Assignment 2

Devansh Jain, 190100044

26 Sept 2021

Contents

1 Perceptron

1.1 CS 337: Theory

1.

Both 1-vs-1 and 1-vs-rest approach use binary classification algorithm.
1-vs-rest has K classifiers - one for each class.
1-vs-1 has K(K-1)/2 classifiers - one for each pair of classes.
In 1-vs-rest, each classifier predicts score (probability) and the class with highest score is chosen.
In 1-vs-1, each classifier predicts one class and the class which has been predicted the most is chosen.

Advantages and Disadvantages
Single classifier in 1-vs-1 uses subset of data, so each classifier is faster for 1-vs-1.
1-vs-rest trains less number of classifiers and hence is faster overall.
1-vs-1 is less prone to imbalance in dataset due to dominance in specific classes.

2.

$$\begin{aligned}
score(f, y) &= \sum_i f_i w_{y_i} \\
&= f^T w_y \\
score(f, y)_{new} &= f^T w_{y_{new}} \\
&= f^T w_{y_{old}} + f^T f \\
&= f^T w_{y_{old}} + ||f||_2^2 \\
&\geq f^T w_{y_{old}} \\
&\geq score(f, y)_{old} \\
score(f, y')_{new} &= f^T w_{y'_{new}} \\
&= f^T w_{y'_{old}} - f^T f \\
&= f^T w_{y'_{old}} - ||f||_2^2 \\
&\leq f^T w_{y'_{old}} \\
&\leq score(f, y')_{old}
\end{aligned}$$

Hence, proved.

3.

$$\mathcal{L}(f, y) = max(0, -yf^T w)$$

$$\nabla_w \mathcal{L}(f, y) = \begin{cases} 0 & \text{if } yf^T w \geq 0 \\ -yf & \text{if } yf^T w < 0, \text{ i.e. point is misclassified} \end{cases}$$

Gradient descent update:

$$w^{(k+1)} = w^{(k)} - \eta \nabla_w \mathcal{L}(f, y)$$

$$w^{(k+1)} = \begin{cases} w^{(k)} & \text{if } yf^T w \geq 0 \\ w^{(k)} + \eta y f & \text{if } yf^T w < 0, \text{ i.e. point is misclassified} \end{cases}$$

We can see that the gradient descent update rule is similar to perceptron update rule.
Using the fact that the perceptron update rule converges for linearly separable dataset, we can conclude that gradient descent algorithm also converges for linearly separable dataset.
Hence, proved.

P.S. The loss function takes into account only a single data point. To be precise, we are doing stochastic gradient descent here.

4.

In proof of the convergence theorem, if we use $\eta = 0.5$.
We get $\sqrt{k}r \geq ||w^{(k)}||_2 \geq u^T w^{(k)} \geq k\eta\gamma$.
Thus, $k \leq \frac{r^2}{\eta^2\gamma^2}$.
Upper bound on number of iterations under the modified algorithm is 4M.

<u>Alt</u>:
If we do double update at every step, i.e. take the same point again for every misclassification. It would be equivalent to original perceptron, so number of iterations would be 2M.
This is possible only if we chose the points in this fashion.

5.

$k \leq \frac{r^2}{\gamma^2}$, where $r \geq ||f||_2 \ \forall f \in \mathcal{D}$ and $\exists u, ||u||_2 = 1, \gamma \leq |u^T f| \ \forall f \in \mathcal{D}$.
Here, $f = [f_1 \ f_2 \ 1]$.
So, $r = \sqrt{3}$ and if we take $u = \frac{[1 \ 1 \ -1.5]}{\sqrt{4.25}}$, then $\gamma = \frac{1}{\sqrt{17}}$.
Thus, $k \leq 51$.

2 LASSO and ISTA

2.1 CS 337: Theory

1.

Given $y_i = x_i.w + \epsilon_i$, where $\epsilon_i \sim^{iid} \mathcal{N}(0, \sigma^2)$ and $w_j \sim^{iid} Lasso(0, \theta)$.

$$\begin{aligned}
\mathbb{P}(\mathcal{D}|w) &\propto \exp\left(- \frac{\sum_{i=1}^{n} ||y_i - x_i.w||_2^2}{2\sigma^2} \right) \\
\mathbb{P}(w) &\propto \exp\left(- \frac{\sum_{j=1} |w_j|}{\theta} \right) \\
&\propto \exp\left(- \frac{||w||_1}{\theta} \right) \\
\mathbb{P}(w|\mathcal{D}) &\propto \exp\left(- \frac{\sum_{i=1}^{n} ||y_i - x_i.w||_2^2}{2\sigma^2} - \frac{||w||_1}{\theta} \right)
\end{aligned}$$

$$\begin{aligned}
w_{\text{MAP}} &= \arg\max_{w} \ \mathbb{P}(w|\mathcal{D}) \\
&= \arg\max_{w} \ \exp\left(- \frac{\sum_{i=1}^{n} ||y_i - x_i.w||_2^2}{2\sigma^2} - \frac{||w||_1}{\theta} \right) \\
&= \arg\min_{w} \ \left(\frac{\sum_{i=1}^{n} ||y_i - x_i.w||_2^2}{2\sigma^2} + \frac{||w||_1}{\theta} \right) && (-\log(x) \text{ is a non-increasing function}) \\
&= \arg\min_{w} \ \sum_{i=1}^{n} ||y_i - x_i.w||_2^2 + \lambda ||w||_1 && \left(\lambda := \frac{2\sigma^2}{\theta} \right) \\
&= \arg\min_{w} \ L(w)
\end{aligned}$$

Hence, proved.

2.

Referencing from the lecture slides and book by Tibshirani:
As shown in Figure 1, the contours of the error and constrained functions meet at axes for l_1 norm unlike l_2 norm.
This results in sparser weight vector as weights which are too closed to zero in OLS and Ridge are made zero here.
We can also see that l_2 norm penalizes larger values of weights much more than l_1 norm, so in l_2 norm, having a small non-zero value is more likely than l_1 norm.

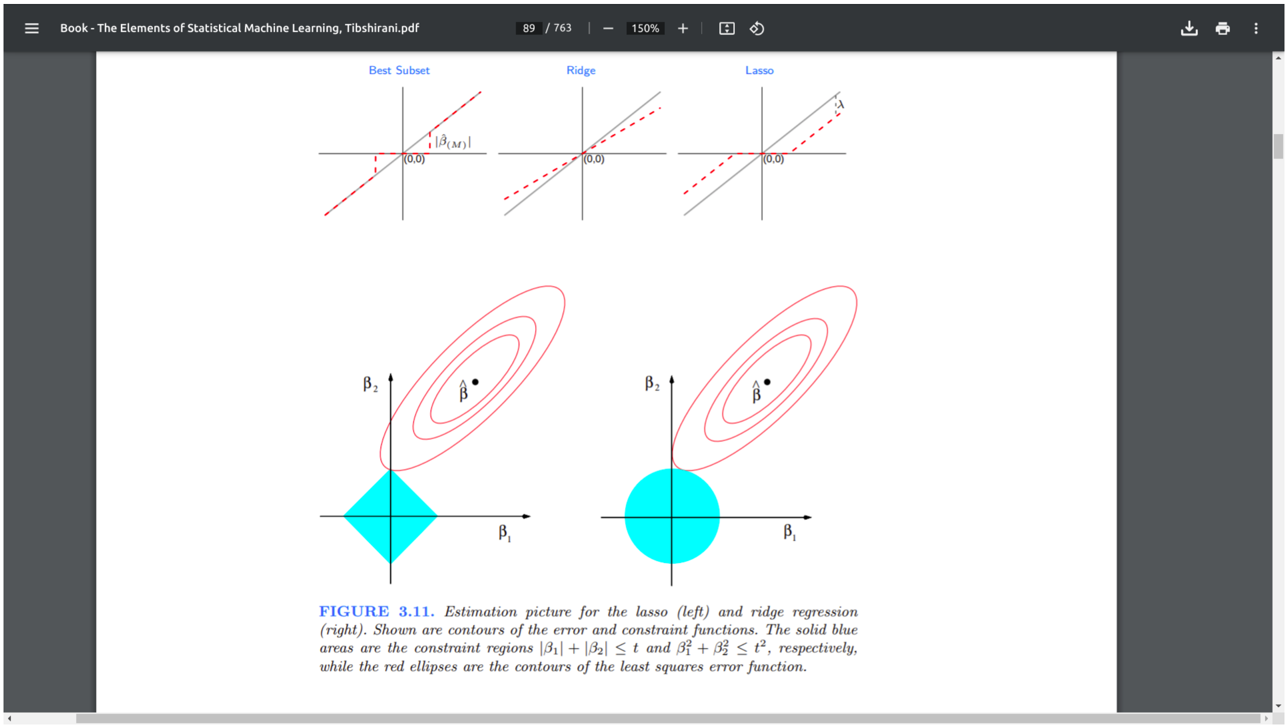
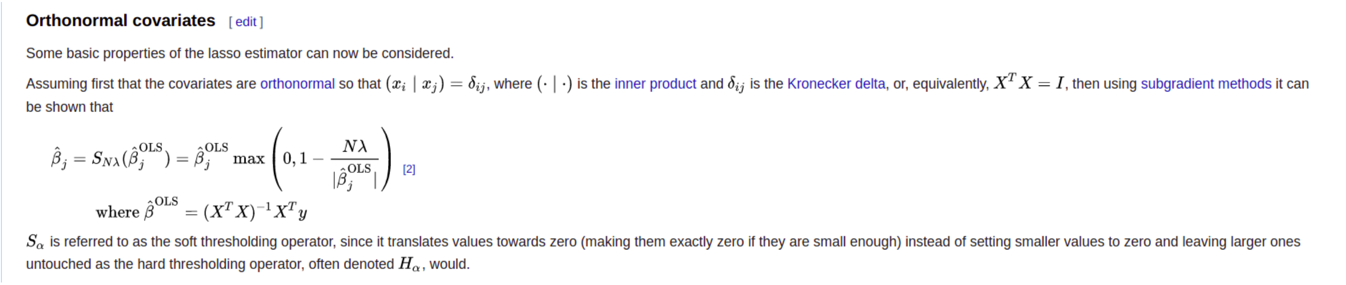


FIGURE 3.11. Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.

Figure 1: Tibshirani Book: Page 71

3.

Yes, a closed form for Lasso exists when the features are uncorrelated.
We can see this from OLS also, there was a unique solution only when the features were linearly independent.
Same way here, if the features are uncorrelated then there is a unique solution which can be found using subgradient methods (Referred to books and search).



Orthonormal covariates [edit]

Some basic properties of the lasso estimator can now be considered.

Assuming first that the covariates are orthonormal so that $(x_i \mid x_j) = \delta_{ij}$, where $(\cdot \mid \cdot)$ is the inner product and δ_{ij} is the Kronecker delta, or, equivalently, $X^T X = I$, then using subgradient methods it can be shown that

$$\hat{\beta}_j = S_{N\lambda}(\hat{\beta}_j^{\text{OLS}}) = \hat{\beta}_j^{\text{OLS}} \max\left(0, 1 - \frac{N\lambda}{|\hat{\beta}_j^{\text{OLS}}|}\right) \quad [2]$$

$$\text{where } \hat{\beta}^{\text{OLS}} = (X^T X)^{-1} X^T y$$

S_α is referred to as the soft thresholding operator, since it translates values towards zero (making them exactly zero if they are small enough) instead of setting smaller values to zero and leaving larger ones untouched as the hard thresholding operator, often denoted H_α, would.

Figure 2: Wikipedia: Lasso (statistics)

3 Bias Variance Trade-of

3.1 CS 337: Theory

1.

As discussed in class, Model complexity increases $\implies$ Bias decreases and Variance increases and, Model complexity decreases $\implies$ Bias increases and Variance decreases

(a) Increasing the value of λ in lasso regression
$\implies$ Model complexity decreases
$\implies$ Bias increases and Variance decreases

(b) Increasing model complexity by adding more features of high degree
$\implies$ Bias decreases and Variance increases

(c) Reducing dimension by choosing only those subset of features which are of more importance
$\implies$ Model complexity decreases
$\implies$ Bias increases and Variance decreases

2.

Let the test point be $(x, y = f(x) + \epsilon)$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$\begin{aligned}
\text{Mean Squared Error} &= \mathrm{E}[(\hat{f(x)} - y)^2] \\
&= \mathrm{E}[\hat{f(x)}^2 + y^2 - 2\hat{f(x)}y] \\
&= \mathrm{E}[(\hat{f(x)} - \overline{\hat{f(x)}})^2] + \overline{\hat{f(x)}}^2 + \mathrm{E}[(y - f(x))^2] + f(x)^2 - 2\overline{\hat{f(x)}}f(x) \\
&\left[\text{Using } \mathrm{E}[X^2] = \mathrm{E}[(X - \overline{X})^2] + \overline{X}^2;\ \mathrm{E}[\hat{f(x)}] = \overline{\hat{f(x)}};\ \mathrm{E}[y] = f(x)\right] \\
&= \mathrm{E}[(\hat{f(x)} - \overline{\hat{f(x)}})^2] + (\overline{\hat{f(x)}} - f(x))^2 + \sigma^2 \\
&= Variance(\hat{f(x)}) + Bias(\hat{f(x)})^2 + \text{irreducible noise variance}
\end{aligned}$$

Hence proved.