Math 6397 Bayesian Inverse Problems

Problem Set 3

Due on Friday, May 5, at 10:00 PM

1 Background

We assume that the observation is given by $\mathbf{y}_{\text{obs}} = \mathbf{K}\mathbf{x}_{\text{true}} + \boldsymbol{\eta}$, with $\mathbf{K} \in \mathbb{R}^{n,m}$, $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \tau^{-1}\mathbf{I}_m)$, $\tau > 0$.

The probability density function for $\mathbf{y}_{\text{obs}}$ given $\mathbf{x}$, τ is

$$\pi(\mathbf{y}_{\text{obs}} \mid \mathbf{x}, \tau) = (\tau/(2\pi))^{m/2} \exp\left(-\frac{\tau}{2}\|\mathbf{K}\mathbf{x} - \mathbf{y}_{\text{obs}}\|_2^2\right). \tag{1}$$

We refer to $\pi(\mathbf{y}_{\text{obs}} \mid \mathbf{x}, \tau) =: \pi_{\text{like}}(\mathbf{y}_{\text{obs}} \mid \mathbf{x})$ as the likelihood. The prior is formally denoted by $\pi(\mathbf{x} \mid \beta) =: \pi_{\text{prior}}(\mathbf{x})$. According to Bayes' law the posterior density function is given by

$$\pi(\mathbf{x} \mid \mathbf{y}_{\text{obs}}, \tau, \beta) = \frac{\pi(\mathbf{y}_{\text{obs}} \mid \mathbf{x}, \tau)\pi(\mathbf{x} \mid \beta)}{\pi(\mathbf{y}_{\text{obs}} \mid \tau, \beta)} \propto \pi(\mathbf{y}_{\text{obs}} \mid \mathbf{x}, \tau)\pi(\mathbf{x} \mid \beta)$$

The posterior density function $\pi(\mathbf{x} \mid \mathbf{y}_{\text{obs}}, \tau, \beta)$ defines a probability density for the unknown $\mathbf{x}$ conditioned on $\mathbf{y}_{\text{obs}}$, τ, and β. We can compute point estimates of this random vector, such as the mean, and also quantify its uncertainty by computing measures of variability of $\mathbf{x}$ such as the covariance.

In the example considered below, $\mathbf{x}$ is a discretized function x defined on either $[0, 1]$ or $[0, 1] \times [0, 1]$. That is, $x_i = x((i - 1/2)h)$ for $h = 1/n$, $i = 1, \ldots, n$ or $\mathbf{x} = \text{vec}(\mathbf{X}) \in \mathbb{R}^n$, where $\mathbf{X}$ is an $\tilde{n} \times \tilde{n}$ matrix with entries $[\mathbf{X}]_{ij} = x((i - 1/2)h, (j - 1/2)h)$ for $i, j = 1, \ldots, \tilde{n}$, and $n = \tilde{n}^2$.

Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \mid \beta \sim \mathcal{N}(\bar{\mathbf{x}}, (\beta\mathbf{L})^\dagger)$. The associated (proper) prior distribution $\pi_{\text{prior}}(\mathbf{x}) := \pi(\mathbf{x} \mid \beta)$ is defined as

$$\pi(\mathbf{x} \mid \beta) \propto \exp\left(-\frac{\beta}{2}(\mathbf{x} - \bar{\mathbf{x}})^\mathsf{T}\mathbf{L}(\mathbf{x} - \bar{\mathbf{x}})\right). \tag{2}$$

where $\mathbf{L}$ denotes the precision (inverse-covariance) matrix. Under the assumption that $\mathbf{L} \succ 0$ the $\pi(\mathbf{x} \mid \beta)$ is proper and $\mathbf{x} \mid \beta \sim \mathcal{N}(\bar{\mathbf{x}}, (\beta\mathbf{L})^{-1})$. Using this model, the posterior density function $\pi_{\text{post}}(\mathbf{x} \mid \mathbf{y}_{\text{obs}}) := \pi(\mathbf{x} \mid \mathbf{y}_{\text{obs}}, \tau, \beta)$ is given by

$$\pi_{\text{post}}(\mathbf{x} \mid \mathbf{y}_{\text{obs}}) \propto \pi_{\text{like}}(\mathbf{y}_{\text{obs}} \mid \mathbf{x})\pi_{\text{prior}}(\mathbf{x}) \propto \exp\left(-\frac{\tau}{2}\|\mathbf{K}\mathbf{x} - \mathbf{y}_{\text{obs}}\|_2^2 - \frac{\beta}{2}\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}\right),$$

where we set $\bar{\mathbf{x}} = \mathbf{0}$. This defines the Gaussian random vector

$$\mathbf{x} \mid \mathbf{y}_{\text{obs}}, \tau, \beta \sim \mathcal{N}\left((\tau\mathbf{K}^\mathsf{T}\mathbf{K} + \beta\mathbf{L})^{-1}\tau\mathbf{K}^\mathsf{T}\mathbf{y}_{\text{obs}}, (\tau\mathbf{K}^\mathsf{T}\mathbf{K} + \beta\mathbf{L})^{-1}\right), \tag{3}$$

with $(\tau\mathbf{K}^\mathsf{T}\mathbf{K} + \beta\mathbf{L}) \succ \mathbf{0}$. The uncertainty of the vector $\mathbf{x}$ (conditioned on $\mathbf{y}_{\text{obs}}$, τ, β) is characterized by the covariance matrix $(\tau\mathbf{K}^\mathsf{T}\mathbf{K} + \beta\mathbf{L})^{-1}$ in (3), which depends on both β and τ.

1.1 Maximum a Posteriori Probability Estimate

The map point $\mathbf{x}_{\text{map}}$ is defined as

$$\mathbf{x}_{\text{map}} = \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} - \ln \pi_{\text{post}}(\mathbf{x} \mid \mathbf{y}_{\text{obs}})$$

where

$$-\ln \pi_{\text{post}}(\mathbf{x} \mid \mathbf{y}_{\text{obs}}) = \frac{\tau}{2}\|\mathbf{K}\mathbf{x} - \mathbf{y}_{\text{obs}}\|_2^2 + \frac{\beta}{2}\mathbf{x}^{\mathsf{T}}\mathbf{L}\mathbf{x} = \frac{1}{2}\|\mathbf{K}\mathbf{x} - \mathbf{y}_{\text{obs}}\|_2^2 + \frac{\alpha}{2}\mathbf{x}^{\mathsf{T}}\mathbf{L}\mathbf{x}.$$

This establishes a direct connection to the variational problem formulation of the deterministic inverse problem with regularization parameter $\alpha = \beta/\tau$. To compute the map point $\mathbf{x}_{\text{map}}$ we can simply solve the associated optimality system

$$(\tau\mathbf{K}^{\mathsf{T}}\mathbf{K} + \beta\mathbf{L})\mathbf{x}_{\text{map}} = \tau\mathbf{K}^{\mathsf{T}}\mathbf{y}_{\text{obs}} \iff \mathbf{x}_{\text{map}} = (\tau\mathbf{K}^{\mathsf{T}}\mathbf{K} + \beta\mathbf{L})^{-1}\tau\mathbf{K}^{\mathsf{T}}\mathbf{y}_{\text{obs}} = (\mathbf{K}^{\mathsf{T}}\mathbf{K} + \alpha\mathbf{L})^{-1}\mathbf{K}^{\mathsf{T}}\mathbf{y}_{\text{obs}}.$$

Notice that the map point $\mathbf{x}_{\text{map}}$ corresponds to the mean in (3). Moreover, if we have computed the Cholesky factorization $(\tau\mathbf{K}^{\mathsf{T}}\mathbf{K} + \beta\mathbf{L})^{-1} = \mathbf{C}^{\mathsf{T}}\mathbf{C}$, we can efficiently compute the map point via

$$\mathbf{x}_{\text{map}} = (\mathbf{C}^{\mathsf{T}}\mathbf{C})^{-1}\tau\mathbf{K}^{\mathsf{T}}\mathbf{y}_{\text{obs}} = \mathbf{C}^{-1}\mathbf{C}^{-\mathsf{T}}\tau\mathbf{K}^{\mathsf{T}}\mathbf{y}_{\text{obs}}. \tag{4}$$

1.2 Gaussian Prior Models

Next, we discuss different models for the precision matrix $\mathbf{L}$ that appears in the Gaussian prior model (2), i.e., different prior models for Gaussian Markov Random Fields (GMRF). In addition to that, we introduce conceptual ideas associated with drawing samples from GMRFs.

1.2.1 Smoothness Prior as GMRF

In this section we consider independent, identically distributed priors. To incorporate smoothness assumptions into the GMRF, we will model the matrix $\mathbf{L}$ as a finite difference approximation of the Laplacian operator. For the purpose of modeling the precision matrix we neglect the spatial step size. Assuming zero Dirichlet boundary conditions, we model $\mathbf{L}$ as $\mathbf{L} = \mathbf{Q}$,

$$\mathbf{Q} := \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n,n} \tag{5}$$

in the one-dimensional setting. In the two-dimensional setting, we have

$$\mathbf{L} = \mathbf{I} \otimes \mathbf{Q} + \mathbf{I} \otimes \mathbf{Q} \in \mathbb{R}^{n,n}, \tag{6}$$

where $n = \tilde{n}^2$, i.e., the product of the number of entries along each spatial direction. Notice that this choice of $\mathbf{L}$ guarantees that $\mathbf{L} \succ \mathbf{0}$ and, therefore, $\pi(\mathbf{x} \mid \beta)$ is proper.

1.2.2 Anisotropic GMRF Prior

In this section we focus on independent increment priors that are not identically distributed (i.e., anisotropic). That is, we allow for the increment variance to vary. The increments are defined as the difference between two neighboring values. For example, in a one-dimensional setting these increments are defined as $x_{i+1} - x_i$. If we assume that the increments are independent and identically distributed Gaussian random variables we have $x_{i+1} - x_i \sim \mathcal{N}(0, \beta^{-1})$. The associated precision matrix $\mathbf{L}$ is given by $\mathbf{L} = \mathbf{D}^\mathsf{T}\mathbf{D}$, where $\mathbf{D}$ is a difference matrix. For example,

$$\mathbf{D} := \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n-1,n}, \tag{7}$$

where we assume implicit Neumann boundary conditions. Notice that $\mathbf{D}^\mathsf{T}\mathbf{D}$ coincides with $\mathbf{Q}$ in (5). In the two-dimensional setting, the precision matrix is given by $\mathbf{L} = \mathbf{D}_1^\mathsf{T}\mathbf{D}_1 + \mathbf{D}_2^\mathsf{T}\mathbf{D}_2$, where $\mathbf{D}_1 = \mathbf{I} \otimes \mathbf{D}$ and $\mathbf{D}_2 = \mathbf{D} \otimes \mathbf{I}$, respectively. This expression can be shown to coincide with (6).

So far, this model coincides with the GMRF smoothness prior described earlier. To allow for a spatially adaptive (i.e., anisotropic) increment variance we slightly modify the precision matrix. We introduce a weight matrix $\mathbf{W} := \text{diag}(w_1, \ldots, w_{n-1})$. For the one-dimensional setting the precision matrix is given by $\mathbf{L} = \mathbf{D}^\mathsf{T}\mathbf{W}\mathbf{D}$. For the two-dimensional case, we have analogously $\mathbf{L} = \mathbf{D}_1^\mathsf{T}\mathbf{W}\mathbf{D}_1 + \mathbf{D}_2^\mathsf{T}\mathbf{W}\mathbf{D}_2$.

The matrix $\mathbf{W}$ controls the anisotropy of the prior. An intuitive choice for $\mathbf{W}$ is $[\mathbf{W}(\mathbf{x})^{-1}]_{ij} \propto |x_{i+1} - x_i|$, since this will ensure that the incremental variance is larger where there are large differences. To avoid a division by zero, we add a small perturbation. We have

$$\mathbf{W}(\mathbf{x}) := \text{diag}\left(\mathbf{e}_n \oslash \sqrt{(\mathbf{D}\mathbf{x})^{\circ 2} + \gamma \mathbf{e}_n}\right), \quad 0 < \gamma \ll 1.$$

In the two-dimensional setting $\mathbf{W}$ is defined as

$$\mathbf{W}(\mathbf{x}) := \text{diag}\left(\mathbf{e}_n \oslash \sqrt{(\mathbf{D}_1\mathbf{x})^{\circ 2} + (\mathbf{D}_2\mathbf{x})^{\circ 2} + \gamma \mathbf{e}_n}\right).$$

1.2.3 Sampling from GMRF Priors

The probability density function for a proper GMRF is given by

$$\pi(\mathbf{x} \mid \beta) = \sqrt{\beta^n |\mathbf{L}|/(2\pi)^d} \exp\left(-\frac{\beta}{2}(\mathbf{x} - \bar{\mathbf{x}})^\mathsf{T}\mathbf{L}(\mathbf{x} - \bar{\mathbf{x}})\right), \quad \mathbf{x} \mid \beta \sim \mathcal{N}(\bar{\mathbf{x}}, (\beta\mathbf{L})^{-1}), \quad \mathbf{L} \succ \mathbf{0}.$$

Since $\mathbf{L} \succ \mathbf{0}$, we can compute the Cholesky decomposition $\mathbf{L} = \mathbf{C}\mathbf{C}^\mathsf{T}$ and efficiently draw samples from $\pi(\mathbf{x} \mid \beta)$ via

$$\mathbf{x} \mid \beta = \mathbf{x} + \beta^{-1/2}\mathbf{C}^{-1}\mathbf{v}, \quad \mathbf{v} \propto \mathcal{N}(\mathbf{0}, \mathbf{I}_n).$$

Moreover, we can express the prior probability density function as

$$\pi(\mathbf{x} \mid \beta) \propto \exp\left(-\frac{\beta}{2}\|\mathbf{C}(\mathbf{x} - \bar{\mathbf{x}})\|_2^2\right).$$

Now suppose that $\mathbf{L} \succeq \mathbf{0}$. In this case, we can not compute the Cholesky decomposition. However, we can diagonalize $\mathbf{L}$ to obtain $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathsf{T}$, $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_i \geq 0$, $i = 1, \ldots, n$, with $\mathbf{V}^{-1} = \mathbf{V}^\mathsf{T}$. It can be shown that we can draw $\mathbf{x} \mid \beta \sim \mathcal{N}(\bar{\mathbf{x}}, (\beta\mathbf{L})^\dagger)$ via the whitening process

$$\mathbf{x} \mid \beta = \bar{\mathbf{x}} + \beta^{-1/2}\mathbf{V}(\mathbf{\Lambda}^{1/2})^\dagger\mathbf{v}, \quad \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n).$$

The associated probability density function is given by

$$\pi(\mathbf{x} \mid \beta) = \left((\beta/(2\pi))^{(n-k)/2} \prod_{i=1}^{n-k} \sqrt{\lambda_i}\right) \exp\left(\frac{\beta}{2}(\mathbf{x} - \bar{\mathbf{x}})^\mathsf{T}\mathbf{L}(\mathbf{x} - \bar{\mathbf{x}})\right)$$

1.2.4 Sampling from the IGMRF Prior

In this section, we are going to consider the anisotropic GMRF prior with precision matrix $\mathbf{L} = \mathbf{D}^\mathsf{T}\mathbf{W}\mathbf{D}$ in 1D and $\mathbf{L} = \mathbf{D}_1^\mathsf{T}\mathbf{W}\mathbf{D}_1 + \mathbf{D}_2^\mathsf{T}\mathbf{W}\mathbf{D}_2$ in 2D, respectively. Suppose $\mathbf{L}$ is not SPD (this is the case for the model described in §1.2.2) and, in addition, the diagonalization of $\mathbf{L}$ is not feasible. Consequently, the strategy outlined above cannot be used. Notice that we can decompose $\mathbf{L}$ into $\mathbf{L} = \tilde{\mathbf{C}}^\mathsf{T}\tilde{\mathbf{C}}$, where

$$\tilde{\mathbf{C}} = \mathbf{W}^{1/2}\mathbf{D} \quad \text{or} \quad \tilde{\mathbf{C}} = \begin{bmatrix}\mathbf{W}^{1/2}\mathbf{D}_1 \\ \mathbf{W}^{1/2}\mathbf{D}_2\end{bmatrix}$$

for the one- and two-dimensional case, respectively. It can be shown that the pseudo-inverse of $\mathbf{L} = \tilde{\mathbf{C}}^\mathsf{T}\tilde{\mathbf{C}}$ is given by

$$\mathbf{L}^\dagger = \tilde{\mathbf{C}}^\dagger(\tilde{\mathbf{C}}^\dagger)^\mathsf{T} \quad \text{and} \quad \tilde{\mathbf{C}}^\dagger = \lim_{\epsilon \searrow 0}(\tilde{\mathbf{C}}^\mathsf{T}\tilde{\mathbf{C}} + \epsilon\mathbf{I})^{-1}\tilde{\mathbf{C}}^\mathsf{T}.$$

If we select $0 < \epsilon \ll 1$, we can compute accurate approximate samples $\mathbf{x} \mid \beta \sim \mathcal{N}(\bar{\mathbf{x}}, (\beta\mathbf{L})^\dagger)$ via

$$\mathbf{x} \mid \beta = \bar{\mathbf{x}} + \beta^{-1/2}(\tilde{\mathbf{C}}^\mathsf{T}\tilde{\mathbf{C}} + \varepsilon\mathbf{I})^{-1}\tilde{\mathbf{C}}^\mathsf{T}\mathbf{v}, \quad \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m),$$

where m is the number of rows in $\tilde{\mathbf{C}}$. Notice that we can use a Cholesky factorization to efficiently invert $\tilde{\mathbf{C}}^\mathsf{T}\tilde{\mathbf{C}} + \varepsilon\mathbf{I} \succ \mathbf{0}$ since it is a sparse and banded matrix.

1.3 Generalized Cross Validation

Generalized cross-validation (GCV) is another common method to compute an optimal regularization parameter α. The optimal value is found by solving the unconstrained optimization problem

$$\underset{\nu \in \mathbb{R}}{\text{minimize}} \frac{\|\mathbf{K}\mathbf{x}_\nu - \mathbf{y}_{\text{obs}}\|_2^2}{\text{trace}(\mathbf{I} - \mathbf{K}\mathbf{K}_\nu)}, \tag{8}$$

where $\mathbf{x}_\nu > 0$ is the Tikhonov solution of the inverse problem, $\mathbf{K}_\nu$ is the regularization matrix such that

$$\mathbf{x}_\nu = \mathbf{R}_\nu\mathbf{y}_{\text{obs}}, \quad \mathbf{K}_\nu := \mathbf{R}_\nu.$$

More precisely,

$$\mathbf{x}_\nu = (\mathbf{K}^\mathsf{T}\mathbf{K} + \nu\mathbf{L})^{-1}\mathbf{K}^\mathsf{T}\mathbf{y}_{\text{obs}} = \mathbf{R}_\nu\mathbf{y}_{\text{obs}}.$$

The minimizer $\nu^\star$ of (8) is the optimal regularization parameter α.

1.4 Test Problem

The test problem will be the deconvolution problem from past homework assignments. All implementations needed for setting up the test problem are provided in the prbsets/sample1D and prbsets/sample2D subfolders.

2 Assignments

1. Write a function to compute the matrix $\mathbf{Q}$ in (5) used to construct the precision matrix $\mathbf{L}$ of the Gaussian smoothness prior $\pi(\mathbf{x} \mid \beta)$ in (2). Implement $\mathbf{L}$ for the one- and two-dimensional case, respectively. **Hint:** *A template for the implementation of the finite difference operator* $\mathbf{Q}$ *can be found in core/getLapMat.m. Matlab provides an implementation for the Kronecker product between matrices; the function is called* `kron`*. To efficiently construct sparse matrices of the form* $\mathbf{Q}$ *Matlab provides a function called* `spdiags`*. An example for the construction of the gradient operator* $\mathbf{D}$ *in* (7) *can be found in core/getGradMat.m. You can use the script prbsets/draw/scPrecMatPD.m to verify that your implementation of* $\mathbf{L}$ *is indeed an SPD matrix.*

2. Implement the GCV algorithm described in §1.3 to estimate an optimal regularization parameter α. That is, implement a function that estimates α by solving the one-dimensional optimization problem (8). **Hint:** *We will use Matlab's* `fminbnd` *to solve* (8)*. A template for the implementation of GCV is core/evalGCV.m. You can use prbsets/sample1D/scDeconvGCVLAP1D.m to test your implementation.*

3. Write an algorithm to compute the MAP point $\mathbf{x}_{\text{map}}$ for a one-dimensional deconvolution problem with independent increment (anisotropic) IGMRF prior. Notice that the precision matrix depends on $\mathbf{x}$ through $\mathbf{W}$. Consequently, as $\mathbf{x}$ changes, we need to update $\mathbf{W}$. This is accomplished by based on an iterative algorithm:

   $k \leftarrow 1$; $\mathbf{W}_k = \mathbf{I}_{n-1}$
   while $k < n_{\text{iter}}$ **do**
   $\mathbf{L}_k \leftarrow \mathbf{D}^{\mathsf{T}}\mathbf{W}_k\mathbf{D}$
   $\alpha_k \leftarrow \text{GCV}(\mathbf{K}, \mathbf{L}_k, \mathbf{y}_{\text{obs}})$
   $\mathbf{x}_k \leftarrow (\mathbf{K}^{\mathsf{T}}\mathbf{K} + \alpha_k\mathbf{L}_k)^{-1}\mathbf{K}^{\mathsf{T}}\mathbf{y}_{\text{obs}}$
   $\mathbf{W}_{k+1} \leftarrow \text{diag}\left(\mathbf{e}_n \oslash \sqrt{(\mathbf{D}\mathbf{x}_k)^{\circ 2} + \gamma\mathbf{e}_{n-1}}\right)$
   $k \leftarrow k + 1$
   end while

   As can be seen above, we use the GCV algorithm to estimate an optimal value for α at each iteration. **Hint:** *A template for the implementation of this algorithm can be found in prbsets/sample1D/scDeconvIGMRFEP1D.m.*

4. Draws $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, (\beta\mathbf{L})^{-1})$ from a proper GMRF for $\mathbf{L} \succ 0$ (see §1.2.3).

   a) Implement an algorithm that will draw a random vector $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, (\beta\mathbf{L})^{-1})$ where $\mathbf{L} \succ 0$ given by $\mathbf{Q}$ and $\beta = 1$. Display 6 realizations of $\mathbf{x}$. **Hint:** *A script for the implementation of this algorithm can be found in prbsets/draw/scDrawGMRFDBC1D.m. To draw random vectors* $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ *from*

a standard normal distribution you can use Matlab's `randn` *function. To compute the Cholesky decomposition* $\mathbf{C}^\mathsf{T}\mathbf{C}$ *of the precision matrix* $\mathbf{L}$ *you can use Matlab's* `chol` *function.*

b) Extend your implementation from part a) to the two-dimensional setting, where $\mathbf{L} = \mathbf{I}\otimes\mathbf{Q}+\mathbf{I}\otimes\mathbf{Q} \succ \mathbf{0}$. Display 6 realizations of $\mathbf{x}$. **Hint:** *A script for the implementation of this algorithm can be found in prbsets/draw/scDrawGMRFDBC2D.m.*

5. Draws $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, (\beta\mathbf{L})^\dagger)$ from a IGMRF for which $\mathbf{L}$ is not SPD (see §1.2.4).

a) Let $\mathbf{L}^\dagger$ denote the pseudoinverse of $\mathbf{L}$. Proof that $\mathbf{L}^\dagger = \lim_{\epsilon\searrow 0} \left(\mathbf{L}^\mathsf{T}\mathbf{L} + \epsilon\mathbf{I}\right)^{-1}\mathbf{L}^\mathsf{T}$.

b) Implement an algorithm that will draw a random vector $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, (\beta\mathbf{L})^\dagger)$, where $\beta = 1$, $\mathbf{L} = \mathbf{D}\mathbf{W}\mathbf{D}^\mathsf{T}$, $\mathbf{D}$ is as in (7), and the matrix $\mathbf{W}$ is given by $\mathbf{W} = \mathrm{diag}(w_1, \ldots, w_n)$ with $w_i = 1$ for all $i = 1, \ldots, n$ with the exception of $w_{n/2} = 25/1000$. Use the algorithm described in §1.2.4. Use a Cholesky decomposition to efficiently apply the inverse of $(\tilde{\mathbf{C}}^\mathsf{T}\tilde{\mathbf{C}} + \varepsilon\mathbf{I})^{-1}$. Set ϵ to the square root of machine precision. Display 6 realizations of $\mathbf{x}$. **Hint:** *A script for the implementation of this algorithm can be found in prbsets/draw/scDrawIGMRFNBC1D.m. The gradient operator* $\mathbf{D}$ *in (7) is implemented in core/getGradMat.m.*

c) Extend your implementation to the two-dimensional setting. The weight matrix $\mathbf{W}$ will have weights $w_{ij} = 1.0$ unless the points are located on the boundary of a circle along which we set $w_{ij} = 25/1000$. Display 6 realizations of $\mathbf{x}$. **Hint:** *A script for the implementation of this algorithm can be found in prbsets/draw/scDrawIGMRFNBC2D.m. The gradient operator for the two-dimensional setting is also implemented in core/getGradMat.m.*

6. We consider the one-dimensional source reconstruction problem from past homework assignments. Our goal is to draw samples from the distribution defined by (3). Since we consider a one-dimensional problem, we can explicitly form the precision matrix $\tau\mathbf{K}^\mathsf{T}\mathbf{K} + \beta\mathbf{L}$. Since this matrix is SPD we can compute its Cholesky factorization. For the precision matrix $\mathbf{L}$ use $\mathbf{Q}$ in (5). Estimate τ based on the variance of $\mathbf{K}^\mathsf{T}\mathbf{K}\mathbf{x}_\alpha - \mathbf{y}_{\text{obs}}$, where $\mathbf{x}_\alpha$ is the Tikhonov solution for an adequate regularization parameter α. Select α based on GCV. Given τ and α, you can estimate β as $\beta = \alpha\tau$. The strategy to sample $\mathbf{x}$ from (3) is identical to your implementation under question 4. That is, it can be shown that samples from the distribution defined by (3) can be computed via

$$\mathbf{x} \mid \mathbf{y}_{\text{obs}}, \tau, \beta = \mathbf{C}^{-1}(\mathbf{C}^{-\mathsf{T}}\tau\mathbf{K}^\mathsf{T}\mathbf{y}_{\text{obs}} + \mathbf{v}), \quad \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n).$$

Notice that this expression involves the MAP point $\mathbf{x}_{\text{map}}$; you can use (4) to compute $\mathbf{x}_{\text{map}}$, where τ is selected as stated above. Draw 1,000 samples and visualize the mean and the 95% credibility interval. **Hint:** *A template implementation for this question is sample1D/scDeconvGMRFLAP1D.m. To compute the mean of the samples, you can use Matlab's* `mean` *function. A function to compute the 95% credibility intervals from the 1,000 samples you have drawn is implemented in core/getEmpQuant.m.*