THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Statistics and Data Sciences
College of Natural Sciences

SDS 385: Stat Models for Big Data

Lecture 2: Linear Regression

Purnamrita Sarkar
Department of Statistics and Data Science
The University of Texas at Austin

https://psarkar.github.io/teaching

Linear regression: recap

Given n pairs $(\boldsymbol{x}_i, y_i) \in \Re^{p+1\times 1}$, consider the model:

$$\boldsymbol{y} = \boldsymbol{X}\beta + \boldsymbol{\epsilon} \qquad \epsilon_i \sim N(0, \sigma^2)$$

where:

$$\boldsymbol{y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \text{and } \boldsymbol{x} = \begin{bmatrix} 1 & x_{12} & \dots & x_{1p} \\ 1 & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n2} & \dots & x_{np} \end{bmatrix}$$

- $\boldsymbol{X}, \boldsymbol{y}$ are given, you need to estimate $\boldsymbol{\beta}$.

MLE - recap

$$f(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\beta}) \propto \exp(\frac{-(\boldsymbol{y}-\boldsymbol{X}\beta)^T(\boldsymbol{y}-\boldsymbol{X}\beta)}{2\sigma^2})$$

- Take Log, we can get:

$$\frac{-(\boldsymbol{y}-\boldsymbol{X}\beta)^T(\boldsymbol{y}-\boldsymbol{X}\beta)}{2\sigma^2} \tag{1}$$

- Same drill – differentiate and set it to zero.

$$-\boldsymbol{X}^T(\boldsymbol{y}-\boldsymbol{X}\hat{\beta}) = 0 \rightarrow \boldsymbol{X}^T\boldsymbol{X}\hat{\beta} = \boldsymbol{X}^T\boldsymbol{y} \rightarrow \hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- What happens when $p \gg n$? $\boldsymbol{X}^T\boldsymbol{X}$ is not invertible.

Ridge regression

- Add a prior to β, i.e. $\boldsymbol{\beta} \sim N(0, \lambda \boldsymbol{I}_p)$, or think of adding a regularization that penalizes large values of $\boldsymbol{\beta}^T \boldsymbol{\beta}$.
- So now we have:

$$f(\boldsymbol{y}|\boldsymbol{X}, \beta) \propto \exp\left(\frac{-(\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta)}{2\sigma^2} - \lambda \boldsymbol{\beta}^T \boldsymbol{\beta}\right)$$

- Differentiating and setting to zero gives:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

- Phew! – no issues with invertibility of $\boldsymbol{X}^T \boldsymbol{X}$

Exact computation

- If $\boldsymbol{X}$ was dense, how much time would the computation of $\boldsymbol{X}^T\boldsymbol{X}$ take?
- Wait, what is dense?
- Well dense means, $\boldsymbol{X}$ has about $\Theta(np)$ non-zero elements.

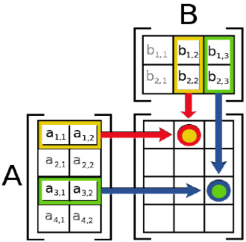


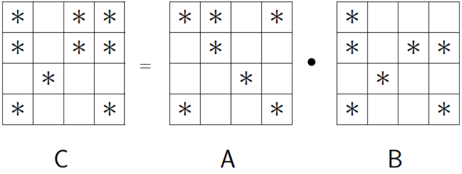
Figure 1: Dense matrix multiplication[1]

- So $O(n)$ computation for each of p^2 entries, and hence np^2.

[1] Borrowed from Cho-Jui Hsieh's classnotes at UC-Davis.

Sparse matrix data structures

- How do you store a sparse vector?
- All you need is two vectors: one is of the indices of nonzero elements and one is the values.



$$C_{ij} = \sum_k A_{ik} \cdot B_{kj}$$

Figure 2: Sparse matrix multiplication[2]

- If A has nnz non-zeroes, then worst case, the complexity is $O(n \times nnz)$ operations for multiplying a sparse matrix with another dense matrix.

[2] Borrowed from Grey Ballard and Alex Druinsky, SIAM conf. on Lin. Algebra

Back to regression

- Inverting a $p \times p$ matrix takes $O(p^3)$ time.
- Alternatives: use linear solvers of the form $A\boldsymbol{u} = \boldsymbol{v}$.
- Here $A = \boldsymbol{X}^T\boldsymbol{X} + \lambda I_p$, $\boldsymbol{v} = \boldsymbol{X}^T\boldsymbol{y}$ and $\boldsymbol{u} = \boldsymbol{\beta}$.
- Unless your matrix A has some structure, linear solvers can also be expensive. However, if it does have structure, e.g. its diagonally dominant, etc, then there are nearly linear time solvers.
- Typically for regression, we don't expect to have such structure.
- So, what can be done?

Iterative solvers

- Lets talk about gradient descent type methods.
- Model: $\sum_{i=1}^{n} f(\underbrace{x_i}_{data}; \underbrace{\beta}_{parameter})$
- Example of f: negative log-likelihood over iid data-points, e.g. linear regression, logistic regression, etc.
- Goal: $\hat{\beta} = \arg\min_{\beta} f(x_i; \beta)$
- Lets deal with convex loss functions.

Convex functions

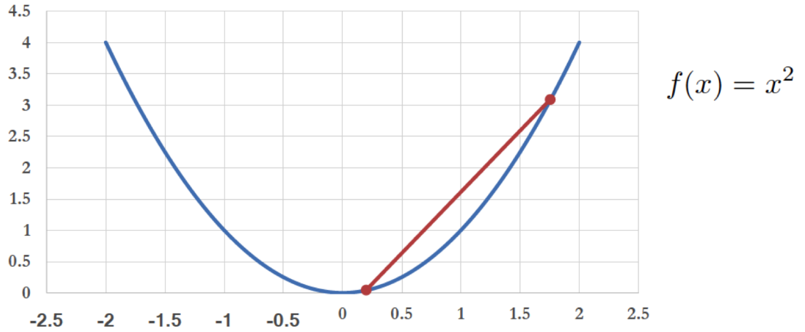


Figure 3: A convex function

$$\forall \alpha \in [0, 1], f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

Quadratic function $f(y) = y^2$

$$
\begin{aligned}
f(\alpha x + (1-\alpha)y) &= (\alpha x + (1-\alpha)y)^2 \\
&= \alpha^2 x^2 + (1-\alpha)^2 y^2 + 2\alpha(1-\alpha)xy \\
&\leq \alpha^2 x^2 + (1-\alpha)^2 y^2 + \alpha(1-\alpha)(x^2 + y^2) \\
&= \alpha x^2 + (1-\alpha)y^2
\end{aligned}
$$

- Where did I use $\alpha \in [0, 1]$?

Convex functions $f(y) = |y|$

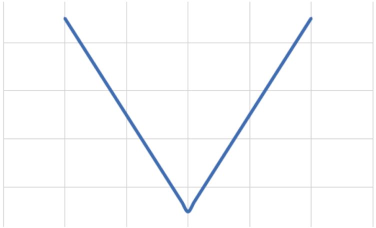


Figure 4: $f(y) = |y|$

$$\begin{aligned} f(\alpha x + (1-\alpha)y) &= |\alpha x + (1-\alpha)y| \\ &\leq |\alpha x| + |(1-\alpha)y| \\ &\leq \alpha|x| + (1-\alpha)|y| \end{aligned}$$

Local optima is also global optima

Theorem

Consider an optimization problem $\min_x f(x)$ *where* f *is convex. Let* x^* *be a local minima. Prove that it is also a global minima.*

Proof.

- By definition, $\exists p > 0$, such that $\forall x \in B(x^*, p)$, $f(x) \geq f(x^*)$.
- If x^* is not the global optima, $z \notin B(x^*, p)$ such that $f(z) < f(x^*)$.
- Take $t \in [0, 1]$ and the point $y = tx^* + (1 - t)z$.
  $f(y) \leq tf(x^*) + (1 - t)f(z) < f(x^*)$
- Now $|y - x^*| = (1 - t)|z - x^*|$. If we take t large enough such that $(1 - t)|z - x^*| \leq p$, then $y \in B(x^*, p)$ but $f(y) < f(x^*)$, which is a contradiction.

$\square$

Properties of convex functions

- Non-negative combinations of convex functions is also convex.
- A convex function composed with an affine function is also convex.
- Point-wise maxima of convex functions is convex.

Properties of convex functions

- Compositions of convex functions not necessarily convex
- f, g convex.
    - Is $f - g$ convex?
    - Is fg convex?

Convex functions: other definitions

- First order:

$$\langle x - y, \nabla f(x) - \nabla f(y) \rangle \geq 0$$

- Second order:

$$\nabla^2 f(x) \succeq 0$$

- Example: $f(x) = x^2$. $(x - y)^2 > 0$ and $f''(x) = 2 \geq 0$.
- Example: $f(x) = \log(1 + e^x)$.
  - $f'(x) = \dfrac{1}{1 + e^{-x}}$ is monotonically increasing with x and so the first order condition is satisfied.
  - Second order: $f''(x) = f(x)(1 - f(x)) \geq 0$

Strongly convex functions – add curvature

- First order:

$$\langle x - y, \nabla f(x) - \nabla f(y) \rangle \geq \mu \|x - y\|^2$$

- Second order:

$$\nabla^2 f(x) \succeq \mu I$$

- So you add a margin to each inequality.

Gradient descent

$$\beta \leftarrow \beta - \alpha \nabla f(\beta)$$

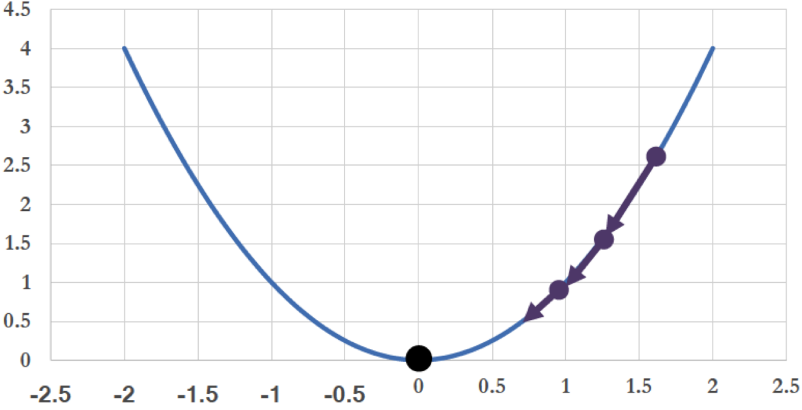


Figure 5: Convex function minimization with gradient descent

Step size

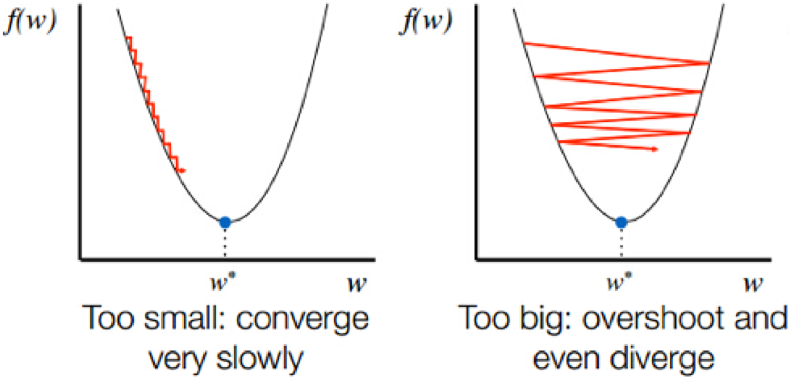


Figure 6: Choice of step size is crucial

Newton Raphson

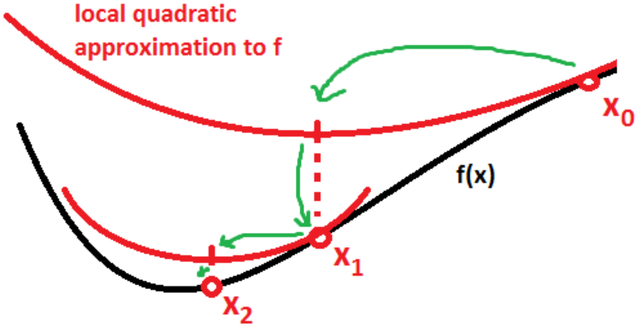


Figure 7: Newton Raphson[4]

[4] Borrowed from Nick Alger, math.stackexchange.com

Newton Raphson cont.

- GD takes into account only first order information.
- NR also takes second order information.
- In particular it uses the Hessian,

$$H(i,j) = \frac{\partial^2 f}{\partial \theta_i \partial \theta_j}, \text{ where } i,j \in \{1,\ldots,k\},$$

- Lets try to minimize a quadratic function:

$$f = \mathbf{a} + \mathbf{b}^T \boldsymbol{\theta} + \frac{1}{2}\boldsymbol{\theta}^T C \boldsymbol{\theta}.$$

- C is positive semidefinite and so this is a convex function.
- We can minimize the function by differentiating it and by setting the result equal to 0:

$$\nabla f(\boldsymbol{\theta}^*) = \mathbf{b} + C\boldsymbol{\theta}^* = 0$$
$$\boldsymbol{\theta}^* = -C^{-1}\mathbf{b}$$

Newton Raphson cont.

- In the neighborhood of $\boldsymbol{\theta}_t$, we can use the approximation:

$$f(\boldsymbol{\theta}^{(t)} + \mathbf{h}) \approx f(\boldsymbol{\theta}^{(t)}) + \nabla f(\boldsymbol{\theta}^{(t)})^T \mathbf{h} + \frac{1}{2}\mathbf{h}^T H(\boldsymbol{\theta}^{(t)})\mathbf{h}. \tag{2}$$

- Therefore the general updating rule is

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - H^{-1}(\boldsymbol{\theta}^{(t)}) \cdot \nabla f(\boldsymbol{\theta}^{(t)})$$

- You can use a stepsize here as well.

Gradient Descent

- for $t = 1 : T$ (or until convergence)
- Do $\beta_{t+1} \leftarrow \beta_t - \alpha \nabla f(\beta)$

Theorem

Let β^* is the global minima, and the second derivative is bounded as $\mu I \preceq H(\beta) \preceq LI$. Then with $\alpha = 2/(L + \mu)$, gradient descent converges geometrically, i.e.

$$\|\beta_{t+1} - \beta^*\| \leq \frac{L - \mu}{L + \mu} \|\beta_t - \beta^*\|$$

Proof

- Lets look at the distance from the optima:

$$\begin{aligned}\beta_{t+1} - \beta^* &= \beta_t - \beta^* - \alpha(\nabla f(\beta_t) - \nabla f(\beta^*)) \\ &= \beta_t - \beta^* - \alpha H(z_t)(\beta_t - \beta^*) \\ &= (I - \alpha \nabla^2 f(z_t))(\beta_t - \beta^*)\end{aligned}$$

- Now take norm of both sides and use Triangle.

$$\begin{aligned}\|\beta_{t+1} - \beta^*\| &\leq \|I - \alpha H(z_t)\| \|\beta_t - \beta^*\| \\ &\leq \max(|1 - \alpha\mu|, |1 - \alpha L|) \|\beta_t - \beta^*\|\end{aligned}$$

Linear convergence

- Set $\alpha = 2/(L+\mu)$. You get

$$\|\beta_{t+1} - \beta^*\| \leq \frac{L-\mu}{L+\mu}\|\beta_t - \beta^*\|$$

- Finally after T iterations, we have:

$$\|\beta_{t+1} - \beta^*\| \leq \left(\frac{L-\mu}{L+\mu}\right)^T \|\beta_t - \beta^*\|$$

- This is a typical "linear" contraction result.

Newton Raphson

Theorem

Let β^* *is the global minima, and the second derivative is L Lipschitz, i.e.* $\|H(x) - H(x')\| \leq \kappa \|x - x'\|$ *and* $\|H^{-1}\| \leq 1/\mu$. *Then with* $\alpha = 1$, *Newton Raphson converges quadratically, i.e.*

$$\|\beta_{t+1} - \beta^*\| \leq \kappa/\mu \|\beta_t - \beta^*\|^2$$

- Note that this is useful only when $\|\beta_{t+1} - \beta^*\| \ll 1$

Proof

$$
\begin{aligned}
\beta_{t+1} - \beta^* &= \beta_t - \beta^* - H^{-1}(\beta_t)(\nabla f(\beta_t) - \nabla f(\beta^*)) \\
&= \beta_t - \beta^* - H^{-1}(\beta_t)H(z_t)(\beta_t - \beta^*) \\
&= (I - H^{-1}(\beta_t)H(z_t))(\beta_t - \beta^*) \\
&= H^{-1}(\beta_t)(H(\beta_t) - H(z_t))(\beta_t - \beta^*) \\
\|\beta_{t+1} - \beta^*\| &\leq \|H^{-1}(\beta_t)\| \|H(\beta_t) - H(z_t)\| (\beta_t - \beta^*) \\
&\leq \kappa/\mu \|\beta_t - z_t\| (\beta_t - \beta^*) \\
&\leq \kappa/\mu \|\beta_t - \beta^*\|^2
\end{aligned}
$$

Scalability concerns

- You have to calculate the gradient every iteration.
- Take ridge regression.
- You want to minimize $1/n\left((\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta) - \lambda\beta^T\beta\right)$
- Take a derivative: $(-2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta) - 2\lambda\beta)/n$
- Grad descent update takes $\boldsymbol{\beta}_{t+1} \leftarrow \boldsymbol{\beta}_t + \alpha(\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_t) + \lambda\boldsymbol{\beta}_t)$
- What is the complexity?
    - Trick: first compute $\boldsymbol{y} - \boldsymbol{X}\beta$.
    - np for matrix vector multiplication, $nnz(\boldsymbol{X})$ for sparse matrix vector multiplication.
    - Remember the examples with humongous n and p?

So what to do?

- For $i = 1 : T$
  - Draw i with replacement from n
  - $\beta_{t+1} = \beta_t - \alpha \nabla f(x_{\sigma_i}; \beta_t)$
- In expectation (over the randomness of the index you chose), for a fixed β,

$$E[f(x_{\sigma_i}; \beta)] = \frac{\sum_i f(x_i; \beta)}{n}$$

- Does this also converge?

Convergence

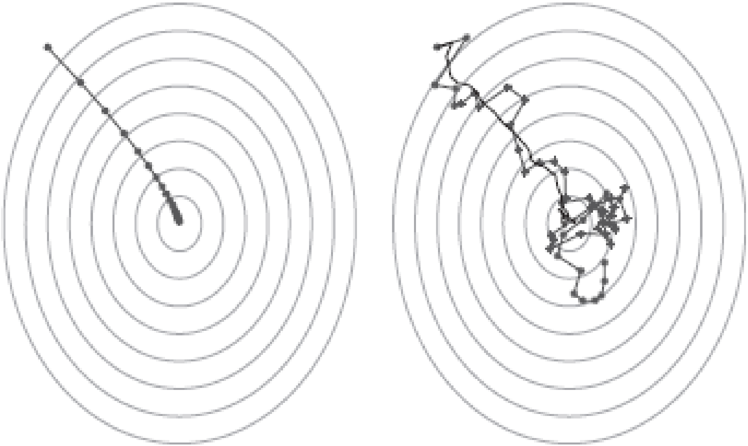


Figure 8: Gradient descent vs Stochastic gradient descent

Convergence

- Let $\nabla f(X;\beta)$ be the full derivative.

$$
\begin{aligned}
&\beta_{t+1} - \beta^* \\
&= \beta_t - \beta^* - \alpha \nabla f(x_{\sigma_i};\beta_t) \\
&= \beta_t - \beta^* - \alpha(\nabla f(X;\beta_t) - \nabla h(X;\beta^*)) + \alpha(\nabla f(X;\beta_t) - \nabla f(x_{\sigma_i};\beta_t)) \\
&= \underbrace{(I - \alpha \nabla^2 h(z_t))(\beta_t - \beta^*)}_{g(\beta_t)} + \alpha \underbrace{(\nabla f(X;\beta_t) - \nabla f(x_{\sigma_i};\beta_t))}_{h(\sigma_i,\beta_t)}
\end{aligned}
$$

- Take the expected squared length:

$$
E[\|\beta_{t+1} - \beta^*\|^2 | \beta_t] = \underbrace{\|g(\beta_t)\|^2}_{\text{Same as before}} + \alpha^2 \underbrace{E[\|h(\sigma_i,\beta_t)\|^2 | \beta_t]}_{\substack{\text{variance of} \\ \text{gradient update} \\ \text{at a random point}}}
$$

SGD cont.

- So by total expectation rule,

$$
\begin{aligned}
E[\|\beta_{t+1} - \beta^*\|^2] &\leq (1 - \alpha\mu)^2 E[\|\beta_t - \beta^*\|^2] + \alpha^2 C \\
\lim_{t\to\infty} E[\|\beta_{t+1} - \beta^*\|^2] &\leq \frac{\alpha M}{2\mu - \alpha\mu^2}
\end{aligned}
$$

- So SGD is converging to a noise ball.
- How to remedy this?

SGD stepsize

- Assume you are far away from the noise ball.
- $\|\beta_t - \beta^*\|^2 \geq 2\alpha M/\mu$.
- Then,

$$
\begin{aligned}
E[\|\beta_{t+1} - \beta^*\|^2 | \beta_t] &\leq (1-\alpha\mu)^2 \|\beta_t - \beta^*\|^2 + \frac{\alpha\mu}{2}\|\beta_t - \beta^*\|^2 \\
&\leq \left(1 - \frac{\alpha\mu}{2}\right)\|\beta_t - \beta^*\|^2 \qquad \text{If } \alpha\mu < 1 \\
E[\|\beta_T - \beta^*\|^2] &\leq e^{-\alpha\mu T/2} C
\end{aligned}
$$

- It takes $2/\alpha\mu \log M$ steps to achieve M factor contraction.

Tradeoff

- Recall that the size of the noise ball is

$$\lim_{t \to \infty} E[\|\beta_{t+1} - \beta^*\|^2] \leq \frac{\alpha M}{2\mu - \alpha\mu^2}$$

- So the size is $O(\alpha)$, i.e. for larger α we converge to a larger noise ball.
- But convergence time is $2/\alpha\mu \log M$, i.e. inversely proportional to step size *alpha*.
- So there is a tradeoff.

Acknowledgment

Cho-Jui Hsieh and Christopher De Sa's large scale ML classes.