Chapter 1

A cruel lasso thesis

We want to relate a response vector $Y \in \mathbb{R}^n$ and an input matrix $X \in \mathbb{R}^{n\times p}$ with a linear parameter $\beta \in \mathbb{R}^p$ and some noise $\epsilon \in \mathbb{R}^n$

$$Y = X\beta_0 + \epsilon$$

1.1 Ordinary Least square method

We look for solutions of

$$\min_\beta \|Y - X\beta\|_2^2, \|\cdot\|_2^2 = \sum_{k=1}^n \cdot_k^2$$

We consider $\widehat{\beta}_{OLS} = \operatorname{argmin}_\beta \|Y - X\beta\|_2^2$ an estimator of β, and we can show that

$$\widehat{\beta}_{OLS} = (X^T X)^\dagger X^T Y$$

Then, when the ϵ_i are uncorrelated random variables with mean 0 and variance σ_0^2

$$\mathbb{E}\|X(\widehat{\beta}_{OLS} - \beta_0)\|_2^2 = \sigma_0^2 \times \sum_i \mathbb{1}_{\lambda_i \neq 0}$$

1.1.1 For Hawkes process

The intensity of a Hawkes process is $\lambda(t) = \mu + \sum_{k s.t. T_k < t} h(t - T_k) = \mu + \int_{-\infty}^{t^-} h(t-u)\ \mathrm{d}N_u$
We want to find the constant μ and the h functions, but the space is too large, so we parametrize by

$$f_a = \sum_{k=00}^{K-1} a_k \phi_k \tag{1.1}$$
$$\phi_{00} = (1, 0, \dots),\ \phi_k = (0, \dots, 0, \mathbb{1}_{(k\delta,(k+1)\delta]}) \tag{1.2}$$
$$a = (a_{00} = \mu, a_0, \dots, a_{K-1})^T \tag{1.3}$$

Now we look at the linear predictable function ψ transforming $f = (\mu, h)$ into (a candidate intensity) $\psi_t(f_a) = \mu + \int_{-\infty}^{t^-} h(t-u)\ \mathrm{d}N_u$
Applied to the restriction f_a

$$\psi_t(f_a) = \sum_{k=00}^{K-1} a_k \psi_t(\phi_k) \tag{1.4}$$
$$\psi_t(\phi_{00}) = 1,\ \psi_t(\phi_k) = \sum_{T<t} \mathbb{1}_{(k\delta,(k+1)\delta]}(t - T) \tag{1.5}$$

So $\psi_t(\phi_k)$ is the number of spikes in the interval $(k\delta, (k+1)\delta]$ weighted by the time elapsed since the spike (integral of a piecewise constant function)

We pose $G_{kl} = \int_0^{T_{\max}} \psi_t(\phi_k)\psi_t(\phi_l)\ \mathrm{d}t$ and $b_k = \int_0^{T_{\max}} \psi_t(\phi_k)\ \mathrm{d}N_t$ and look for

$$\text{minimize}\left\{a^T G a - 2b^T a\right\}$$

The least square estimator is then $\widehat{a} \in \operatorname{argmin}_a \left\{a^T G a - 2b^T a\right\}$, and assuming $G \geq c \times I_d$ then $\widehat{a} = G^{-1} b$
How good is the estimator?

$$\mathbb{E}\|\psi_t(f_a) - \lambda\|^2_{\text{proc}} \leq \frac{1}{c} \sum_k \mathbb{E}\left(\int_0^{T_{\max}} \psi_t^2(\phi_k)\lambda_t \ \mathrm{d}t\right)$$

1.2 Lasso method

The lasso estimator is $\widehat{\beta} \in \operatorname{argmin}_\beta \left\{\|Y - X\beta\|_2^2 + 2\lambda\|\beta\|_1\right\}$ which is equivalent of solving a 2-polynom in β.

1.3 Lasso shooting

We still want to minimize $\widehat{a} \in \operatorname{argmin}_a \left\{a^T G a - 2b^T a + 2\|d^T a\|_1\right\}$. The shooting algorithms simplifies the resolution by considering n independent one-dimension optimization problem (one for each of a's coordinate), and the solution is given by

$$a_i^* = \begin{cases} \frac{b_i - \sum_{j\neq i} a_j G_{ij} - d_i}{G_{ii}}, & \text{if} \quad b_i - \sum_{j\neq i} a_j G_{ij} > d_i \\ \frac{b_i - \sum_{j\neq i} a_j G_{ij} - d_i}{G_{ii}}, & \text{if} \quad b_i - \sum_{j\neq i} a_j G_{ij} < -d_i \\ 0, & \text{otherwise} \end{cases}$$

Hence the following algorithm

Algorithm 1 Lasso shooting algorithm

```
1: Initialize m = 1 and a^0
2: repeat
3:     for i ← 00 to K-1 do
4:         Update a_i following rule given above
5: until |F(a_m) − F(a_{m−1})| < ε
```

The computation of a depends on the values of G, b and d, but thanksfully one can compute them by simply counting the number of pair of events the difference of which are in the time bins $(k\delta, (k+1)\delta]$. They are stored in a matrix A:

Algorithm 2 Counting the pairs

```
Require: T; array of spike trains
  δ; delay
  k; number of partitions
  count ← 0, N ← length(T)
  j_start ← first index s.t. T_j > 0
  for j ← j_start to N do
      i ← j − 1
      T_low ← T[j] − (k + 1)δ, T_up ← T[j] − kδ
      while i > 0 & i < N do
          if T_low > spike(i) then
              break
          else if spike(i) < T_up then
              count+ = 1
              A_ij = k
          i ← i − 1
```

Knowing the matrix of A we can now compute the remaining bits. First the matrix G:
Then the matrix d of errors:
Now, having b, G and d we can implement the Lasso shooting algorithm:

Algorithm 3 Compute G_{kl} given k and l

```
for l ← 0 to K-1 do
    for k ← l+1 to K-1 do
        rescase1 = rescase2 = 0
        Case 1
        for all (i,j)|A_ij = k − l do
            length= min(T_i + (k+1)δ, T_max) − max(T_j + lδ, 0)
            if length> 0 then
                rescase1+=length
        Case 2
        for all (i,j)|A_ij = k − l − 1 do
            length= min(T_j + (l+1)δ, T_max) − max(T_i + kδ, 0)
            if length> 0 then
                rescase2+=length
        G_(k+1)(l+1) =rescase1+rescase2
for k ← 0 to K-1 do
    rescase1=0
    for all (i,j)|A_ij = 0 do
        length= min(T_i + (k+1)δ, T_max) − max(T_i + kδ, 0)
        if length> 0 then
            rescase1+=length
    for i ← 1 to N do
        length= min(T_i + (k+1)δ, T_max) − max(T_i + kδ, 0)
        if length> 0 then
            rescase3+=length
    G_(k+2)(k+2) 2× =rescase1+rescase3
G_11 = T_max
for k ← 0 to K − 1 do
    rescase= 0
    for i ← 1 to length(T) do
        length= min(T_i + (k+1)δ, T_max) − max(T_i + kδ, 0)
        if length> 0 then
            rescase+=length
    G_(k+2)(1) =rescase
```

Algorithm 4 Computing d

d=zeros(K+1,1)
d(1)=$\sqrt{2*\gamma*\log(T_{\max})*N}+\frac{\gamma*\log(T_{\max})}{3}$
for k $\leftarrow$ 0 **to** K-1 **do**
  d(k+2)=COMPUTED

function COMPUTED(A, del, spike, $T_{\max}$, k, γ)
  for $i \leftarrow 1$ **to** lengthspike **do**
    if spike$(i)+(k+1)*\text{del} \leq T_{\max}$&spike$(i)+(k+1)*\text{del} \geq 0$ **then**
      $z(i)$ =COUNTSPIKE(spike, del, k, spike$(i)+(k+1)*\text{del}$)
    if spike$(i)+k*\text{del} < T_{\max}$&spike$(i)+k*\text{del} \geq 0$ **then**
      $z(i)=\max(z(i),1)$
  $I_{\text{col}}=\{i|A_{ij}=k\}$
  count = 0, conse = 1
  if isempty(I_{col}) **then**
    res = 0
  else
    compare= $x(1)$
    $x_{n+1}=-1$
    while $i \leq n$ **do**
      if x_{i+1} =compare **then**
        conse += 1
      else
        count+ = conse $*$ (conse $-$ 1)
        conse = 1
        compare = x_{i+1}
      i += 1
    res = n + count
  $d_k=\frac{\gamma*\log(T_{\max})}{3}*\max(z)+\sqrt{2*\gamma*\log(T_{\max})*\text{res}}$

Algorithm 5 countspike

resuk $\leftarrow$ 0; $N \leftarrow$ length(spike); $\varepsilon \leftarrow$ 1e-12
for $i \leftarrow 1$ **to** N **do** ▷ N should be ≥ 1!
  if (spike$[i] \geq (t-(k+1)*\text{del})$) & (spike$[i] < (t-k*\text{del}-\varepsilon)$) **then** ▷ ε for numerical errors
    resuk += 1
return resuk

Algorithm 6 The Lasso algorithm

$a=(0,\ldots,0)$
repeat
  $a_{\text{old}}=a$
  for $i \leftarrow 1$ **to** $K+1$ **do**
    $J=\{j \neq i \& j \leq K+1\}$
    $z=b_i-G_{ii}*a_i$
    if $|z| \leq d_i$ **then**
      $a^*=0$
    else if $z > d_i$ **then**
      $a^*=(z-d_i)/G_{ii}$
    else
      $a^*=(z+d_i)/G_{ii}$
  $\text{las}=a^T*G*a-2*b^T*a+2*d^T*|a|$
  $\text{las}_{\text{old}}=a_{\text{old}}^T*G*a_{\text{old}}-2*b^T*a_{\text{old}}+2*d^T*|a_{\text{old}}|$
until $|\text{las}-\text{las}_{\text{old}}| < \epsilon$

Chapter 2

Active sets and optimisations

2.1 Model

$y \in \mathbb{R}^{dn}$ a signal recorded by d sensors
Generated by k (fixed) neurons
n: number of recordings (time length) (=number of samples) H is the convolution between the shape (=action potential) of the neurons

$$x = x$$

2.2 Lasso

We assume some sparsity on a.
LASSO: $\min_{a \in \mathbb{R}^{kn}} \underbrace{\frac{1}{2}\|y - Ha\|_2^2 + \lambda\|a\|_1}_{:=F(a)}$, with $\lambda > 0$
F is strictly convex but not differentiable$\Rightarrow$need a generalisation of differentiable to sub-differentiable

2.2.1 Subdifferentiability

Def: for $g : \mathbb{R}^p \to \mathbb{R}$ convex and a vector ω on $\mathbb{R}^p$

$$\partial g(\omega) = \{z \in \mathbb{R}^p | \underbrace{g(\omega) + z^T(\omega' - \omega)}_{\text{tangente: } \forall \omega' \in \mathbb{R}^p} \leq g(\omega')\}$$

Ex: $\partial g(\omega) = \{\nabla g(\omega)\}$ if g is convex and differentiable For the absolute value $\partial H(0) = [-1, 1]$ Prop: $\omega^* \in \mathbb{R}^p$ is a global minimum of g iif $0 \in \partial g(\omega^*)$

2.3 Improving the Lasso with an active set dynamic constraint

F is strictly convex but not differentiable
$\partial F(a) = H^T(Ha - y) + \lambda \partial \| \cdot \|_1(a)$, with $\partial \| \cdot \|(a) = \{x \in \mathbb{R}^{kn} \begin{cases} x_j = \text{sign}(a_j), & \text{if } a_j \neq 0 \\ x_j = [-1, 1], & \text{else} \end{cases} \}$
a^* is a minimum of F iif $\forall j \in \{1, \dots, kn\}$, $\begin{cases} H_j^T(y - Ha) = \lambda \text{sign}(a_j^*), & \text{if } a_j^* \neq 0 \\ |H_j^T(y - Ha^*)| \leq \lambda & \text{else} \end{cases}$
In particular: $|H_j^T(Ha^* - y)| < \lambda \Rightarrow a_j^* = 0$ (ST)
Instead of solving LASSO in $\mathbb{R}^{kn}$, we will start from $a = 0$ and activate the meaningfull variables, and stop when the (ST) condition is met $\forall j$

Algorithm 7 Active set 1

$a \leftarrow 0$, $g \leftarrow |H^T(y - Ha)|$, $j \leftarrow \text{argmax}_l g_l$
Active set: $J = \{j\}$
while An arbitrary condition is not met (eg: $m < kn - 1$) **do**
  $a \leftarrow$Lasso solution on J
  $g \leftarrow |H^T(y - Ha)|$
  $j \leftarrow \text{argmax}_{l \notin J} g_l$
  if $g_j > \lambda + \epsilon$ **then**
    $J+ = \{j\}$
  else
    break

2.4 Speeding up some matrix multiplications

$R := y - \underbrace{Ha}_{\text{too expensive}}$ ($H \in \mathbb{R}^{dn \times kn}$, typically $d = 5, k = 5, n = 10^5$)

$$\begin{aligned}
(Ha)_i &= \sum_{j=1}^{kn} H_{ij} a_j \\
&= \sum_{j \in J} H_{ij} a_j \\
&= \sum_{j=1}^{\text{card}(J)} \widetilde{H}_{ij} \widetilde{a}_j = \widetilde{H}\widetilde{a} \rightarrow \begin{cases} \widetilde{H} = (H_j)_{j \in J} \\ \widetilde{a} = a_J \end{cases}
\end{aligned}$$

We end up only having to compute the elements from the active set reducing a lot the overall complexity.

2.5 Some more matrix multiplication speeding up

We can compute R faster, but can we speed up $H^T R$?
$H^T R$: "correlation between the shapes of the spikes and R" Current a set J
New variable j in the active set
How to update the information?
$|\text{mod}(J, n) - \text{mod}(j, n)| \geq t$, solve LASSO on $1 - D$: LASSO$(y, \underbrace{H_j}_{\in \mathbb{R}^{dn}}, 0)$ $J = \{1, 2, 10, 50\} = \{1, 2\} \cup \{10\} \cup \{50\}$

Chapter 3

Some improvements about the algorithms

Appendices

If A is a matrix

- A^T is the transpose of the matrix $(\forall (i,j), \left(A^T\right)_{ij} = A_{ji})$
- A^* is the conjugate transpose $(\forall (i,j), (A^*)_{ij} = \overline{A_{ji}})$
- $A^\dagger$ is the pseudo-inverse of the matrix A, which is a generalisation of the inverse of a matrix. The pseudo-inverse verify 4 properties:

$$AA^\dagger A = A \tag{1}$$
$$A^\dagger AA^\dagger = A^\dagger \tag{2}$$
$$\left(AA^\dagger\right)^* = AA^\dagger \tag{3}$$
$$\left(A^\dagger A\right)^* = A^\dagger A \tag{4}$$