Unit 5 review – Iterative Methods

Gradient descent for solving general optimization problems: $\arg\min_w L(X, y; w)$

(Idea) Iterate in the direction of greatest descent $w^{(k+1)} = w^{(k)} - \tau \nabla_w L\left(X, y; w^{(k)}\right)$

Proximal gradient methods for solving regularized least-squares: $\arg\min_w \|Xw - y\|_2^2 + \gamma R(w)$

(Idea) Iterate between 2 steps:
$$\begin{cases} z^{(k)} = w^{(k)} - \tau X^T\left(Xw^{(k)} - y\right) & \text{Gradient descent on } \|Xw - y\|_2^2 \\ w^{(k+1)} = \arg\min_w \left\|w - z^{(k)}\right\|_2^2 + \gamma R(w) & \text{Regularize. Often closed–form solution exists.} \end{cases}$$

LASSO: $R(w) = \sum_i |w_i|$

- L1 regularization tends to encourage sparse solutions
- No closed form solution
- Proximal gradient method to solve LASSO iteratively

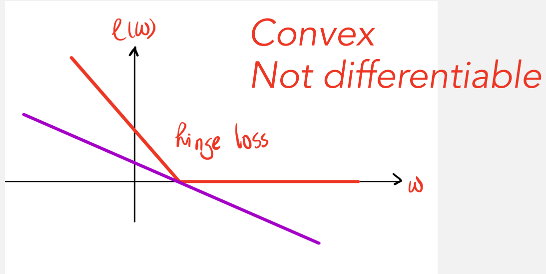
Support Vector Machines $\arg\min_w \sum_i (1 - y_i w^T x_i)_+ + \gamma \|w\|_2^2$

- Why hinge loss? Least squares penalizes easy samples.
- Minimizing $\|w\|_2^2$ subject to $y_i w^T x_i \geq 1$ maximizes the margin of the classifier.

(Today) **Sub gradients** and their role with nondifferentiable function optimization

Activity 20

Sub-gradients



Sub-gradient: *any plane that lies below function.*

any $\boldsymbol{v}$ such that $\ell(\boldsymbol{w}) \geq \ell(\boldsymbol{w}_0) + (\boldsymbol{w} - \boldsymbol{w}_0)^T \boldsymbol{v}$

Classifying new data:

$(\boldsymbol{x}_i, y_i), i = 1, \ldots,$ a million

$\boldsymbol{x} =$ 

$\widehat{y} = \text{sign}(\boldsymbol{x}^T \boldsymbol{w})$

if $\widehat{y} = 1$ then dog

if $\widehat{y} = -1$ then cat

Training a classifier:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{\text{a million}} (\boldsymbol{x}_i^T \boldsymbol{w} - y_i)^2$$

Problem: computing the loss is too slow.

Stochastic Gradient Descent

$$\min_{\boldsymbol{w}} \sum_{i=1}^{\text{a million}} \ell_i(\boldsymbol{w})$$

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \tau \nabla \ell(\boldsymbol{w}^k)$$

(Gradient Descent)

Main idea

Do gradient descent, but on a random subset of training examples at each iteration.

$$\boldsymbol{w}^{(1)} = \boldsymbol{w}^{(0)} - \tau \sum_{i=1}^{100} \nabla \ell_i(\boldsymbol{w}^{(0)})$$

$$\boldsymbol{w}^{(2)} = \boldsymbol{w}^{(1)} - \tau \sum_{i=101}^{200} \nabla \ell_i(\boldsymbol{w}^{(1)})$$

- Image/video classification and recognition
- ML translation
- Large scale prediction and regression tasks

1. An exponential loss function $f(w)$ is defined as

$$f(w) = \begin{cases} e^{-2(w-1)}, & w < 1 \\ e^{w-1}, & w \geq 1 \end{cases}$$

a) Is $f(w)$ convex? Why? *Hint:* Graph the function.

b) Is $f(w)$ differentiable everywhere? If not, where not?

c) The "differential set" $\partial f(\boldsymbol{w})$ is the set of subgradients $\boldsymbol{v} \in \partial f(\boldsymbol{w})$ for which $f(\boldsymbol{u}) \geq f(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^T \boldsymbol{v}$. Find the differential set for $f(w)$ as a function of w.

2. We are trying to predict whether a certain chemical reaction will take place as a function of our experimental conditions: temperature, pressure, concentration of catalyst, and several other factors. For each experiment $i = 1, \dots, m$ we record the experimental conditions in the vector $\boldsymbol{x}_i \in \mathbb{R}^n$ and the outcome in the scalar $b_i \in \{-1, 1\}$ ($+1$ if the reaction occurred and -1 if it did not). We will train our linear classifier to minimize hinge loss. Namely, we solve:

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \sum_{i=1}^{m} (1 - b_i \boldsymbol{x}_i^T \boldsymbol{w})_+ \qquad \text{where } (u)_+ = \max(0, u) \text{ is the hinge loss operator}$$

a) Derive a gradient descent method for solving this problem. Explicitly give the computations required at each step. *Note:* you may ignore points where the function is non-differentiable.

b) Explain what happens to the algorithm if you land at a $\boldsymbol{w}^k$ that classifies all the points perfectly, and by a substantial margin.

3. You have four training samples $y_1 = 1, \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $y_2 = 2, \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$, $y_3 = -1, \boldsymbol{x}_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, and $y_4 = -2, \boldsymbol{x}_4 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$. Use cyclic stochastic gradient descent to find the first two updates for the LASSO problem

$$\min_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 + 2||\boldsymbol{w}||_1$$

assuming a step size of $\tau = 1$ and $\boldsymbol{w}^{(0)} = 0$. Also indicate the data used for the first six updates.