

Numerical Optimisation:
Least squares

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat [edu_assist_pro](#)

Department of Computer Science
Centre for Medical Image Computing
Centre for Inverse Problems
University College London

Lecture 10 & 11

Least squares is a problem where the objective function has the following special form

Assignment Project Exam Help

m

—

<https://eduassistpro.github.io>

where

j

each of the r_j as a *residual*, and we assume that

Add WeChat edu_assist_pro

Least squares problems are ubiquitous in applications where the discrepancy between the model and the observed data is minimised.

Let's assemble the individual components r_j into the *residual vector* $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$r(x) = (r_1(x), r_2(x), \dots, r_m(x))^T.$$

Using this vector, f becomes $f(x) = \frac{1}{2} \|r(x)\|_2^2$. The derivatives of f can be calculated with help of the Jacobian

$$\nabla r_1(x)^T$$

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

$$\begin{aligned}\nabla f(x) &= \sum_{j=1}^m r_j(x) \nabla r_j(x) = \\ \nabla^2 f(x) &= \sum_{j=1}^m \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \\ &= J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x),\end{aligned}$$

Example

Model of concentration of drug in bloodstream

$$\phi(x; t) = x_1 + tx_2 + t^2x_3 + x_4 \exp^{-x_5 t}.$$

Find a set of parameters x_0 that the model best matches the data solving

$$\frac{1}{m}$$

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Figure: Nocedal Wright Fig 10.1 (left), Fig 10.2 (right)

Bayes' theorem

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} \propto \pi(y|x)\pi(x).$$

Assignment Project Exam Help

Den

at da

<https://eduassistpro.github.io>

Assume that ϵ_j s are independent and identic

variance σ^2 and probability density function

of a particular set of observations y_j, j

parameter vector x is given by

$$\pi(y|x) = \prod_{j=1}^m g_{\sigma}(\epsilon_j) = \prod_{j=1}^m g_{\sigma}(\phi(x; t_j) - y_j).$$

The *maximum a posteriori probability* (MAP) estimate vs the *maximum likelihood* estimate

$$x_{\text{MAP}} = \max_x \pi(x|y) = \max_x \pi(y|x)\pi(x).$$

Assignment Project Exam Help

If g_σ is a normal distribution

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

and reads

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

$$x_{\text{MAP}} = \max_x \frac{1}{(\sqrt{2\pi}\sigma)^m} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^m (x - y_j)^2\right)$$

$$= \min_x \sum_{j=1}^m (\phi(x; t_j) - y_j)^2.$$

If $\phi(x; t)$ is linear function in x , the least squares problem becomes linear.

The re
nota

<https://eduassistpro.github.io>

where

- the vector of measurements $y \in \mathbb{R}^n$
- the matrix J with rows $J_{j,:} = \phi(x; t_j)$

are both independent of x .

The linear least squares has the form

$$f(x) = \frac{1}{2} \|Jx - y\|^2.$$

Assignment Project Exam Help

The gradient and Hessian are

<https://eduassistpro.github.io>

Note: $f(x)$ is convex i.e. the stationary point is the
minimiser $\nabla f(x^*) = 0$

Add WeChat edu_assist_pro

Normal equations

$$\nabla f(x^*) = J^T(Jx^* - y) = 0 \quad \Leftrightarrow \quad J^T Jx^* = J^T y.$$

Roadmap: solution of linear least squares

- Solve the normal equations $J^T J x^* = J^T y$
 - + If $m \gg n$, computing $J^T J$ explicitly results in a smaller matrix easier to store than J . This can be solved by e.g. Cholesky decomposition.
 - Formulating $J^T J$ squares the condition number.

<https://eduassistpro.github.io>

- Solve the least squares $x^* = \arg \min_x$
 - If J is of moderate size you can use direct methods like QR decomposition or SVD decomposition.
 - + If J is large and sparse or given in operator form, you can use iterative methods like CGLS or LSQR.
 - + Does not square the condition number.
 - + In particular SVD and iterative methods e.g. LSQR can easily deal with ill-conditioning.

Let

$$JP = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \left[\underbrace{Q_1}_{\in \mathbb{R}^{m \times n}} \underbrace{Q_2}_{\in \mathbb{R}^{m \times (m-n)}} \right] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R,$$

where



$n \times n$



<https://eduassistpro.github.io>



Recall: Multiplication with orthogonal mat

Add WeChat edu_assist_pro

$$\begin{aligned} \|Jx - y\|_2^2 &= \|Q^T(JPP^T x - y)\|_2^2 = \|(Q \ JP)P^T x - Q^T y\|_2^2 \\ &= \|RP^T x - Q_1^T y\|_2^2 + \|Q_2^T y\|_2^2. \end{aligned}$$

Solution: $x^* = PR^{-1}Q_1^T y$. In practice we perform backsubstitution on $Rz = Q_1^T y$ and permute for $x^* = Pz$.

Singular value decomposition

Let

$$J = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S \\ 0 \end{bmatrix} V^T = U_1 S V^T$$

$\underbrace{\begin{bmatrix} U_1 & U_2 \end{bmatrix}}_{\in \mathbb{R}^{m \times n}} \underbrace{\begin{bmatrix} S \\ 0 \end{bmatrix}}_{\in \mathbb{R}^{n \times (m-n)}}$

where

- <https://eduassistpro.github.io>
- $n > 0$.

$$\begin{aligned} \|Jx - y\|_2^2 &= \| \overbrace{U^T J V^T}^{=I} x - y \|_2^2 \\ &= \| S V^T x - U_1^T y \|_2^2 + \| U_2^T y \|_2^2 \end{aligned}$$

Solution: $x^* = V S^{-1} U_1^T y = \sum_{i=1}^n \frac{u_i^T y}{\sigma_i} v_i$. If σ_i are small, they would undue amplify the noise and can be omitted from the sum.

Picard condition: $|u_i^T y|$ should decay faster than σ_i .

LSQR applied to

$$\min_f \|Af - g\|_2^2 + \tau \|f\|_2^2 = \min_f \left\| \begin{bmatrix} A \\ \sqrt{\tau}I \end{bmatrix} f - \begin{bmatrix} g \\ 0 \end{bmatrix} \right\|_2^2,$$

where

equi

avoid

the G

G-K bidiagonalisation yields the projected least

$$\min_{y_i} \left\| \begin{bmatrix} B_i \\ \sqrt{\tau}I \end{bmatrix} y_i - \beta_1 e_1 \right\|, \quad (\text{P-LS})$$

which is then solved using QR decomposition yielding the approximation for the solution of the original problem, $f_i = V_i y_i$.

G-K bidiagonalization with a starting vector g for $\min_f \|g - Af\|$

Assignment Project Exam Help

$$\begin{aligned} U_{i+1}(\beta_1 e_1) &= g \\ AV_i &= U_{i+1}B_i \\ A^T U_{i+1} &= V_i B^T + \alpha_{i+1} v_{i+1} e^T, \end{aligned}$$

$e_i: i$ <https://eduassistpro.github.io> \hat{v}_i

$$B_i = \begin{bmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \alpha_i & \\ & & & \beta_{i+1} & \end{bmatrix}, \quad U_i = [u_1, \dots, u_i]$$

Add WeChat edu_assist_pro

(1)

Preconditioned LSQR

Assignment Project Exam Help

$$\hat{f} = \operatorname{argmin} \|g - AL^{-1}\hat{f}\|.$$

The c
prec

<https://eduassistpro.github.io>

$$L^{-T}A^TAL^{-1}\hat{f} = L^{-T}A^Tg \quad (2)$$

Similarly as for CG the LSQR algorithm can be for
without the need to provide a factorization
algorithm [Arridge, B, Harhanen '14].

Add WeChat [edu_assist_pro](#)

1: **Initialization:**

2: $\beta_1 u_1 = g$

3: $\tilde{p} = A^T u_1$

4: $\tilde{v}_1 = M^{-1} \tilde{p}, \alpha_1 = (\tilde{v}_1, \tilde{p})^{1/2}, \tilde{v}_1 = \tilde{v}_1 / \alpha_1$

5: $\tilde{w}_1 = \tilde{v}_1, f_0 = 0, \phi_1 = \beta_1, \bar{\rho}_1 = \alpha_1$

6: **f**

7:

8:

9:

10: $\tilde{v}_{i+1} = M^{-1} \tilde{p}, \alpha_{i+1} = (\tilde{v}_{i+1}, \tilde{p})^{1/2},$

11: $\tilde{p} = \tilde{p} / \alpha_{i+1}, \tilde{v}_{i-1} = \tilde{v}_{i-1} / \alpha_{i+1}$

12: **Orthogonal transformation:** arr

13: **Update:**

14: $f_i = f_{i-1} + (\phi_i / \rho_i) \tilde{w}_i$

15: $\tilde{w}_{i+1} = \tilde{v}_{i+1} - (\theta_{i+1} / \rho_i) \tilde{w}_i$

16: **Break if stopping criterion satisfied**

17: **end for**

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

LSQR with explicit regularization ($\tau \neq 0$)

- In preconditioned formulation, Tikhonov (explicit) regularization amounts to damping. For a fixed value of τ , damping can be easily incorporated in LSQR at the cost of doubling the number of Givens rotations [Paige, Saunders '82].
- Due to the shift invariance of Krylov spaces, V_i are the same

<https://eduassistpro.github.io>

- Solving (P-LS) with a variable τ using singular value decomposition of the B_i (the efficient SVD update even though row and a column in each iteration). Those q_i obtained at the cost $\mathcal{O}(i^2)$ at the i^{th} iteration.
- For larger i , the algorithm described in [Elden '77] for the least squares solution of (P-LS) at the cost of $\mathcal{O}(i)$ for each value of τ is the preferable option.

Stopping LSQR / MLSQR

[Saunders Paige '82] discusses three stopping criteria:

S1: $\|\bar{r}_i\| \leq \text{BTOL}\|g\| + \text{ATOL}\|\bar{A}\|\|f_i\|$ (consistent systems),

S2: $\frac{\|\bar{A}^T \bar{r}_i\|}{\|\bar{A}\|\|\bar{r}_i\|} \leq \text{ATOL}$ (inconsistent systems)

S3: $\text{cond}(\bar{A}) \geq \text{CONLIM}$ (both)

where

[Arri

(suitable for ill-posed problems)

S4: $\|r_i\| \leq \eta\delta$, $\eta > 1$,

where $r_i := g - A\bar{x}_i$, δ is the (estimated) noise level, η is the overregularization

- + if $\tau = 0$, $r_i = \bar{r}_i$ and the sequence $\|r_i\| = \|\bar{r}_i\|$ is monotonically decreasing. Moreover if initialised with f_0 , $\|f_i\|$ is strictly monotonically growing (relevant for damped problem),
- + priorconditioning does not alter the residual.

Gauss-Newton (GN) method

Gauss-Newton (GN) can be viewed as a modified Newton method with line search.

Recall the specific form of gradient and Hessian of least squares problems

$$\nabla f$$

m

$$^2 r_j(x).$$

Subs

$$\nabla^2 f(x_k) p_k = -$$

and using the approximation $\nabla^2 f(x_k)$

$$\underbrace{J(x_k)^T}_{=: J_k^T} J(x_k) p_k^{\text{GN}} = -J(x_k) \underbrace{r(x_k)}_{=: r_k}.$$

Implementations of GN usually perform a line search along p_k^{GN} requiring the step length to satisfy e.g. Armijo or Wolfe conditions.

- Does not require computation of the individual Hessians $\nabla^2 r_j, j = 1, \dots, m$. If the Jacobian J_k has been computed when evaluating the gradient no other derivatives are needed.

- Frequently the first term $J_k^T J_k$ dominates the second term in the Hessian i.e. $\|r_j(x)\|$ are much smaller than the eigenvalues $J^T J$. This happens if either the residual r_j or $\nabla^2 f_k$

- <https://eduassistpro.github.io> or f and hence suitable for line search

$$\begin{aligned} (p_k^{\text{GN}})^T \nabla^2 f_k &= (p_k^{\text{GN}})^T J_k^T J_k r_k \\ &= -\|J_k p_k^{\text{GN}}\|^2 \end{aligned}$$

The final inequality is strict unless $J_k p_k^{\text{GN}} = 0$ in which case by the GN equation and J_k being full-rank we have $0 = J_k^T r_k = \nabla f_k$ and x_k is a stationary point.

The GN equation

$$J_k^T J_k p_k^{\text{GN}} = -J_k^T r_k$$

is exactly the normal equation for the linear least squares problem

Assignment Project Exam Help

Hence
squares

<https://eduassistpro.github.io>

We can view GN equation as obtained from a linear m
vector function $r(x_k + p) \approx r_k + J_k p$,

Add WeChat: edu_assist_pro

$$f(x_k + p) = \frac{1}{2} \|r_k(x_k + p)\|^2 \approx \frac{1}{2} \|J_k p + r_k\|^2$$

and $p_k^{\text{GN}} = \arg \min_p \frac{1}{2} \|J_k p + r_k\|^2$.

Global convergence of GN

The global convergence is a consequence of the convergence theorem for line search methods [Zoutendijk].

Assignment Project Exam Help
To satisfy the conditions of Zoutendijk's theorem, we need to make following assumptions:

-

- <https://eduassistpro.github.io>

$$\|J(x)z\| \geq \gamma \|z\|,$$

Then for the iterates x_k generated by the length satisfying Wolfe conditions, we have

$$\lim_{k \rightarrow \infty} J_k^T r_k = \nabla f(x_k) = 0.$$

Similarly as for the line search, we check that the angle $\theta_k = \angle(p_k^{\text{GN}}, -\nabla f_k)$ is uniformly bounded away from $\pi/2$

$$\cos \theta_k = \frac{(p_k^{\text{GN}})^T \nabla f_k}{\|p_k^{\text{GN}}\| \|\nabla f_k\|} = \frac{\|J_k p_k^{\text{GN}}\|^2}{\|p_k^{\text{GN}}\|^2 \|J_k\|^2} \geq \frac{\gamma^2 \|p_k^{\text{GN}}\|^2}{\beta^2 \|p_k^{\text{GN}}\|^2} = \frac{\gamma^2}{\beta^2} > 0$$

where

SS

of the r_j

<https://eduassistpro.github.io>

Then from $\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty$ follows $\|\nabla f_k\| \rightarrow 0$.

Add WeChat edu_assist_pro

If J_k for some k is rank deficient, the matrix J_k is singular and the system has infinitely many solutions, however $\cos \theta_k$ is not necessarily bounded away from 0.

Convergence rate GN

The convergence of GN can be rapid if $J_k^T J_k$ dominates the second term in the Hessian. Similarly as showing the convergence rate of Newton iteration, if x_k is sufficiently close to x^* , $J(x)$ satisfies the uniform full rank condition, we have for a unit step in GN direction

$$:= J(x_k)^T J(x_k)$$

$x_k +$

<https://eduassistpro.github.io>

Using $H(x)$ to denote the second term in the Hessian from Taylor theorem that

$$\begin{aligned}\nabla f(x_k) - \nabla f(x^*) &= \int_0^1 J^T J(x^* + t(x_k - x^*))(x_k - x^*) dt \\ &\quad + \int_0^1 H(x^* + t(x_k - x^*))(x_k - x^*) dt,\end{aligned}$$

Putting everything together and assuming Lipschitz continuity of $H(\cdot)$ near x^* and using L.c.d. of $r_j \Rightarrow$ L.c. of $J^T r(x)$

$$\|x_k + p_k^{\text{GN}} - x^*\| \leq \int_0^1 \| [J^T J(x_k)]^{-1} H(x^* + t(x_k - x^*)) \| \|x_k - x^*\| dt$$

$$+ \int_0^1 \frac{\| [J^T J(x_k)]^{-1} (J^T J(x^* + t(x_k - x^*)) - J^T J(x_k)) \| \|x_k - x^*\| dt}{\| \cdot \| \leq \gamma^{-2} \quad \| \cdot \| = \mathcal{O}(\|x_k - x^*\|) \text{ would need L.c. of } J^T J(x)}$$

$\| \cdot \|^2$).

Hence
quic

is quadratic (Newton).

When the Jacobian $J(x)$ is large and sparse, the equation can be replaced by an *inexact* solve as in inexact Newton methods but with the true Hessian $\nabla^2 f(x_k)$ replaced with $J(x_k)^T J(x_k)$. The positive semidefiniteness of $J(x_k)^T J(x_k)$ simplifies the algorithms. Instead of (preconditioned) CG, (preconditioned) LSQR should be used.

Levenberg-Marquardt (LM) method

Levenberg-Marquardt (LM) makes use of the same Hessian approximation as GN but within the framework of trust region methods. Trust region methods can cope with (nearly) rank-deficient Hessian, which is a weakness of GN.

The c

<https://eduassistpro.github.io> (CM-LM)

where $\Delta_k > 0$ is the trust region radius.

Note: The least squares term corresponds to q

$$m_k(p) = \frac{1}{2} \|r_k\|^2 + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p.$$

Solution of the constraint model problem

The solution of the constraint model problem (CM-LM) is an immediate consequence of the general result for trust region methods [More, Sorensen]:

- If the solution p_k^{GN} of the GN equation lies strictly inside the trust region i.e. $p_k^{\text{GN}} < \Delta_k$, then $p_k^{\text{LM}} = p_k^{\text{GN}}$ solves

- <https://eduassistpro.github.io>
(CM-LM) satisfies $\|p_k\| = \Delta_k$ and

$$(J_k^T J_k + \lambda I) p_k^{\text{LM}}$$

Note. The last equation is the normal equation \pm squares problem

$$\min_p \frac{1}{2} \left\| \begin{bmatrix} J_k \\ \sqrt{\lambda} I \end{bmatrix} p + \begin{bmatrix} r \\ 0 \end{bmatrix} \right\|^2,$$

which gives us a way of solving (CM-LM) without computing $J_k^T J_k$.

Global convergence LM

Global convergence is a consequence of the corresponding trust region global convergence theorem.

To satisfy the conditions of that theorem we make the following assumptions:

- $\eta \in (0, \frac{1}{4})$ (for strong convergence)

-

- <https://eduassistpro.github.io>

$$m_k(0) - m_k(p_k) \geq c_1 \|J_k^T r_k\|$$

for some constant $c_1 > 0$, and in addition
some constant $\gamma \geq 1$.

We then have that

$$\lim_{k \rightarrow \infty} \nabla f_k = \lim_{k \rightarrow \infty} J_k^T r_k = 0.$$

- As for trust region methods, there is no need to evaluate the right hand side of the decrease condition, but it is sufficient to ensure reduction of at least the Cauchy point, which can be calculated inexpensively. If the iterative CG-Steihaus

- <https://eduassistpro.github.io>
solution x^* , at which the first term $J(x^*) - J(x)$ of the Hessian $\nabla^2 f(x^*)$ dominates the second term. The algorithm takes G local convergence.

- [Paige, Saunders '82] Link to website with papers and codes
<https://web.stanford.edu/group/SOL/software/lsqr/>

- [Björck '96] A. Björck, "Numerical Methods for Least Squares Problems", SIAM, 1996



- <https://eduassistpro.github.io>

- ill-conditioned least squares problems"

- [Aldridge, B, Harhanen '14] S. R. Aldridge, M. M. B. Harhanen, "Iterated preconditioned L problems on unstructured grids", *Inverse Problems*, 30(7) 2014

- [Hansen '98] P. C. Hansen "Rank-Deficient and Discrete Ill-Posed Problems", SIAM, 1998