Introduction to Machine Learning - CS725

Instructor: Prof. Ganesh Ramakrishnan

Lecture 6 - Support Vector Regression and Optimization Basics

From Bayesian Estimates to (Pure) Bayesian Prediction

	Point?	$p(x\|D)$
MLE	$\hat{\theta}_{MLE} = \text{argmax}_\theta \, LL(D\|\theta)$	$p(x\|\theta_{MLE})$
Bayes Estimator	$\hat{\theta}_B = E_{p(\theta\|D)} E[\theta]$	$p(x\|\theta_B)$
MAP	$\hat{\theta}_{MAP} = \text{argmax}_\theta \, p(\theta\|D)$	$p(x\|\theta_{MAP})$
Pure Bayesian		$p(\theta\|D) = \frac{p(D\|\theta)p(\theta)}{\int_m p(D\|\theta)p(\theta)d\theta}$ $p(D\|\theta) = \prod_{i=1} p(x_i\|\theta)$ $p(x\|D) = \int_\theta p(x\|\theta)p(\theta\|D)d\theta$

where θ is the parameter

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

$$\begin{aligned}
\Pr(y \mid \mathbf{x}, \mathcal{D}) &= \Pr(y \mid \mathbf{x}, < \mathbf{x}_1, y_1 > \ldots < \mathbf{x}_m, y_m >) \\
&= \int_{\mathbf{w}} \Pr(y|\mathbf{w}; \mathbf{x}) \Pr(\mathbf{w} \mid \mathcal{D}) d\mathbf{w} \\
&\sim \mathcal{N}\left(\mu_m^T \phi(\mathbf{x}), \sigma^2 + \phi^T(\mathbf{x}) \Sigma_m \phi(\mathbf{x})\right)
\end{aligned}$$

where

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

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

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

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

MAP (and Bayes) Inference

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\text{argmax}} \; \Pr(\mathbf{w} \mid \mathcal{D}) = \underset{\mathbf{w}}{\text{argmax}} \; \log \Pr(\mathbf{w} \mid \mathcal{D}), \text{ where,}$$

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

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

..... (expanding & canceling out redundant terms & completing squares: Tutorial 3)

MAP (and Bayes) Inference

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}} \ \Pr(\mathbf{w} \mid \mathcal{D}) = \underset{\mathbf{w}}{\operatorname{argmax}} \ \log \Pr(\mathbf{w} \mid \mathcal{D}), \text{ where,}$$

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

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

..... (expanding & canceling out redundant terms & completing squares: Tutorial 3)

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

is the same as that of *Regularized Regression*.

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

Penalized Regularized Least Squares Regression

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

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

- **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. Some norms are mathematically easier to handle

Constrained Regularized Least Squares Regression

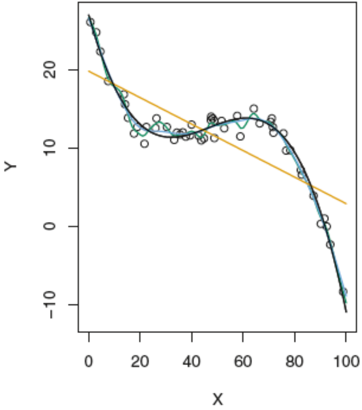
- **Intuition:** To discourage redundancy and/or stop coefficients of $\mathbf{w}$ from becoming too large in magnitude, constrain the error minimizing estimate using a penalty
- The general **Constrained Regularized L.S. Problem**:

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

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

- 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**
- **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

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

Lasso Regularized Least Squares Regression

- 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**

- *Lasso* Regression

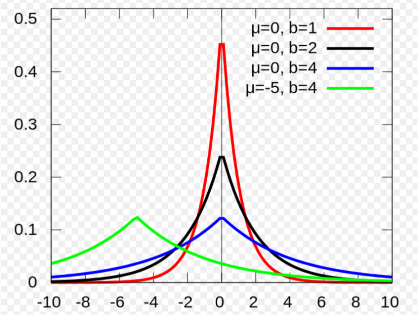
$$\mathbf{w}_{lasso} = \arg\min_{\mathbf{w}} \ ||\Phi\mathbf{w} - \mathbf{y}||_2^2 + \lambda||\mathbf{w}||_1^2$$

- Lasso is the MAP estimate of Linear Regression subject to Laplace Prior on $\mathbf{w} \sim Laplace(0, \theta)$

$$Laplace(w_i \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right)$$

Gaussian Hare vs. Laplacian Tortoise

- Gaussian easier to estimate



- Laplacian yields more sparsity

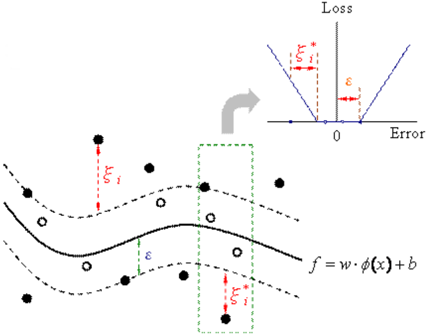
Support Vector Regression

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

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, **Support Vector Regression**
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

Support Vector Regression (SVR)



- Any point in the band (of ϵ) is not penalized. Thus the loss function is known as ϵ-*insensitive loss*
- Any point outside the band is penalized, and has slackness ξ_i or ξ_i^*
- The SVR model curve may not pass through any training point

- The tolerance ϵ is fixed
- It is desirable that $\forall i$:

- The tolerance ϵ is fixed
- It is desirable that $\forall i$:
    - $y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) - b \leq \epsilon + \xi_i$
    - $b + \mathbf{w}^\top \phi(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i^*$

SVR objective

- 1-norm Error, and L_2 regularized:

SVR objective

- 1-norm Error, and L_2 regularized:
  - $\min_{\mathbf{w},b,\xi_i,\xi_i^*} \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\sum_i(\xi_i + \xi_i^*)$
    s.t. $\forall i$,
    $y_i - \mathbf{w}^\top\phi(\mathbf{x}_i) - b \leq \epsilon + \xi_i$,
    $b + \mathbf{w}^\top\phi(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i^*$,
    $\xi_i, \xi_i^* \geq 0$
- 2-norm Error, and L_2 regularized:

SVR objective

- 1-norm Error, and L_2 regularized:
  - $\min_{\mathbf{w}, b, \xi_i, \xi_i^*} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i (\xi_i + \xi_i^*)$
    s.t. $\forall i$,
    $y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) - b \leq \epsilon + \xi_i$,
    $b + \mathbf{w}^\top \phi(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i^*$,
    $\xi_i, \xi_i^* \geq 0$
- 2-norm Error, and L_2 regularized:
  - $\min_{\mathbf{w}, b, \xi_i, \xi_i^*} \frac{1}{2} \|w\|^2 + C \sum_i (\xi_i^2 + \xi_i^{*2})$
    s.t. $\forall i$,
    $y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) - b \leq \epsilon + \xi_i$,
    $b + \mathbf{w}^\top \phi(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i^*$
  - Here, the constraints $\xi_i, \xi_i^* \geq 0$ are not necessary

Need for Optimization so far

- **Unconstrained (Penalized) Optimization:**

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

- **Constrained Optimization 1:**

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

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

- **Constrained Optimization 2 ($t = 1$ or 2):**

$$\arg\min_{\mathbf{w},b,\xi_i,\xi_i^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i (\xi_i^t + \xi_i^{*t})$$

s.t. $\forall i,\; y_i - w^\top\phi(\mathbf{x}_i) - b \leq \epsilon + \xi_i;\; b + \mathbf{w}^\top\phi(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i^*$

- **Equivalence**: λ (Penalized) $\equiv \theta$ (Constrained)
- **Duality**: Dual of Support Vector Regression

Solving Unconstrained Minimization Problem

- Intuitively: Minimize by setting derivative (gradient) to 0 and hoping to find **closed form** solution.
- When is such a solution a global minimum?
- For most optimization problems, finding closed form solutions is difficult. Even for linear regression (for which closed form solution exists), are there alternative methods?
  - Eg: Consider, $\mathbf{y} = \Phi\mathbf{w}$,where Φ is a matrix with full column rank, the least squares solution, $\mathbf{w}^* = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{y}$. Now, imagine that Φ is a very large matrix. with say, 100,000 columns and 1,000,000 rows. Computation of closed form solution might be challenging.
- How about iterative methods?