contains the feature vectors labelled $y = -1$. As explained above, each feature vector is of the form $x = (1, x_1, x_2)$, so the second and third coordinates can be visualized as a point in the plane. Both arrays can be found in the file 'SoftMax.mat'. By design, this data is linearly separable.

(i) Load and display the data as a scatter plot of $(x_1(i), x_2(i)) \in \mathbb{R}^2$, using separate colors for the two categories.[3] Choose an arbitrary starting value for β. (for example, use `beta = [4,-1,1];` because this made for an initial decision boundary that was very much in the *wrong* direction.) Now plot the initial decision boundary, on the same graph that displays the training data.

(ii) Run the gradient ascent until the classifier appears to be reasonably close to the maximum margin classifier. (e.g., 10,000 iterations with $\epsilon = 0.2$. You can raise ϵ and lower the number of iterations to make it faster.) Make another scatter plot of the data and superimpose the last computed decision boundary.

(iii) Plot the value of the log of the likelihood as a function of the number of iterations.

(iv) Plot the magnitude (length) of β as a function of the number of iterations.

(v) At each iteration, compute the distances to the closest two feature vectors with labels +1 and to the closest two with labels -1. Then plot these four distances as a function of the number of iterations.

(c) In light of Soudry's result and the results of problem (1):

(i) As $t \to \infty$, what should be the limit of the log likelihood? Is this consistent with your results?
0, since we expect the likelihood to converge to 1 (as in Problem 1), which is exactly what we see in the plot.

(ii) What will be the limit of the magnitude of β?
$\beta / ||\beta||$ converges to the maximum margin classifier. Since we want the margin to be as large as possible, we want $||\beta|| \to \infty$ and this is what we see in the plot.

(iii) What, if anything, can you say about the limits of the four distances? (For each of these, be sure to explain your reasoning.)
The distance of the closest feature vectors should converge to the margin of the maximum margin classifier. Hence, we expect the distances to converge to the same value. This is what we see in the plot.

(d) Use problem (1) and the expression for the gradient (equation 1) to argue that only the support vectors will be relevant in determining the limiting classifier.

Only the support vectors will be relevant because, by our previous work, for all x_k not a support vector, $\beta(t) \cdot x_k \to \infty$. Meanwhile, for the support vectors, we have $\beta(t) \cdot x_k \to 1$. In the denominator of a sum, all other terms vanish.

[3] In Matlab, you can use `load('SoftMax.mat','XSampPos','XSampNeg')`.

3. Consider the following joint pmf for the random variables $X_1, \ldots, X_6$ over $\mathbb{Z}^6$, where $\mathbb{Z}$ is the integers.

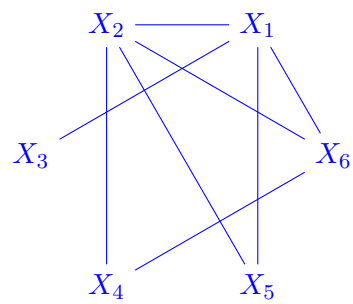
$$f(x_1, \ldots, x_6) \triangleq \beta\big(x_1 x_2\big)^2 \big(\sin(x_2 + x_5)\big)^4 \big(\cos(x_1 e^{x_3})\big)^2 \big(x_2 + x_4 + x_6\big)^{10} e^{-\sum_{i=1}^{6} x_i^2 - |x_1 x_6| - |x_1 x_5|}$$

where β is a normalization constant. Use the marginalization and conditioning rules to show that X_4 and X_6 are conditionally independent from X_3 and X_5 given X_1 and X_2.

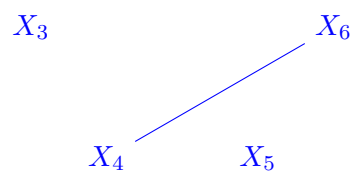
It suffices to show that $f(x_3, x_4, x_5, x_6 \mid x_1, x_2)$ can be factored in the form

$$f(x_3, x_4, x_5, x_6 \mid x_1, x_2) = \frac{1}{Z} \phi_{46}(x_4, x_6) \phi_{35}(x_3, x_5)$$

Notice that we can draw the graph that f respects



Conditioning on X_1 and X_2 using the theorems from class gives us the subgraph



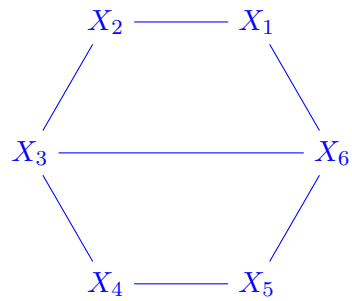
with disjoint components $\{X_3\}, \{X_5\}, \{X_4, X_6\}$. By a theorem from class, we know that this implies that X_4 and X_6 are conditionally independent from X_3 and X_5 given X_1 and X_2.

4. Suppose that $X = (X_1, \ldots, X_6)$ respects the graph $G = (V, E)$ given by $V = \{X_1, \ldots, X_6\}$ and

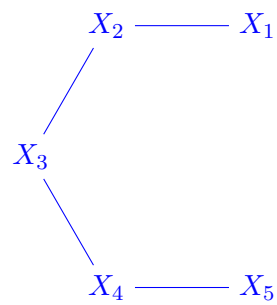
$$E = \{(X_1, X_2), (X_2, X_3), (X_3, X_4), (X_4, X_5), (X_5, X_6), (X_1, X_6), (X_3, X_6)\}.$$

(a) Use the marginalization and conditioning rules to show that the conditional distribution of $X_1, \ldots, X_5$ given $X_6 = x_6$ is a Markov chain, meaning that it respects the nearest-neighbor linear graph.

We can draw the graph as follows:

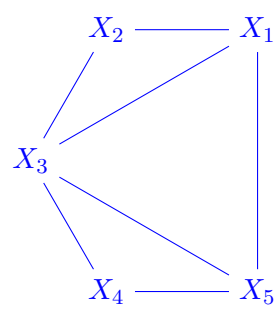


Conditioned on X_6, the distribution of $X_1, \ldots, X_5$ is given by the subgraph $G \cap \{X_6\}^c$:



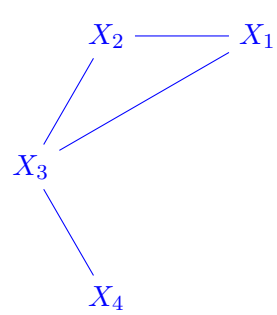
which is just the nearest-neighbor linear graph.

(b) What is the simplest graph that $X_1, \ldots, X_5$ is guaranteed to respect?[4]



(c) Is the conditional distribution of $X_1, \ldots, X_4$ given $X_5 = x_5$ necessarily a Markov chain? In other words, does $X_{1:4}$, given X_5, necessarily respect the nearest-neighbor linear graph?

First, we need to marginalize X_6 out of the graph and then condition on X_5. The marginalized graph is precisely the simplest graph from part (b). Hence, we can draw the graph $(X_1, X_2, X_3, X_4 \mid X_5)$ as



[4] A graph G' would be "guaranteed" if it were respected by $X_1, X_2, \ldots, X_5$ for *every* distribution on $X_1, X_2, \ldots, X_6$ which respects G. By "simplest," there should be no graph with fewer edges that is also guaranteed to be respected.

While the nearest neighbor linear graph is indeed a subgraph of the above, we do not have a guarantee that this is a Markov chain for every possible distribution on (X_1, X_2, X_3, X_4).

5. Consider the following joint pmf for the random variables U, V, W, X, Y, Z, each taking values in the finite set $\{1, \ldots, L\}$:

$$f(u, v, w, x, y, z) = \phi_1(u, v)\phi_2(u, v, w)\phi_3(w, x, y)\phi_4(x, z)\phi_5(x, y, z)\phi_6(u, z)$$

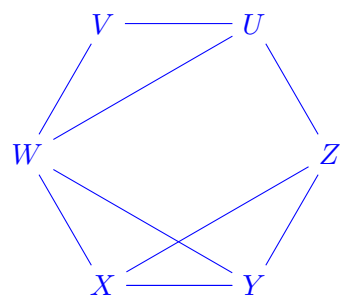
where $\phi_1, \ldots, \phi_6$ are each nonnegative functions.

(a) Do these variables respect the complete graph over 6 nodes? Why or why not?

Any distribution $p(x)$ for $x \in \mathbb{R}^n$ will be a GRF on the complete graph K_n. Hence, this distribution respects the complete graph.

(b) Draw the simplest possible graph these variables are guaranteed to respect.

We can draw the graph as follows:



Trivially, if we were to remove any edge from the graph, we would not respect the maximum cliques of the pmf.

(c) Are U and X necessarily independent? Could they be independent?

No. Lack of an edge does not imply independence. Consider the case $Z \sim \text{Bernoulli}(1/2)$ and

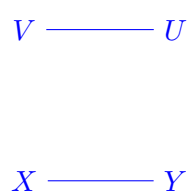
$$\begin{cases} U, X \sim \text{Bernoulli}(0.01) & \text{if } Z = 1 \\ U, X \sim \text{Bernoulli}(0.99) & \text{if } Z = 0 \end{cases}$$

Then U and X do not share an edge, but they are also not independent.

However, we could also imagine a case where $U, X \overset{\text{iid}}{\sim} p$ and $Z = U + X$ in which case they would be independent.

(d) Find a collection of variables (from the set V, W, Y, Z) such that U and X are necessarily conditionally independent given the values of these variables. Your collection should have the fewest number of variables possible. Justfy your choice or prove that no such collection necessarily exists.

We can draw $(U, V, X, Y \mid W, Z)$ as



in which case U and X are independent.

In any case where we were to remove only one vertex, we would still have a path between U and X in the subgraph. Hence, to guarantee independence, we need to remove at least two vertices.

6. **Proof of the conditioning rule for Gibbs random fields.** Let $X = (X_1, \ldots, X_d)$ be a random vector over a discrete space. Suppose that X respects the graph G. Assume that $\mathbb{P}(X_{A^c} = x_{A^c}) > 0$. Prove that the conditional distribution on X_A given $X_{A^c} = x_{A^c}$ (i.e. $X_A | X_{A^c} = x_{A^c}$) respects the subgraph of G over the vertices in A.

By a slight abuse of notation,

$$\begin{aligned} f(x_A \mid x_{A^c}) &= \frac{f(x_A, x_{A^c})}{f(x_{A^c})} \\ &= \frac{f(x)}{f(x_{A^c})} \\ &= \frac{\frac{1}{Z} \prod_{i \subset A} \phi_i(x_i) \prod_{\substack{j \subset G \\ j \not\subset A^c, j \not\subset A}} \phi_j(x_j)}{\frac{1}{Z'} \prod_{k \subset A^c} \phi_k(x_k)} \end{aligned}$$

However, since A^c is fixed, each $\phi_j(x_j)$ will be only a function of $x \in A$, hence included already in the first product. Therefore, we can write

$$f(x_A \mid x_{A^c}) = \frac{\frac{1}{Z} \prod_{i \subset A} \phi_i(x_i) \prod_{j \subset A^c} \phi_j(x_j)}{\frac{1}{Z'} \prod_{k \subset A^c} \phi_k(x_k)} = \frac{1}{Z''} \prod_{i \subset A} \phi_i(x_i)$$

which respects the graph $G|_A$.

For 2610 or for extra credit:

7. **Two formulations of the maximum-margin problem.**

Let $x_1, \ldots, x_n \in \mathbb{R}^d$ and let $y_1, \ldots, y_n \in \{-1, 1\}$. Assume that $\{(x_i, y_i)\}_{i=1:n}$ is consistent (i.e. $x_i = x_j \Rightarrow y_i = y_j$) and linearly separable. (Any pair, $\alpha \in \mathbb{R}^1$ and $\beta \in \mathbb{R}^d$, defines a classifier, $h(x)$: $h(x) \triangleq \text{sign}\,(\alpha + \beta \cdot x)$.)

The goal is to show that the following two formulations of the problem of finding a maximum margin classifier were equivalent:

(i) Find $\widehat{\alpha}$, $\widehat{\beta}$, and $\widehat{M}$ to solve

$$\text{argmax}_{M,\alpha,\beta:\|\beta\|=1} M \qquad \text{subject to } \; y_k\,(\alpha + \beta \cdot x_k) \geq M \quad \forall\, k = 1:n$$

(ii) Find $\widehat{\alpha}$ and $\widehat{\beta}$ to solve

$$\arg\min_{\alpha,\beta} \frac{1}{2} \sum_{i=1}^{d} \beta^2 \qquad \text{subject to } \; y_k\,(\alpha + \beta \cdot x_k) \geq 1 \quad \forall\, k = 1:n$$

Construct a transformation from a solution to (i) into a solution to (ii), and another transformation from a solution to (ii) into a solution to (i).

8. **Proof of the marginalizing rule for Gibbs random fields.** Let $X = (X_1, \ldots, X_d)$ be a random vector over a discrete space. Suppose that X respects the graph G. Prove that X_A respects the graph G' over the vertices in A, where i and j are connected in G' if they are directly connected in G or if there is a path from i and j completely within A^c in G.

9. **Equivalence of two-sided and one-sided Markov properties for positive pmfs.** Let $X = (X_1, \ldots, X_d)$ be a random vector over a discrete space. The familiar "one-sided Markov property" is

$$\mathbb{P}(X_t = x_t | X_{1:t-1} = x_{1:t-1}) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1})$$

for all x and each $t = 1, \ldots, d$.[5] The "two-sided Markov property" is

$$\mathbb{P}(X_t = x_t | X_{1:t-1} = x_{1:t-1}, X_{t+1:d} = x_{t+1:d}) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, X_{t+1} = x_{t+1})$$

for all x and each $t = 1, \ldots, d$.[6] Prove that these two properties are equivalent as long as X has a strictly positive pmf. You can assume the independence, conditioning, and marginalizing rules for GRFs, and can use the Hammersley-Clifford theorem. (It is also instructive to try to prove it without the Hammersley-Clifford theorem. It is surprisingly hard.)

[5] Interpreted as $\mathbb{P}(X_1 = x_1) = \mathbb{P}(X_1 = x_1)$, when $t = 1$.

[6] Interpreted as $\mathbb{P}(X_1 = x_1 | X_{2:d} = x_{2:d}) = \mathbb{P}(X_1 = x_1 | X_2 = x_2)$ when $t = 1$, and $\mathbb{P}(X_d = x_d | X_{1:d-1} = x_{1:d-1}) = \mathbb{P}(X_d = x_d | X_{d-1} = x_{d-1})$ when $t = d$.