Machine Learning (CSE 446): Probabilistic Approaches

Sham M Kakade

University of Washington
cse446-staff@cs.washington.edu

Announcements

- Midterm on Monday
- You may use a single side of a single sheet of handwritten notes that you prepared.

Probabilistic machine learning:

Probabilistic machine learning:

- **define a probabilistic model** relating random variables x to y
- **estimate its parameters**.

Maximum Likelihood Estimation and the Log loss

The principle of maximum likelihood estimation is to choose our parameters to make our observed data as likely as possible (under our model).

- Mathematically: find $\hat{\mathbf{w}}$ that maximizes the probability of the labels $y_1, \dots y_N$ given the inputs $x_1, \dots x_N$.
- The Maximum Likelihood Estimator (the **'MLE'**) is:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{N} p_{\mathbf{w}}(y_i \mid \mathbf{x}_i)$$
$$= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} -\log p_{\mathbf{w}}(y_i \mid \mathbf{x}_i)$$

Linear Regression-MLE is same as Squared Loss Minimization!

- Linear regression defines $p_{\mathbf{w}}(Y \mid X)$ as follows:

$$p_{\mathbf{w}}(Y \mid \mathbf{x}) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{(Y - \mathbf{w} \cdot \mathbf{x})^2}{2\sigma^2}$$

this is a modeling assumption.

- the MLE is then:

$$\operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{N} -\log p_{\mathbf{w}}(y_i \mid \mathbf{x}_i) \equiv \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \underbrace{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}_{SquaredLoss_i(\mathbf{w}, b)}$$

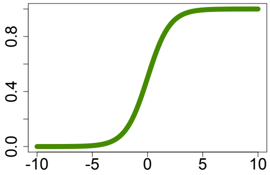
A Probabilistic Model for Binary Classification: Logistic Regression

- For $Y \in \{-1, 1\}$ define $p_{\mathbf{w},b}(Y \mid X)$ as:
  1. Transform feature vector $\mathbf{x}$ via the "activation" function:

$$a = \mathbf{w} \cdot \mathbf{x} + b$$

  2. Transform a into a binomial probability by passing it through the logistic function:

$$p_{\mathbf{w},b}(Y = +1 \mid \mathbf{x}) = \frac{1}{1 + \exp -a} = \frac{1}{1 + \exp -(\mathbf{w} \cdot \mathbf{x} + b)}$$



- If we learn $p_{\mathbf{w},b}(Y \mid \mathbf{x})$, we can (almost) do whatever we like!

The MLE for Logistic Regression

- the MLE for the logistic regression model:

$$\operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{N} -\log p_{\mathbf{w}}(y_i \mid \mathbf{x}_i) = \operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{N} \log\left(1 + \exp(-y_i \mathbf{w} \cdot \mathbf{x}_i)\right)$$

  for this expression we need $y \in \{-1, 1\}$

- This is the logistic loss function that we saw earlier.
- How do we compute the MLE?

Derivation for Log loss for Logistic Regression

- for $Y = 1$,

$$p_{\mathbf{w}}(Y = +1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w} \cdot \mathbf{x})}$$

- For $Y = -1$,

$$p_{\mathbf{w}}(Y = -1 \mid \mathbf{x}) = 1 - p_{\mathbf{w}}(Y = 1 \mid \mathbf{x}) = \frac{\exp(-\mathbf{w} \cdot \mathbf{x})}{1 + \exp(-\mathbf{w} \cdot \mathbf{x})} = \frac{1}{1 + \exp(\mathbf{w} \cdot \mathbf{x})}$$

- So we can write for both $y \in \{-1, 1\}$,

$$p_{\mathbf{w}}(Y = y \mid \mathbf{x}) = \frac{1}{1 + \exp(-y\mathbf{w} \cdot \mathbf{x})}$$

  (as this agrees with both cases above).
- Finally, we just plug in this expression and we obtain: the MLE for the logistic regression model:

$$\operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{N} -\log p_{\mathbf{w}}(y_i \mid \mathbf{x}_i) = \operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{N} \log\left(1 + \exp(-y_i \mathbf{w} \cdot \mathbf{x}_i)\right)$$

Loss Minimization & Gradient Descent

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \underbrace{\ell(\mathbf{x}_i, y_i, \mathbf{w})}_{\ell_i(\mathbf{w})} + R(\mathbf{w})$$

What is GD here?

What do we do if N is large?

Stochastic Gradient Descent (SGD): by example

$$\underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2$$

- Gradient descent:

- Note we are computing an average. What is a crude way to estimate an average?
- Stochastic gradient descent:

Will it converge? **If the step size in SGD is a constant, we will not converge.**

Stochastic Gradient Descent (SGD) (without regularization)

Data: loss functions $\ell(\cdot)$, training data, number of iterations K, step sizes $\langle \eta^{(1)}, \ldots, \eta^{(K)} \rangle$
Result: parameters $\mathbf{w} \in \mathbb{R}^d$
initialize: $\mathbf{w}^{(0)} = \mathbf{0}$;
for $k \in \{1, \ldots, K\}$ **do**
    $i \sim \text{Uniform}(\{1, \ldots, N\})$;
    $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \eta^{(k)} \cdot \nabla_{\mathbf{w}} \ell_i(\mathbf{w}^{(k-1)})$;
end
return $\mathbf{w}^{(K)}$;

Algorithm 1: SGD

Stochastic Gradient Descent: Convergence

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \ell_i(\mathbf{w})$$

- $\mathbf{w}^{(k)}$: our parameter after k updates.
- Thm: Suppose $\ell(\cdot)$ is convex (and satisfies mild regularity conditions). There exists a way to decrease our step sizes $\eta^{(k)}$ over time so that our function value, $F(\mathbf{w}^{(k)})$ will converge to the minimal function value $F(\mathbf{w}^*)$.
- This Thm is different from GD in that **we need to turn down our step sizes over time!**