Tutorials 3 and 4

Monday 22^{nd} August, 2016

Problem 1. Equivalence between Ridge Regression and Bayesian Linear Regression (with fixed σ^2 and λ):

Consider the Bayesian Linear Regression Model

$$y = \mathbf{w}^T\phi(\mathbf{x}) + \varepsilon \; \textit{and} \; \varepsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\mathbf{w} \sim \mathcal{N}(0, \alpha I) \; \textit{and} \; \mathbf{w} \mid \mathcal{D} \sim \mathcal{N}(\mu_m, \Sigma_m)$$
$$\mu_m = (\lambda\sigma^2 I + \phi^T\phi)^{-1}\phi^T\mathbf{y} \; \textit{and} \; \Sigma_m^{-1} = \lambda I + \phi^T\phi/\sigma^2$$

Show that $\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}} \; \Pr(\mathbf{w} \mid \mathcal{D})$ is the same as that of *Regularized Ridge Regression.*

$$\mathbf{w}_{Ridge} = \underset{\mathbf{w}}{\operatorname{argmin}} \; ||\phi\mathbf{w} - \mathbf{y}||_2^2 + \lambda\sigma^2||\mathbf{w}||_2^2$$

In other words, The Bayes and MAP estimates for Linear Regression coincide with that of *Regularized Ridge Regression.*

Solution Sketch: Taking the negative log of the log likelihood we see that maximizing the log of the posterior distribution is equivalent to minimizing the ridge regression objective.

$$\Pr(\mathbf{w} \mid \mathcal{D}) = \mathcal{N}(\mathbf{w} \mid \mu_m, \Sigma_m) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_m|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{w}-\mu_m)^T\Sigma_m^{-1}(\mathbf{w}-\mu_m)}$$

$$-\log\Pr(\mathbf{w}) = \cancel{\frac{n}{2}\log(2\pi)} + \cancel{\frac{1}{2}\log|\Sigma_m|} + \frac{1}{2}\underbrace{(\mathbf{w}-\mu_m)^T\Sigma_m^{-1}(\mathbf{w}-\mu_m)}_{}$$

independent of ω

$\cancel{\frac{1}{2}\mu_m^T\Sigma_m^{-1}\mu_m}$

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}} -\log\Pr(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}}\frac{1}{2}\mathbf{w}^T\Sigma_m^{-1}\mathbf{w} - \mathbf{w}^T\Sigma_m^{-1}\mu_m$$

that is, (substituting μm & Σ_m^{-1}) You can drop terms ind. of ω for argmax but NOT for max

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\frac{1}{2}\mathbf{w}^T\left(\lambda I + \phi^T\phi/\sigma^2\right)\mathbf{w} - \mathbf{w}^T\left(\lambda\cancel{I + \phi^T\phi}/\sigma^2\right)\left((\lambda\cancel{\sigma^2 I + \phi^T}\phi)^{-1}\phi^T\mathbf{y}\right)$$

and after expanding and canceling out redundant terms, and later, after completing squares:

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\frac{1}{2\sigma^2}\mathbf{w}^T\left(\phi^T\phi\mathbf{w} - 2\phi^T\mathbf{y}\right) + \lambda\mathbf{w}^T\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmax}}\frac{1}{2}||\phi\mathbf{w} - \mathbf{y}||^2 + \sigma^2\lambda||\mathbf{w}||^2 = \mathbf{w}_{Ridge}$$

1 $\frac{1}{2}y^Ty$

You can add terms independent of ω in argmax

Problem 2. Ridge Regression and Error Minimization:

1. *Prove the following Claim:*
   The sum of squares error on training data using the weights obtained after minimizing ridge regression objective is greater than or equal to the sum of squares error on training data using the weights obtained after minimizing the ordinary least squares (OLS) objective.

   More specifically, if ϕ and $\mathbf{y}$ are defined on the training set $\mathcal{D} = \{(\mathbf{x}_1, y_1)...(\mathbf{x}_m, y_m)\}$ as

$$\phi = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & ...... & \phi_n(\mathbf{x_1}) \\ . & & & \\ . & & & \\ \phi_1(\mathbf{x}_m) & \phi_2(\mathbf{x}_m) & ...... & \phi_n(\mathbf{x}_m) \end{bmatrix} \tag{1}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ . \\ y_m \end{bmatrix} \tag{2}$$

   and if

$$\mathbf{w}_{Ridge} = \underset{\mathbf{w}}{\text{argmin}} \ ||\phi\mathbf{w} - \mathbf{y}||_2^2 + \lambda||\mathbf{w}||_2^2$$

and

$$\mathbf{w}_{OLS} = \underset{\mathbf{w}}{\text{argmin}} \ ||\phi\mathbf{w} - \mathbf{y}||_2^2$$

then you should prove that

$$||\phi\mathbf{w}_{Ridge} - \mathbf{y}||_2^2 \geq ||\phi\mathbf{w}_{OLS} - \mathbf{y}||_2^2$$

Proof: If

$$\mathbf{w}_{OLS} = \underset{\mathbf{w}}{\text{argmin}} \ ||\phi\mathbf{w} - \mathbf{y}||_2^2$$

then by definition of argmin,

$$||\phi\mathbf{w}_{Ridge} - \mathbf{y}||_2^2 \geq ||\phi\mathbf{w}_{OLS} - \mathbf{y}||_2^2$$

→ That is sum of squares error is greater on training set using soln to ridge (than soln to OLS)

Also, one can reformulate

$$\mathbf{w}_{Ridge} = \underset{\mathbf{w}}{\text{argmin}} \ ||\phi\mathbf{w} - \mathbf{y}||_2^2 + \lambda||\mathbf{w}||_2^2$$

as

$$\mathbf{w}_{Ridge} = \underset{\mathbf{w}}{\text{argmin}} \ ||\phi\mathbf{w} - \mathbf{y}||_2^2$$

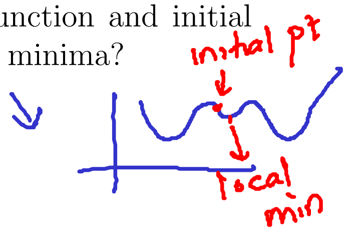
$$\textit{such that } ||\mathbf{w}||_2^2 \leq \theta$$

for some θ corresponding to a value of λ. The solution to a constrained minimization problem will always be greater than or equal to its unconstrained counterpart.

2. If it is the case that ridge regression leads to greater error than ordinary least squares regression, then why should one be interested in ridge regression at all?

   Answer: This is still acceptable since ridge regression incorporates prior (as per Bayesian interpretation). The idea is ultimately to do well on unseen (test) data as well. Therefore, high training error might be acceptable if test error can be lowered.

Problem 3. Gradient descent is a very helpful algorithm. But it is not guaranteed to converge to global minima always. Give an example of a continuous function and initial point for which gradient descent converges to a value which is not global minima?



Problem 4. Step Length Considerations

1. Consider the function

$$f(x) = x_1^2 - 4x_1 + 2x_1x_2 + 2x_2^2 + 2x_2 + 14$$

   This function has a minimum at $x = (5, 3)$. Suppose you are at a point $(4, -4)^T$ after few iterations, using the **exact line search algorithm** discussed in the class, find the point for the next iteration.

2. Now consider solving the Least Squares Linear Regression problem using the gradient descent algorithm. And let us say $w^{(0)} = 0$ and that the step length $t^{(k)}$ is computed using exact line search for each value of k. In how many steps will the gradient descent algorithm converge? What would be your answer if we had a different initialization for $w^{(0)}$?

   Solution:

$$t^{(k)} = \operatorname*{argmin}_t \left(\mathbf{w^{(k)}} + \mathbf{2t}\left(\phi^{\mathbf{T}}\mathbf{y} - \phi^{\mathbf{T}}\phi\mathbf{w^{(k)}} - \lambda\mathbf{w^{(k)}}\right)\right) \tag{3}$$

Problem 5. Suppose you are solving the equation $A\mathbf{x} = \mathbf{b}$ using gradient descent on least squares solution. How do you think the Eigenvalues of the matrix affect the convergence? (Hint: Consider a 2x2 diagonal matrix for A what do you observe?)

[Source : Quora]

Solution: https://www.quora.com/Why-is-the-Speed-Of-Convergence-of-gradient-descent-depends-on-the-maximal-and-minimal-eigenvalues-of-A-in-solving-AX-b-t

Look at the countours of the objective $||A\mathbf{x} - \mathbf{b}||^2$. The larger is the ratio $\frac{\lambda_{max}(A)}{\lambda_{min}(A)}$, the more skewed are the level curves and more is the time gradient descent will take for convergence. Thus, the matrix A with small value of $\frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ is always desirable.

In general, by the Courant-Fischer min-max Theorem, if A and B are two $n \times n$ symmetric matrices, and suppose the k^{th} largest eigenvalue of matrix X is $\lambda_k(X)$, $k = 1, 2, , n$: $\lambda_1(X) \geq \lambda_2(X) \ldots \geq \lambda_n(X)$ then

$\lambda_k(A) + \lambda_n(B) \leq \lambda_k(A + B) \leq \lambda_k(A) + \lambda_1(B)$

$\lambda_1(A) \geq \lambda_2(A) \cdot - \geq \lambda_n(A)$

$\lambda_1(B) \geq \lambda_2(B) \; \cdot \cdot \cdot \geq \lambda_n(B)$

$\lambda_1(A+B) \geq \lambda_2(A+B) \cdot \cdot \geq \lambda_n(A+B)$

In our setting

$$\underbrace{\phi^T\phi}_{A} + \underbrace{\lambda I}_{B} = \text{Hessian } \nabla^2 f$$

$\lambda_1(B) = \lambda_n(B) = \lambda$

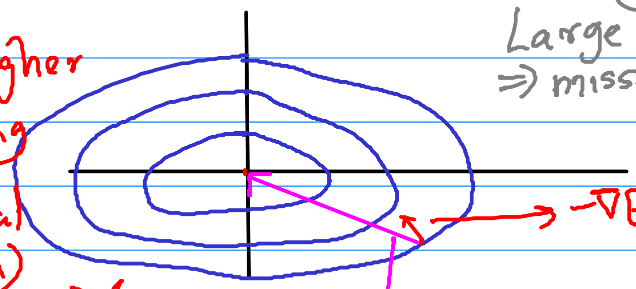
$\lambda_k(\phi^T\phi + \lambda I) = \lambda_k(\phi^T\phi) + \lambda$

For an understanding of how increased $\lambda(A)$ results in increased value of x^TAx, see page 207 of
https://www.cse.iitb.ac.in/~cs725/notes/classNotes/misc/LinearAlgebra.pdf

For 2 dims
$A=\lambda_1 q_1 q_1^T + \lambda_2 q_2 q_2^T$

$\|Ax-b\|^2$ Contours / level curves

- ∇E will have higher chance of pointing toward the global min if $\frac{\lambda_{max}(A)}{\lambda_{min}(A)} \to 1$

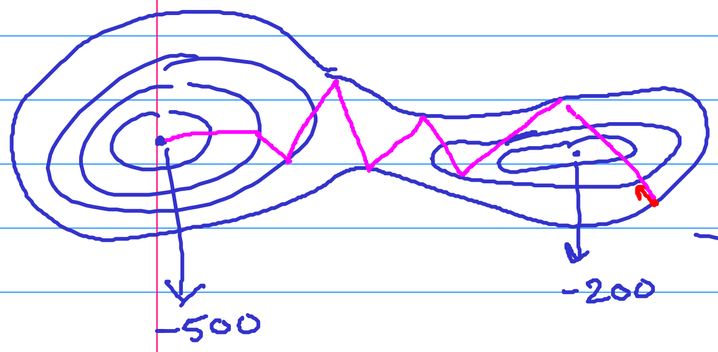


(a) Large step t ⇒ missing global min by miles

(b) small step t ⇒ long time to local & miss global

$-\nabla E$ = local view of direction of steepest descent

global view of direction of steepest decrease



→ Classic case is of deep neural networks. [Step size called learning rate]

For any symmetric A, $A=\sum \lambda_i q_i q_i^T$

q_i's are diff axes & λ_i's are lengths

q_i's are orthonormal

Regularize $\theta\|x\|^2$ tends to decrease $\frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ to $\frac{\lambda_{max}(A)+\theta}{\lambda_{min}(A)+\theta}$ thus helping gradient descent!