UiO : Matematisk institutt

Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2021
SGD

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

Last time

- EM in Exponential family
- Variance estimate in EM
- Bootstrap
- EM for hidden Markov model

- Stochastic gradient decent
  - What it is
  - Minibatch is one type of randomness
  - Proof of convergence Part 1

Question

- When finding the derivative of the Q_lagran with respect to beta (for eksample), it seems that t_i(theta^s) is treated as a constant, but doesn't this contain beta? ISn't beta part of the normalising sum? Or is beta^(s) only used in the expectation?

$$l(\boldsymbol{\theta}) = n_{z,0}\log(\alpha) + \sum_{i=0}^{16}[n_{t,i}(\log(\beta) + i\log(\mu) - \mu) + n_{p,i}(\log(\gamma) + i\log(\lambda) - \lambda)]$$

Then (with $\boldsymbol{n} = (n_0, ..., n_{16})$ and using s to denote iteration number in order not to confuse with t in model)

$$Q_{lagr}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + \phi(1 - \alpha - \beta - \gamma)$$

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) =& E[N_{z,0}|\boldsymbol{n}, \boldsymbol{\theta}^{(s)}]\log(\alpha) + \\ & \sum_{i=0}^{16}[E[N_{t,i}|\boldsymbol{n}, \boldsymbol{\theta}^{(s)}](\log(\beta) + i\log(\mu) - \mu) + \\ & \sum_{i=0}^{16} E[N_{p,i}|\boldsymbol{n}, \boldsymbol{\theta}^{(s)}](\log(\gamma) + i\log(\lambda) - \lambda)] \end{aligned}$$

When geting the max of Q, these expectations are given
The current estimate of parameter is used.

$$E[N_{t,i}|\boldsymbol{n},\boldsymbol{\theta}^{(s)}] = n_i\frac{\beta^{(s)}(\mu^{(s)})^i\exp(-\mu^{(s)})}{\pi_i(\boldsymbol{\theta}^{(s)})} = n_i t_i(\boldsymbol{\theta}^{(s)})$$

We similarly get

$$E[N_{z,0}|\boldsymbol{n},\boldsymbol{\theta}^{(s)}] = n_0\frac{\alpha^{(s)}}{\pi_0(\boldsymbol{\theta}^{(s)})} = n_0 z_0(\boldsymbol{\theta}^{(s)})$$
$$E[N_{p,i}|\boldsymbol{n},\boldsymbol{\theta}^{(s)}] = n_i\frac{\gamma^{(s)}(\lambda^{(s)})^i\exp(-\lambda^{(s)})}{\pi_i(\boldsymbol{\theta}^{(s)})} = n_i p_i(\boldsymbol{\theta}^{(s)})$$

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = E[N_{z,0}|\boldsymbol{n},\boldsymbol{\theta}^{(s)}]\log(\alpha) + \sum_{i=0}^{16}[E[N_{t,i}|\boldsymbol{n},\boldsymbol{\theta}^{(s)}](\log(\beta)+i\log(\mu)-\mu) + \sum_{i=0}^{16}E[N_{p,i}|\boldsymbol{n},\boldsymbol{\theta}^{(s)}](\log(\gamma)+i\log(\lambda)-\lambda)]$$

Further, introducing the Lagrange term,

$$Q_{lagr}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + \phi(1-\alpha-\beta-\gamma)$$

we get

$$\frac{\partial}{\partial\alpha}Q_{lagr}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = n_0 z_0(\boldsymbol{\theta}^{(s)})\frac{1}{\alpha} - \phi$$

so

$$\alpha^{(s+1)} = \frac{1}{\phi}n_0 z_0(\boldsymbol{\theta}^{(s)})$$

$$\frac{\partial}{\partial\beta}Q_{lagr}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = \sum_{i=0}^{16} n_i t_i(\boldsymbol{\theta}^{(s)})\frac{1}{\beta} - \phi$$

so

$$\beta^{(s+1)} = \frac{1}{\phi}\sum_{i=0}^{16} n_i t_i(\boldsymbol{\theta}^{(s)})$$

$$\frac{\partial}{\partial\gamma}Q_{lagr}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = \sum_{i=0}^{16} p_i(\boldsymbol{\theta}^{(s)})\frac{1}{\gamma} - \phi$$

so

$$\gamma^{(s+1)} = \frac{1}{\phi}\sum_{i=0}^{16} n_i p_i(\boldsymbol{\theta}^{(s)})$$

By noting that

$$n_0 z_0(\boldsymbol{\theta}^{(s)}) + \sum_{i=0}^{16} n_i t_i(\boldsymbol{\theta}^{(s)}) + \sum_{i=0}^{16} n_i p_i(\boldsymbol{\theta}^{(s)}) = N$$

we get: $\phi = N$

Info

- Many good videos on course topics online
  - Some gives a an overview
  - Some gives details
  - Be critical, is what you get what you need?
- Explanation and example of the EM algorithm for the for the mixture gaussian case (thanks to Susie Jentoft)
  - https://www.youtube.com/watch?v=REypj2sy_5U
  - https://www.youtube.com/watch?v=iQoXFmbXRJA
- Next week (After exercise 15.15-15.35) your co-student Susie Jentoft will give a presentation of R-Markdown. Usefull for documentation in R [will be recorded].

SGD convergence; Assumptions

- Requirements on the sequence $\{\alpha_t\}$:

$$\alpha_t > 0 \tag{A-1}$$

$$\sum_{t=2}^{\infty} \frac{\alpha_t}{\alpha_1 + \cdots + \alpha_{t-1}} = \infty \tag{A-2}$$

$$\sum_{t=1}^{\infty} \alpha_t^2 < \infty \tag{A-3}$$

Note that (A-2) implies $\sum_{t=1}^{\infty} \alpha_t = \infty$

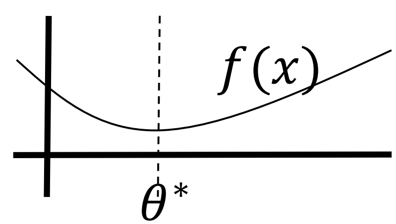
- Requirements on the function $g(x)$ combined with its estimate:

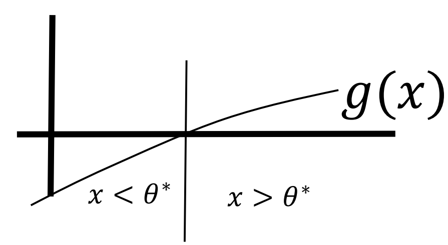
$g(x)$ has same sign as $(x - \theta^*)$

$$\exists \delta \geq 0 \text{ such that } g(x) \leq -\delta \text{ for } x < \theta^* \text{ and } g(x) \geq \delta \text{ for } x > \theta^*. \tag{A-4}$$

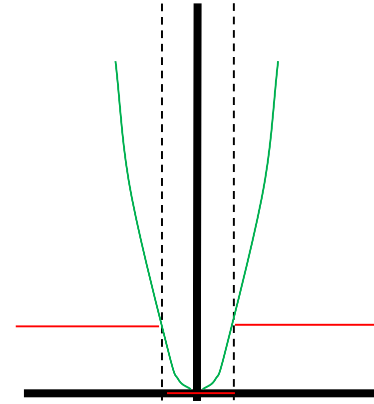
$$E[Z(\theta;\phi)] = g(\theta) \text{ and } \Pr(|Z(\theta;\phi)| < C) = 1 \tag{A-5}$$

The constraint $|Z(\theta;\phi)| < C$ is included to simplify the proof. More general results are available.





The SGD procedure is consistent



- Three steps (with some sub-steps on the way)
  1. Prove that L2 convergence gives consistency
  2. Prove that the sequence converge

$$b_{t+1} = b_1 + \sum_{s=1}^{t} \alpha_s^2 e_s - 2 \sum_{s=1}^{t} \alpha_s d_s. \qquad \boxed{e_t = E[Z_t^2] \quad d_t = E[(\theta^t - \theta^*) g(\theta^t)],}$$

  3. Prove that we converge to the true parameter
     Needed to find k_t such that:

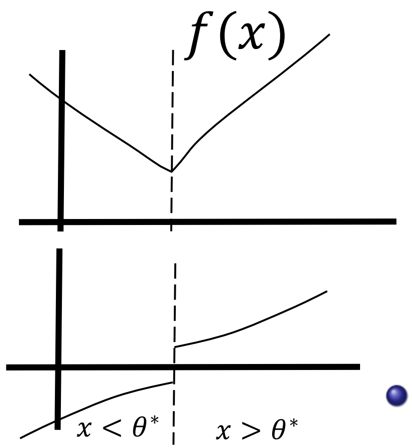
$$k_t b_t \leq d_t, \quad \sum_{t=1}^{\infty} \alpha_t k_t = \infty$$

Proposed value for k_t is: $k_t = \inf_{x \in [\theta^* - A_n, \theta^* + A_n]} \left[\frac{g(x)}{x - \theta^*} \right] \geq 0$ from (A-4)

Where A_n is an upper limit on distance between θ^t and θ^*

1. k_t is constructed such that $k_t b_t < d_t$, Need to show the last sum is infinite
2. Separate into two cases

Case 1



(A4) with a strictly positive δ

$\exists \delta > 0$ such that $g(x) \leq -\delta$ for $x < \theta^*$ and $g(x) \geq \delta$ for $x > \theta^*$.

- By (A-2), $\sum_{t=1}^{\infty} \alpha_t = \infty$ which implies that for t larger than some T

$$2C(\alpha_1 + \cdots + \alpha_{t-1}) = A_t + C(\alpha_1 + \cdots + \alpha_{t-1}) - |\theta^1 - \theta^*| \geq A_t.$$

This results in that

$$\sum_{t=1}^{\infty} \alpha_t k_t \geq \sum_{t=\max\{N,T\}}^{\infty} \alpha_t k_t \geq \sum_{t=\max\{N,T\}}^{\infty} \frac{\alpha_t \delta}{A_t}$$

$$\boxed{k_t = \inf_{x \in [\theta^* - A_n, \theta^* + A_n]} \left[\frac{g(x)}{x - \theta^*} \right]}$$

$$\geq \sum_{t=\max\{N,T\}}^{\infty} \frac{\alpha_t \delta}{2C(\alpha_1 + \cdots + \alpha_{t-1})}$$

$$= \frac{\delta}{2C} \sum_{t=\max\{N,T\}} \frac{\alpha_t}{\alpha_1 + \cdots + \alpha_t} = \infty$$

$$\left\{ \begin{array}{l} \delta < |g(x)| \\ \dfrac{1}{|x - \theta^*|} < \dfrac{1}{A_t} \end{array} \right.$$

Lemma 3.

Assume (A-1), (A-2), (A-3), (A-4) *and* (A-5). *Assume for some constant* $\delta > 0$ *that*

$$\inf_{z \in [\theta^* - A_t, \theta^* + A_t]} \left[\frac{g(z)}{z - \theta^*} \right] \geq \frac{\delta}{A_t} \text{ for } t > N \tag{7}$$

where

$$A_t = |\theta^1 - \theta^*| + C(\alpha_1 + \cdot\cdot|\cdot + \alpha_{t-1}). \tag{8}$$

Then $\lim_{t\to\infty} b_t = 0$.

Theorem 2.

Assume (A-1), (A-2), (A-3), (A-4) *and* (A-5). *Assume further* $\delta > 0$ *in* (A-4). *Then* $\lim_{t\to\infty} b_t = 0$.

Proof:
We have for any $z \in [\theta - A_n, \theta + A_n]$

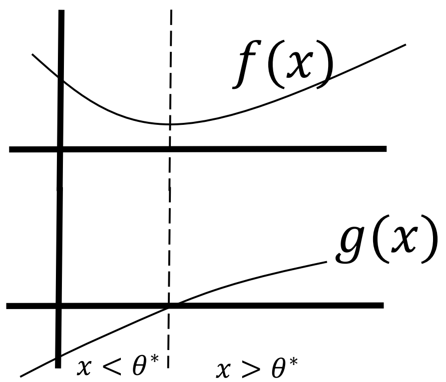
$$\frac{g(z)}{z - \theta} \geq \frac{\delta}{|z - \theta|} \geq \frac{\delta}{A_t}$$

implying that (7) is fulfilled which by Lemma 3 imply the result.

Here
«δ» in (A-4)
can be used directly as
«δ» in Lemma 3

(A4) with a $\delta = 0$

$\exists \delta \geq 0$ such that $g(x) \leq -\delta$ for $x < \theta^*$ and $g(x) \geq \delta$ for $x > \theta^*$.



Theorem 3.

Assume (A-1), (A-2), (A-3) *and* (A-5). *Assume further*

$$g(z) \text{ is nondecreasing;} \tag{9}$$

$$g(\theta^*) = 0; \tag{10}$$

$$g'(\theta^*) > 0. \tag{11}$$

Then $\lim_{t\to\infty} b_t = 0$.

Lemma 3.

Assume (A-1), (A-2), (A-3), (A-4) *and* (A-5). *Assume for some constant* $\delta > 0$ *that*

$$\inf_{z \in [\theta^* - A_t, \theta^* + A_t]} \left[\frac{g(z)}{z - \theta^*}\right] \geq \frac{\delta}{A_t} \text{ for } t > N \tag{7}$$

where

$$A_t = |\theta^1 - \theta^*| + C(\alpha_1 + \cdots + \alpha_{t-1}). \tag{8}$$

Then $\lim_{t\to\infty} b_t = 0$.

Need to be clever to come up with a "new δ" to this Lemma. We do not have a lover limit on g(z) directly.

Proof of Theorem 3

$g(z)$ is nondecreasing;
$g(\theta^*) = 0$;
$g'(\theta^*) > 0$.

- $g'(\theta^*) = \lim_{x\to\theta^*} \frac{g(x)-g(\theta^*)}{x-\theta^*}$ imply

$$\frac{g(x)}{x-\theta^*} = g'(\theta^*) + \varepsilon(x-\theta^*), \quad \text{with } \lim_{t\to 0} \varepsilon(t) = 0$$

giving

$$\varepsilon(x-\theta^*) = \frac{g(x)}{(x-\theta^*)} - g'(\theta^*) \geq -\frac{1}{2} g'(\theta^*)$$

for $|x-\theta^*| < \delta$ and δ small enough. Thereby

$$\frac{g(x)}{x-\theta^*} \geq \frac{1}{2} g'(\theta^*), \quad \text{for } |x-\theta^*| \leq \delta$$

Since $g'(\theta^*) > 0$
we can choose a δ so small that the inequality is fulfilled for all values closer to θ^*

- For $\theta^* + \delta \leq x \leq \theta^* + A_t$, since $g(z)$ is nondecreasing

$$\frac{g(x)}{x-\theta^*} \geq \frac{g(x+\delta)}{A_t} \geq \frac{\delta g'(\theta^*)}{2A_t}$$

while for $\theta^* - A_t \leq x \leq \theta^* - \delta$

$$\frac{g(x)}{x-\theta^*} = \frac{-g(x)}{\theta^* - x} \geq \frac{-g(x-\delta)}{A_t} \geq \frac{\delta g'(\theta^*)}{2A_t}$$

$$A_t = |\theta^1 - \theta^*| + C(\alpha_1 + \cdots + \alpha_{t-1})$$

- Assuming (without loss of generality) $\delta/A_t \leq 1$ gives

$$\frac{g(x)}{x-\theta^*} \geq \frac{\delta g'(\theta^*)}{2A_t} \quad \text{for } 0 < |x-\theta^*| \leq A_t \Rightarrow \ (7)$$

Here
«δ» in Lemma 3
Is: $\frac{\delta g'(\theta^*)}{2}$
where «δ» is selected above

Stochastic gradients and neural nets

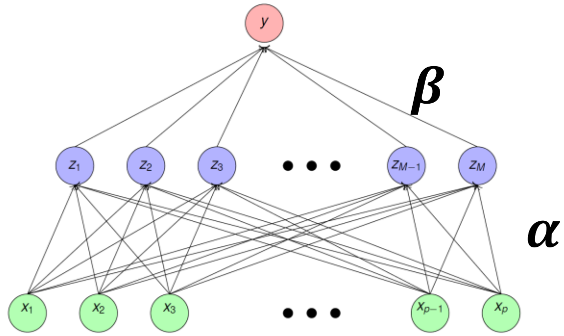
$$Q(\theta) = R(\theta) + \lambda J(\theta) \qquad R(\theta) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2$$

$$f(X) = \sum_{m=1}^{M_{NN}} \beta_m \sigma(\alpha_m^T X + \alpha_0)$$

- Q and their derivatives require a sum of n terms
- Can use a stochastic version by sampling randomly a subset of $\{1, \ldots, n\}$
- Called mini-batching
- Advantages (LeCun et al., 2012)
  - Much faster
  - Often give better solutions
  - Can be used to track changes
- Initial values (assuming $g(z) = z$):
  - Given $\boldsymbol{\alpha}$, the model is

$$y_i = \beta_0 + \boldsymbol{\beta}^T \mathbf{z}_i$$

  - Can obtain reasonable values of $\boldsymbol{\beta}$ through least squares
  - Random guess on $\boldsymbol{\alpha}$.
- `Stoch_grad_NN.R`



One hidden layer

Stochastic gradients and neural nets

- Many versions implemented
  - In R: ANN2::neuralnetwork, RSNNS::mlp, nnet::nnet, ...
- Typically, different tricks applied
  - Slow convergence: Fixed learning rate
  - SG through
    - randomly dividing data into minibatches
    - Updating sequentially on each minibach
    - epoch: One go through all data
  - Normalizing imput!
  - Momentum:

$$v^{t+1} = \gamma v^t + \alpha \nabla F(\theta^t)$$
$$\theta^{t+1} = \theta^t - v^{t+1}$$

  - Adaptive learning rates

$$\theta^{t+1} = \theta^t - \frac{\alpha}{\sqrt{||\nabla F(\theta^t)||^2 + \varepsilon}} \nabla F(\theta^t)$$

- Reference: LeCun et al. (2012)
- Example: `ANN2_zip.R`

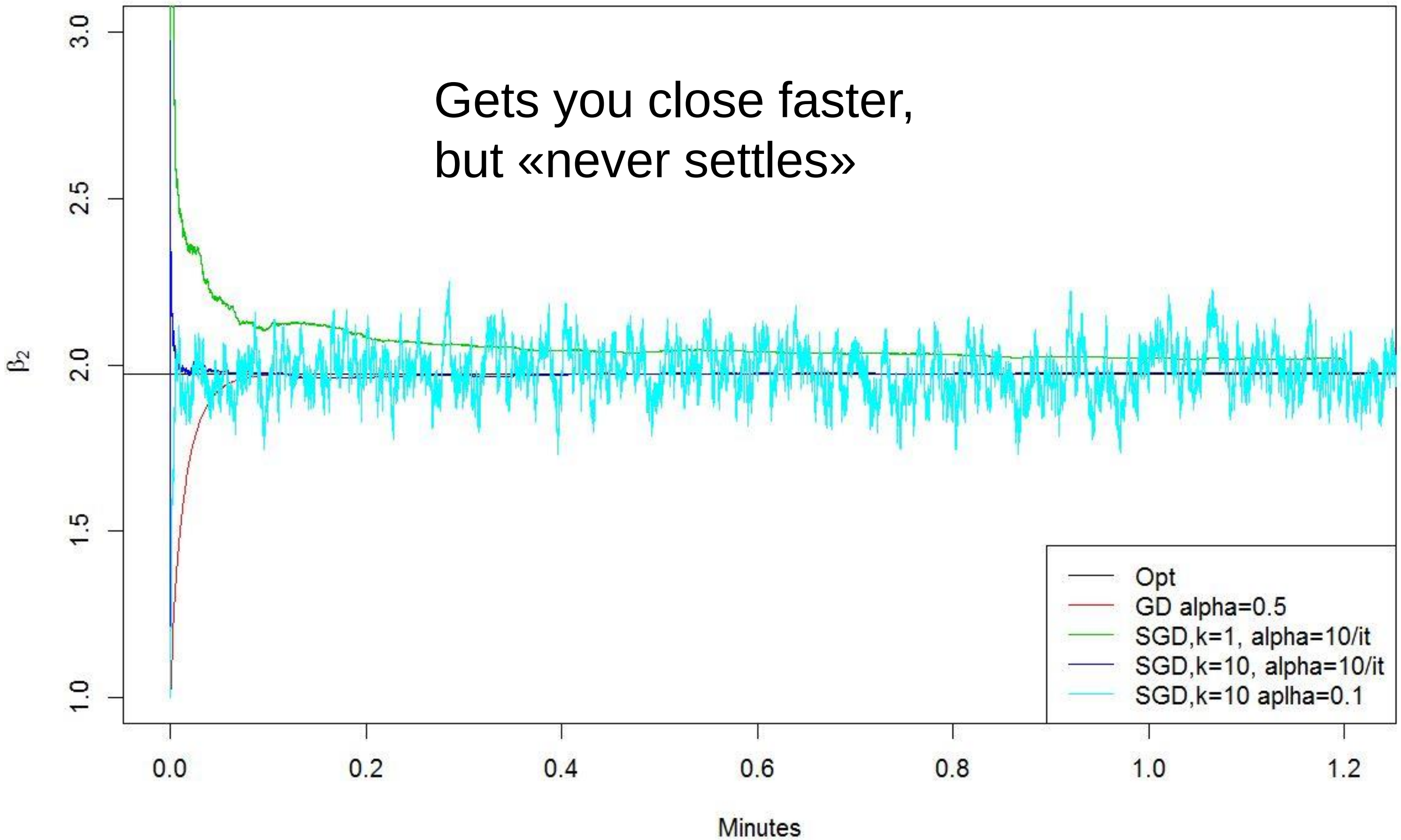
Questions?

- What is the convergence result for SGD?
  - If the function is sufficiently regular (A-4) & the stochastic gradient is unbiased and not too large and (A-5). SGD will converge to the optimum by choosing the learning rate according to (A-1), (A-2)& (A-3)

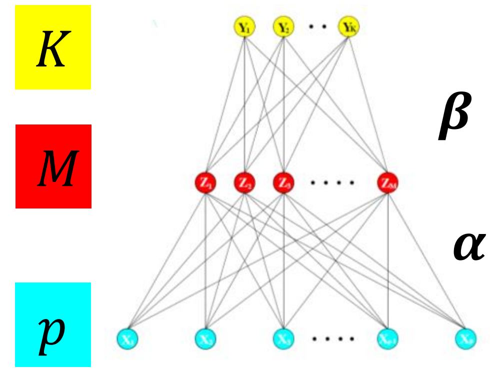
- Have we proven convergence of SGD for Neural Nets?
  - No, we haven't proven that the NN- function is well behaved

- It is common to use fixed step size when applying SGD for Neural Nets, what might be the reason for this?
  - It converges faster to something close to the optimum.
  - Do you need to get all the way to θ^* to have a good enough result?
  - Half the stepsize if the convergence stall…

Constant learning rate



Fitting neural networks



θ: Statistical slang for all parameters

Here:

$\{\alpha_{0,m}, \alpha_m\}$, # parameters: $(p+1)M$

$\{\beta_{0,m}, \beta_m\}$, # parameters: $(M+1)K$

Quadratic loss
K output variables

$$
\begin{aligned}
R(\theta) &= L\left(Y, \hat{f}(X)\right) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} \left(y_{ik} - \hat{f}_k(x_i)\right)^2 \\
&= \sum_{i=1}^{N} R_i(\theta)
\end{aligned}
$$

Contribution of
the i'th data record

$$
R_i(\theta) = \sum_{k=1}^{K} \left(y_{ik} - \hat{f}_k(x_i)\right)^2
$$

The “standard” approach:

- Minimize the loss
- Use steepest decent to solve this minimization problem

- The key to success is the efficient way of computing the gradient

Steepest decent

$$R(\theta) = \sum_{i=1}^{N} R_i(\theta)$$

- Minimize $R(\theta)$ wrt θ,
  - Initialize: $\theta^{(0)}$
  - Iterate:

$$\theta_j^{(r+1)} = \theta_j^{(r)} - \gamma_r \left.\frac{\partial R(\theta)}{\partial \theta_j}\right|_{\theta=\theta^{(r)}}$$

Learning rate

$$\frac{\partial R(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} \frac{\partial R_i(\theta)}{\partial \theta_j}$$

we compute term per data record (easily aggregated from parallel computation) $\frac{\partial R_i(\theta)}{\partial \theta_j}$

$$Here: g_k(T) = T_k$$
$$T_k = \beta_k^T z_i$$

Squared error loss

Output layer:

$$\frac{\partial R_i(\theta)}{\partial \beta_{k,m}} = \underbrace{-2\left(y_{i,k} - f_k(x_i)\right) g_k'\left(\beta_k^T z_i\right)}_{\delta_{k,i}} z_{m,i}$$
$$= \delta_{k,i} \cdot z_{m,i}$$

Hidden layer:

$$\frac{\partial R_i(\theta)}{\partial \alpha_{m,l}} = \underbrace{-\sum_{k=1}^{K} 2\left(y_{ik} - f_k(x_i)\right) g_k'\left(\beta_k^T z_i\right) \beta_{km}\, \sigma'(\alpha_m^T x_i)}_{s_{m,i}} x_{i,l}$$
$$= s_{m,i} \cdot x_{i,l}$$

Back propagation equation

$$s_{m,i} = \sigma'(\alpha_m^T x_i) \sum_{k=1}^{K} \beta_{km} \delta_{k,i}$$

Back propagation (delta rule)

- At top level. compute:

$$\delta_{k,i} = -2\left(y_{i,k} - f_k(x_i)\right) g_k'\left(\beta_k^T z_i\right), \qquad \forall (i,k)$$

- At hidden level, compute:

$$s_{m,i} = \sigma'(\alpha_m^T x_i) \sum_{k=1}^{K} \beta_{k,m} \delta_{k,i}\,, \qquad \forall (i,m)$$

- Evaluate:

$$\frac{\partial R_i(\theta)}{\partial \beta_{k,m}} = \delta_{k,i} z_{m,i} \;\;\&\;\; \frac{\partial R_i(\theta)}{\partial \alpha_{m,l}} = s_{m,i} x_{i,l}$$

- Update : γ_r is fixed

$$\beta_{k,m}^{(r+1)} = \beta_{k,m}^{(r)} - \gamma_r \sum_{i=1}^{N} \left.\frac{\partial R_i}{\partial \beta_{k,m}}\right|_{\theta=\theta^{(r)}}$$

$$\alpha_{m,l}^{(r+1)} = \alpha_{m,l}^{(r)} - \gamma_r \sum_{i=1}^{N} \left.\frac{\partial R_i}{\partial \alpha_{m,l}}\right|_{\theta=\theta^{(r)}}$$

Stochastic gradient decent

$$\beta_{k,m}^{(r+1)} = \beta_{k,m}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{k,m}}\bigg|_{\theta=\theta^{(r)}} \quad \alpha_{m,l}^{(r+1)} = \alpha_{m,l}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{m,l}}\bigg|_{\theta=\theta^{(r)}}$$

- Equations above updates with all data at the same time
- The form invites to update estimate using fractions of data
  - Perform a random partition of training data in to batches: $\{B_j\}_{j=1}^{\#\mathrm{Batches}}$
  - For all batches cycle over the data in this batch to update data

$$\beta_{k,m}^{(r+1)} = \beta_{k,m}^{(r)} - \gamma_r \sum_{i \in \mathrm{B}_j} \frac{\partial R_i}{\partial \beta_{k,m}}\bigg|_{\theta=\theta^{(r)}} \quad \alpha_{m,l}^{(r+1)} = \alpha_{m,l}^{(r)} - \gamma_r \sum_{i \in B_j} \frac{\partial R_i}{\partial \alpha_{m,l}}\bigg|_{\theta=\theta^{(r)}}$$

  - Repeat

- One **iteration** is one update of the parameter (using one batch)
- One **Epoch** is one scan through all data (using all batches in the partition)

Online learning (Batch size =1)

- Learning based on one data point at the time

$$\beta_{k,m}^{(r)} = \beta_{k,m}^{(r-1)} - \gamma_r \frac{\partial R_i}{\partial \beta_{k,m}}\bigg|_{\theta=\theta^{(r-1)}}$$
$$\alpha_{m,l}^{(r)} = \alpha_{m,l}^{(r-1)} - \gamma_r \frac{\partial R_i}{\partial \alpha_{m,l}}\bigg|_{\theta=\theta^{(r-1)}}$$

- You might re-iterate (for several epochs) when completed or if you have an abundance of data just take on new data as they come along (hence the name)

- For convergence: $\gamma_r \to 0$, as $\sum \gamma_r \to \infty$ and $\sum \gamma_r^2 < \infty$, e.g. $\gamma_r = \frac{1}{r}$ (as shown earlier)

SGD for dependent data

- Consider spatial data:

$$\boldsymbol{Y} = \begin{pmatrix} Y(s_1) \\ Y(s_2) \\ \vdots \\ Y(s_n) \end{pmatrix} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\mu} = \mu \boldsymbol{I}$ and $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{R} + \boldsymbol{\tau}^2 \boldsymbol{I}$, that is

$$\mathrm{cov}[Y(s_i), Y(s_j)] = \begin{cases} \sigma^2 r(||s_i - s_j||; \boldsymbol{\phi}) & s_i \neq s_j \\ \sigma^2 + \tau^2 & s_i = s_j \end{cases}$$

- Realisation of a process defined continuously in a space $\mathcal{S}$
- Log-likelihood with $\boldsymbol{\theta} = (\boldsymbol{\mu}, \sigma^2, \tau^2, \boldsymbol{\phi})$

$$l(\boldsymbol{\theta}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log(|\boldsymbol{\Sigma}|) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{\mu})$$

- In general, computational burden is $O(n^3)$, problematic for large n

ML and Kullback-Leibler divergence

- True distribution $g(y)$, assumed model $f_\theta(y)$
- Aim: Specify θ so that $f_\theta(y) \approx g(y)$
- Approach: Minimize Kullback-Leibler distance

$$
\begin{aligned}
KL(f_\theta, g) &= \int \log\left(\frac{g(y)}{f_\theta(y)}\right) g(y)dy \\
&= \int \log(g(y))g(y)dy - \int \log(f_\theta(y))g(y)dy \geq 0
\end{aligned}
$$

- Equivalent to maximize $\int \log(f_\theta(y))g(y)dy$, problem $g(y)$ unknown
- IID data: Approximate $g(y)$ by $\hat{g}(y) : \Pr(Y = y_i) = \frac{1}{n}$
  - Maximize $\sum_{i=1}^n \frac{1}{n} \log(f_\theta(y_i)) = \frac{1}{n}\ell(\theta)$
- Spatial data:

$$
\begin{aligned}
KL(f_\theta, g) &= \int\int \log\left(\frac{g(\boldsymbol{y}|\boldsymbol{s})}{f_\theta(\boldsymbol{y}|\boldsymbol{s})}\right) g(\boldsymbol{y}|\boldsymbol{s})g(\boldsymbol{s})d\boldsymbol{y}d\boldsymbol{s} \\
&= \int\int \log(g(\boldsymbol{y}|\boldsymbol{s}))g(\boldsymbol{y}|\boldsymbol{s})g(\boldsymbol{s})d\boldsymbol{y}d\boldsymbol{s} - \\
&\quad \int\int \log(f_\theta(\boldsymbol{y}|\boldsymbol{s}))g(\boldsymbol{y}|\boldsymbol{s})g(\boldsymbol{s})d\boldsymbol{y}d\boldsymbol{s}
\end{aligned}
$$

Not obvious how to approximate $g(\boldsymbol{y}, \boldsymbol{s}) = g(\boldsymbol{y}|\boldsymbol{s})g(\boldsymbol{s})$!

KL and Geostatistics

- We have one set of observations $\boldsymbol{y}$. Can approximate $g(\boldsymbol{y}, \boldsymbol{s})$ giving probability 1 to this.
    - Leads to the maximum (log-)likelihood approach
    - Has the computational burden mentioned earlier
    - Also has a problem in a poor description of g, lead to that ML estimate may not behave well!
- Liang et al. (2013): Approximate KL by

$$\widehat{KL}(f_\theta, g) = C - \frac{1}{\binom{n}{m}} \sum_{k=1}^{\binom{n}{m}} \log(f_\theta(\boldsymbol{y}_k | \boldsymbol{s}_k))$$

  where $(\boldsymbol{y}_k, \boldsymbol{s}_k)$ is a subset of $(\boldsymbol{y}, \boldsymbol{s})$ of size m.
- Find θ as the solution of

$$\frac{\partial}{\partial\theta}\widehat{KL}(f_\theta, g) = C - \frac{1}{\binom{n}{m}} \sum_{k=1}^{\binom{n}{m}} H(\theta, \boldsymbol{y}_k, \boldsymbol{s}_k)$$

$$H(\theta, \boldsymbol{y}_k, \boldsymbol{s}_k) = \frac{\partial}{\partial\theta} \log(f_\theta(\boldsymbol{y}_k | \boldsymbol{s}_k))$$

  by the stochastic gradient algorithm!

Example

$$\log f(\boldsymbol{y}_k|\boldsymbol{s}_k) = -\frac{m}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(y_k - \mu\mathbf{1}_{\boldsymbol{m}})^T \Sigma_k^{-1}(y_k - \mu\mathbf{1}_{\boldsymbol{m}})$$

$$(\boldsymbol{\Sigma}_k)_{i,j} = \text{cov}(Y(\boldsymbol{s}_{k,i}) - Y(\boldsymbol{s}_{k,j}) = \tau^2 I(j=j) + \sigma^2 \exp(-(||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi)$$
$$(\boldsymbol{R}_k)_{i,j} = \exp(-(||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi)$$

$$H_\mu(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = \mathbf{1}_m^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$
$$H_{\sigma^2}(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = -\frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\boldsymbol{R}_k) + \frac{1}{2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)^T \boldsymbol{\Sigma}_k^{-1}\boldsymbol{R}_k\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$
$$H_{\tau^2}(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = -\frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}) + \frac{1}{2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)^T \boldsymbol{\Sigma}_k^{-2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$
$$H_\phi(\boldsymbol{\theta}, \boldsymbol{y}_k, \boldsymbol{s}_k) = -\frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\frac{d\boldsymbol{R}_k}{d\phi}) + \frac{1}{2}(\boldsymbol{y}_k - \mu\mathbf{1}_m)^T \boldsymbol{\Sigma}_k^{-1}\frac{d\boldsymbol{R}_k}{d\phi}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_k - \mu\mathbf{1}_m)$$

$$\frac{d(\boldsymbol{R}_k)_{i,j}}{d\phi} = ||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi^2 \cdot \exp(-(||\boldsymbol{s}_{k,i} - \boldsymbol{s}_{k,j}||/\phi)$$