Introduction to Machine Learning - CS725
Instructor: Prof. Ganesh Ramakrishnan
Lecture 4 - Linear Regression - Bayesian Inference and Regularization

Building on questions on Least Squares Linear Regression

1. Is there a probabilistic interpretation?
   - Gaussian Error, Maximum Likelihood Estimate
2. Addressing overfitting
   - Bayesian and Maximum Aposteriori Estimates, Regularization
3. How to minimize the resultant and more complex error functions?
   - Level Curves and Surfaces, Gradient Vector, Directional Derivative, Gradient Descent Algorithm, Convexity, Necessary and Sufficient Conditions for Optimality

Prior Distribution over $\mathbf{w}$ for Linear Regression

$$y = \mathbf{w}^T \phi(x) + \varepsilon$$

$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$y \sim \mathcal{N}(\mathbf{w}^T \phi(x), \sigma^2)$$

- We saw that when we try to maximize log-likelihood we end up with $\hat{\mathbf{w}}_{MLE} = (\phi^T \phi)^{-1} \phi^T \mathbf{y}$
- We can use a Prior distribution on $\mathbf{w}$ to avoid over-fitting

  $\mu_0 = 0$ ← $w_i \sim \mathcal{N}(0, \frac{1}{\lambda})$ → $1/\lambda = (\Sigma_0)_{ii}$

  (that is, each component w_i is approximately bounded within $\pm \frac{3}{\sqrt{\lambda}}$ by the $3 - \sigma$ rule)
- We want to find $P(\mathbf{w}|D) = \mathcal{N}(\mu_m, \Sigma_m)$

  Invoking the Bayes Estimation results from before:

$\Sigma^{-1} = \frac{1}{\sigma^2} \phi^T \phi$ & $\Sigma_0 = \frac{1}{\lambda} I \Leftrightarrow \Sigma_m^{-1} = \Sigma_0^{-1} + m\Sigma^{-1}$

$\mu = 0$ & $\mu_0 = 0$ $\Leftrightarrow \Sigma_m^{-1} \mu_m = m\Sigma^{-1} \hat{\mu}_{MLE} + \Sigma_0^{-1} \mu_0$

$$\underline{w} \sim N(\mu_0, \Sigma_0) = N\left(0, \frac{1}{\lambda} I\right)$$

$$P(w|D) \propto \underline{P(D|w)}\ \underline{P(w)}$$

$$\frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^m} \prod_{i=1}^{m} \exp\left[\frac{-1}{2\sigma^2}(y_i - w^T\phi(x_i))^2\right] = N(\mu, \Sigma)$$

$$= \qquad\qquad (\quad -\mu)^T \Sigma^{-1} (\quad -\mu))$$

Homework: Complete the derivation

Prior Distribution over $\mathbf{w}$ for Linear Regression

$$y = \mathbf{w}^T \phi(x) + \varepsilon$$
$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

- We saw that when we try to maximize log-likelihood we end up with $\hat{\mathbf{w}}_{MLE} = (\phi^T \phi)^{-1} \phi^T \mathbf{y}$
- We can use a Prior distribution on $\mathbf{w}$ to avoid over-fitting

  $$w_i \sim \mathcal{N}(0, \tfrac{1}{\lambda})$$

  (that is, each component w_i is approximately bounded within $\pm \frac{3}{\sqrt{\lambda}}$ by the $3 - \sigma$ rule)
- We want to find $P(\mathbf{w}|D) = \mathcal{N}(\mu_m, \Sigma_m)$
  Invoking the Bayes Estimation results from before:

$$\Sigma_m^{-1} \mu_m = \Sigma_0^{-1} \mu_0 + \phi^T \mathbf{y} / \sigma^2$$
$$\Sigma_m^{-1} = \Sigma_0^{-1} + \frac{1}{\sigma^2} \phi^T \phi$$

$[\Sigma^{-1} = \frac{1}{\sigma^2} \phi^T \phi]$

Finding μ_m & Σ_m for $\mathbf{w}$

Setting $\Sigma_0 = \frac{1}{\lambda} I$ and $\mu_0 = \mathbf{0}$

$$\Sigma_m^{-1} \mu_m = \phi^T \mathbf{y} / \sigma^2$$
$$\Sigma_m^{-1} = \lambda I + \phi^T \phi / \sigma^2$$
$$\mu_m = \frac{(\lambda I + \phi^T \phi / \sigma^2)^{-1} \phi^T \mathbf{y}}{\sigma^2}$$

or

$$\mu_m = (\lambda \sigma^2 I + \phi^T \phi)^{-1} \phi^T \mathbf{y}$$

If $\lambda = 0 \Rightarrow$ No prior belief abt λ

$$\mu_m = (0 + \phi^T \phi)^{-1} \phi^T y = (\phi^T \phi)^{-1} \phi^T y = w_{LS}$$

MAP and Bayes Estimates

- $\Pr(\mathbf{w} \mid \mathcal{D}) = \mathcal{N}(\mathbf{w} \mid \mu_m, \Sigma_m)$
- The **MAP estimate** or mode under the Gaussian posterior is the mode of the posterior $\Rightarrow$

$$\hat{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}} \mathcal{N}(\mathbf{w} \mid \mu_m, \Sigma_m) = \mu_m$$

$\lambda \to 0 \Rightarrow w_{MAP} = w_{MLE}$

- Similarly, the **Bayes Estimate**, or the expected value under the Gaussian posterior is the mean $\Rightarrow$

$$\hat{w}_{Bayes} = E_{\Pr(\mathbf{w}|\mathcal{D})}[\mathbf{w}] = E_{\mathcal{N}(\mu_m, \Sigma_m)}[\mathbf{w}] = \mu_m$$

- Summarily:

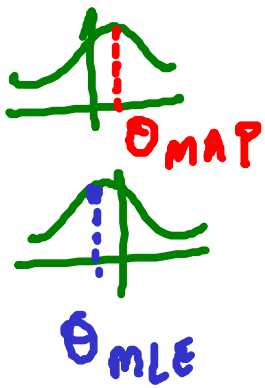
$$\mu_{MAP} = \mu_{Bayes} = \mu_m = (\lambda\sigma^2 I + \phi^T\phi)^{-1}\phi^T\mathbf{y}$$

$$\Sigma_m^{-1} = \lambda I + \frac{\phi^T\phi}{\sigma^2}$$

$\phi = \begin{bmatrix} \phi(x_1) \\ \phi(x_2) \\ \phi(x_m) \end{bmatrix}$

From Bayesian Estimates to (Pure) Bayesian Prediction

Most important

	Point?	$p(x\|D)$
MLE	$\hat{\theta}_{MLE} = \text{argmax}_\theta \, LL(D\|\theta)$	$p(x\|\theta_{MLE}) = N(w_{MLE}^T \phi(x), \sigma^2)$
Bayes Estimator	$\hat{\theta}_B = E_{p(\theta\|D)} E[\theta]$	$p(x\|\theta_B) = N(w_B^T \phi(x), \sigma^2)$
MAP	$\hat{\theta}_{MAP} = \text{argmax}_\theta \, p(\theta\|D)$	$p(x\|\theta_{MAP}) = N(w_{MAP}^T \phi(x), \sigma^2)$
Pure Bayesian (Quite ambitious)		$p(\theta\|D) = \frac{p(D\|\theta)p(\theta)}{\int_m p(D\|\theta)p(\theta)d}$ dθ $p(D\|\theta) = \prod_{i=1}^{n} p(x_i\|\theta)$ $p(x\|D) = \int_\theta p(x\|\theta)p(\theta\|D)$ dθ

where θ is the parameter

Averaging over all models/parameters

For Bernoulli with Beta prior:

$p_{MAP} = \frac{h+\alpha-1}{n+\alpha+\beta-2}$ $p_{Bayes} = \frac{h+\alpha}{n+\alpha+\beta}$

Predictive distribution for linear Regression

- $\hat{\mathbf{w}}_{MAP}$ helps avoid overfitting as it takes regularization into account
- But we miss the modeling of uncertainty when we consider only $\hat{\mathbf{w}}_{MAP}$
- **Eg:** While predicting diagnostic results on a new patient x, along with the value y, we would also like to know the uncertainty of the prediction $\Pr(y \mid x, D)$. Recall that $y = \mathbf{w}^T \phi(x) + \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

$$\Pr(y \mid \mathbf{x}, \mathcal{D}) = \Pr(y \mid \mathbf{x}, < \mathbf{x}_1, y_1 > \ldots < \mathbf{x}_m, y_m >)$$

Pure Bayesian Regression Summarized

- By definition, regression is about finding $(y \mid \mathbf{x}, < \mathbf{x}_1, y_1 > \ldots < \mathbf{x}_m, y_m >)$
- By Bayes Rule

$\mathbf{x}$ = Not a r.v & hence preceded by "j"

$$\Pr(y \mid \mathbf{x}, \mathcal{D}) = \Pr(y \mid \mathbf{x}, < \mathbf{x}_1, y_1 > \ldots < \mathbf{x}_m, y_m$$

$$= \int_{\mathbf{w}} \Pr(y|\mathbf{w}; \mathbf{x}) \Pr(\mathbf{w} \mid \mathcal{D}) d\mathbf{w}$$

$$\sim \mathcal{N}\left(\mu_m^T \phi(\mathbf{x}), \sigma^2 + \phi^T(\mathbf{x}) \Sigma_m \phi(\mathbf{x})\right)$$

where

$$y = \mathbf{w}^T \phi(\mathbf{x}) + \varepsilon \text{ and } \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$\mathbf{w} \sim \mathcal{N}(0, \alpha I) \text{ and } \mathbf{w} \mid \mathcal{D} \sim \mathcal{N}(\mu_m, \Sigma_m)$$

$$\mu_m = (\lambda \sigma^2 I + \phi^T \phi)^{-1} \phi^T \mathbf{y} \text{ and } \Sigma_m^{-1} = \lambda I + \phi^T \phi / \sigma^2$$

Finally $y \sim \mathcal{N}(\mu_m^T \phi(\mathbf{x}), \phi^T(\mathbf{x}) \Sigma_m \phi(\mathbf{x}))$

Penalized Regularized Least Squares Regression

- The Bayes and MAP estimates for Linear Regression coincide with *Regularized Ridge Regression*

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

- **Intuition:** To discourage redundancy and/or stop coefficients of $\mathbf{w}$ from becoming too large in magnitude, add a penalty to the error term used to estimate parameters of the model.
- The general **Penalized Regularized L.S Problem**:

$$\mathbf{w}_{Reg} = \arg\min_{\mathbf{w}} \; ||\phi\mathbf{w} - \mathbf{y}||_2^2 + \lambda\Omega(\mathbf{w})$$

  - $\Omega(\mathbf{w}) = ||\mathbf{w}||_2^2 \Rightarrow$ **Ridge Regression**
  - $\Omega(\mathbf{w}) = ||\mathbf{w}||_1 \Rightarrow$ **Lasso**
  - $\Omega(\mathbf{w}) = ||\mathbf{w}||_0 \Rightarrow$ **Support-based penalty**

- Some $\Omega(\mathbf{w})$ correspond to priors that can be expressed in close form. Some give good working solutions. However, for mathematical convenience, some norms are easier to handle

Constrained Regularized Least Squares Regression

- **Intuition:** To discourage redundancy and/or stop coefficients of $\mathbf{w}$ from becoming too large in magnitude, constrain the error minimizing estimate using a penalty
- The general **Constrained Regularized L.S. Problem**:

$$\mathbf{w}_{Reg} = \arg\min_{\mathbf{w}} \ ||\phi\mathbf{w} - \mathbf{y}||_2^2$$

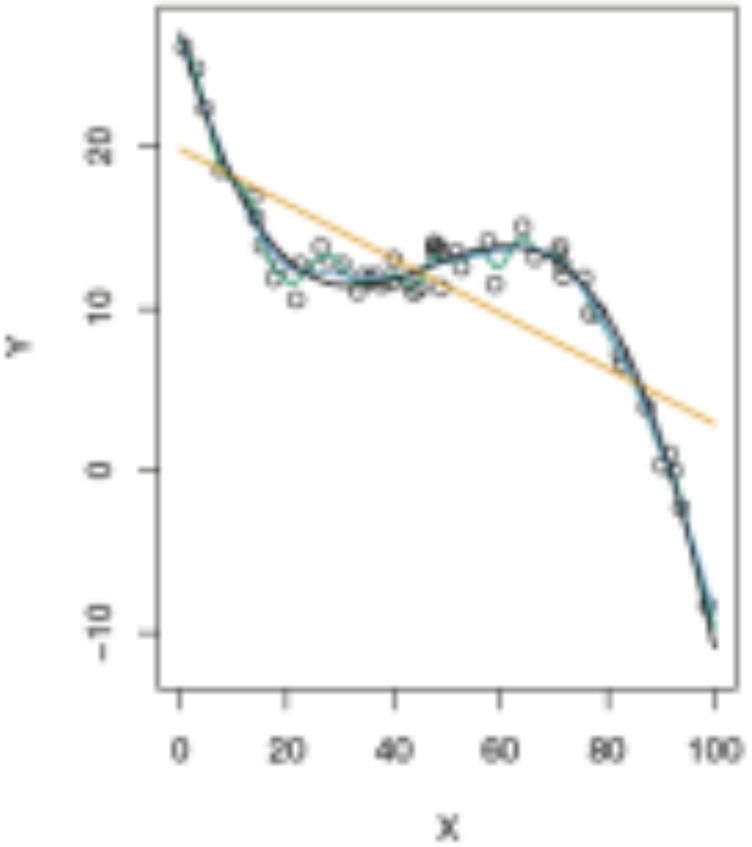
$$\textit{such that } \Omega(\mathbf{w}) \leq \theta$$

- Claim: For any Penalized formulation with a particular λ, there exists a corresponding Constrained formulation with a corresponding θ
  - $\Omega(\mathbf{w}) = ||\mathbf{w}||_2^2 \Rightarrow$ **Ridge Regression**
  - $\Omega(\mathbf{w}) = ||\mathbf{w}||_1 \Rightarrow$ **Lasso**
  - $\Omega(\mathbf{w}) = ||\mathbf{w}||_0 \Rightarrow$ **Support-based penalty**

: we will later see $\theta \leftrightarrow \lambda$ mapping for ridge regression

- **Proof of Equivalence:** Requires tools of Optimization/duality

Polynomial regression



- Consider a degree 3 polynomial regression model as shown in the figure
- Each bend in the curve corresponds to increase in $\|w\|$
- Eigen values of $(\phi^\top \phi + \lambda I)$ are indicative of curvature. Increasing λ reduces the curvature

$\phi(x) = [x, x^2 \ldots]$

$w_{ridge} = (\phi^T \phi + \lambda \Sigma)^{-1} \phi^T y$

Let $\lambda_1 \ldots \lambda_n$ be eigenvalues of $\phi^T\phi$

Then $(\phi^T\phi + \lambda I)$ has eigenvalues

$\lambda_1+\lambda, \lambda_2+\lambda, \ldots \lambda_n+\lambda$

If $\min(\lambda_i)$ was close to 0, matrix $\phi^T\phi$ can be tricky to invert (low condition #)

$\Rightarrow (\phi^T\phi + \lambda I)$ → more robustly non-singular (improved condition number)

Do Closed-form solutions Always Exist?

- Linear regression and Ridge regression both have closed-form solutions
  - For linear regression,

$$w^* = (\phi^\top \phi)^{-1} \phi^\top y$$

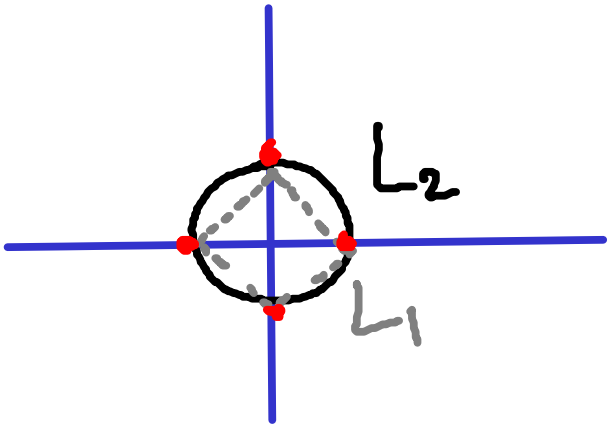
  - For ridge regression,

$$w^* = (\phi^\top \phi + \lambda I)^{-1} \phi^\top y$$

    (for linear regression, $\lambda = 0$)
- What about optimizing the formulations (constrained/penalized) of Lasso (L_1 norm)? And support-based penalty (L_0 norm)?: Also requires tools of Optimization/duality

Why is Lasso Interesting?

$L_0 \Rightarrow$ # non-zero components



Level curves of L_1, L_2

Level curves for L_1 tend to have more corners (•) than L_2 and hence, tend to yield sparser solutions

ie more w_i's = 0

Support Vector Regression

One more formulation before we look at Tools of Optimization/duality