Machine Learning (CSE 446): Probabilistic View of Logistic Regression and Linear Regression

Sham M Kakade

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

Announcements

- Midterm: Weds, Feb 7th. Policies:
  - You may use a single side of a single sheet of handwritten notes that you prepared.
  - You must turn your sheet of notes in, with your name on it, in at the conclusion of the exam, even if you never looked at it.
  - You may not use electronics devices of any sort.
- Today:
  Review: Regularization and Optimization
  New: (wrap up GD) + probabilistic modeling!

Review

Regularization / Ridge Regression

- **Regularize** the optimization problem:

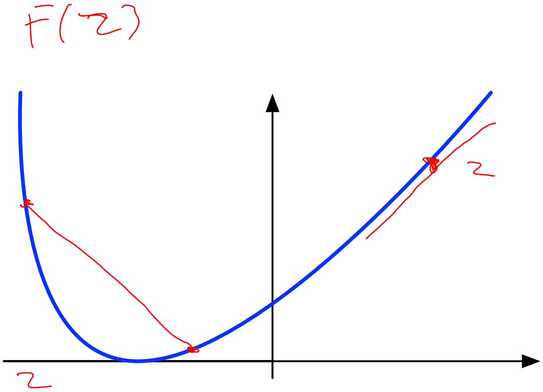
$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 + \lambda \|\mathbf{w}\|^2 = \min_{\mathbf{w}} \frac{1}{N} \|Y - X^\top \mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2$$

- This particular case: “Ridge” Regression, Tikhonov regularization
- The solution is the **least squares estimator**:

$$\mathbf{w}^{\text{least squares}} = \underbrace{\left(\frac{1}{N} X^\top X + \lambda \mathbb{I} \right)}_{d \times d}^{-1} \left(\frac{1}{N} X^\top Y \right)$$

Regularization is often necessary for the “exact” solution method (regardless of if d bigger/less than N)

Gradient Descent



- Want to solve:

$$\min_{z} F(z)$$

- How should we update z?

$z \leftarrow z - \eta \nabla F(z)$

stepsize

Gradient Descent

Data: function $F : \mathbb{R}^d \to \mathbb{R}$, number of iterations K, step sizes $\langle \eta^{(1)}, \ldots, \eta^{(K)} \rangle$
Result: $\mathbf{z} \in \mathbb{R}^d$
initialize: $\mathbf{z}^{(0)} = \mathbf{0}$;
for $k \in \{1, \ldots, K\}$ **do**   step size
  $\mathbf{z}^{(k)} = \mathbf{z}^{(k-1)} - \eta^{(k)} \cdot \nabla_{\mathbf{z}} F(\mathbf{z}^{(k-1)})$;
end
return $\mathbf{z}^{(K)}$;

Algorithm 1: GradientDescent

Today

Gradient Descent: Convergence

curvature condition

- Denote:
  $\mathbf{z}^* = \mathrm{argmin}_{\mathbf{z}} F(\mathbf{z})$: the global minimum
  $\mathbf{z}^{(k)}$: our parameter after k updates.
- Thm: Suppose F is convex and "L-smooth". Using a **fixed step size** $\eta \leq \frac{1}{L}$, we have:

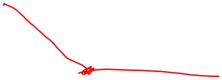
$$F(\mathbf{z}^{(k)}) - F(\mathbf{z}^*) \leq \frac{\|\mathbf{z}^{(0)} - \mathbf{z}^*\|^2}{\eta \cdot k}$$

  That is the **convergence rate** is $O(\frac{1}{k})$.
- **This Thm applies to both the square loss and logistic loss!**

cvx + smooth

Proof intuition: smoothness and GD Convergence

cvx



~~smooth~~

- L-Smooth functions: "The gradients don't change quickly." Precisely, For all z, z'

$$\|\nabla F(z) - \nabla F(z')\| \leq L\|z - z'\|$$

- Proof idea:
  1. If our gradient is large, we will make good progress decreasing our function value:

  2. If our gradient is small, we must have value near the optimal value:

A better idea?

- Remember the Bayes optimal classifier. $\mathcal{D}(x, y)$ is the true probability of (x, y).

$$f^{(\mathsf{BO})}(x) = \operatorname*{argmax}_y \mathcal{D}(x, y) = \operatorname*{argmax}_y D(y|x)D(x)$$
$$= \operatorname*{argmax}_y \mathcal{D}(y \mid x)$$

- Of course, we don't have $\mathcal{D}(y \mid x)$.

Probabilistic machine learning: **define a probabilistic model** relating random variables x to y and **estimate its parameters**.

A Probabilistic Model for Binary Classification: Logistic Regression

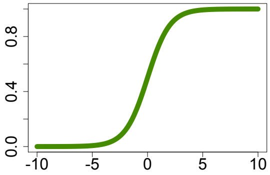
- For $Y \in \{-1, 1\}$ define $p_{\mathbf{w},b}(Y \mid X)$ as:
  1. Transform feature vector $\mathbf{x}$ via the "activation" function:

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

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

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

$$\frac{1}{1+e^{-a}}$$



a

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

Maximum Likelihood Estimation

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

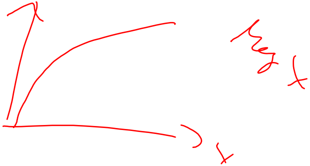
- Mathematically: find $\hat{\mathbf{w}}$ that maximizes the probability of the labels $y_1, \ldots y_n$ given the inputs $x_1, \ldots x_n$.
- Note, by the i.i.d. assumption:

$$\mathcal{D}(y_1, \ldots y_n \mid \mathbf{x}_1, \ldots \mathbf{x}_N) = P(y_1|x_1) D(y_2|x_2) \ldots D(y_n|x_n)$$

- The Maximum Likelihood Estimator (the **'MLE'**) is:

$$\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} \prod_{n=1}^{N} p_{\mathbf{w}}(y_n \mid \mathbf{x}_n)$$

Maximum Likelihood Estimation and the Log loss



- The 'MLE' is:

$$
\begin{aligned}
\hat{\mathbf{w}} &= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{n=1}^{N} p_{\mathbf{w}}(y_n \mid \mathbf{x}_n) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \log \prod_{n=1}^{N} p_{\mathbf{w}}(y_n \mid \mathbf{x}_n) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{n=1}^{N} \log p_{\mathbf{w}}(y_n \mid \mathbf{x}_n) \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{n=1}^{N} -\log p_{\mathbf{w}}(y_n \mid \mathbf{x}_n)
\end{aligned}
$$

- This is referred to as the **log loss**.

The MLE for Logistic Regression

$$p_w(y=1 \mid x) = \frac{1}{1+e^{-(w\cdot x)}}$$

CVX.

- the MLE for the logistic regression model:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{n=1}^{N} -\log p_{\mathbf{w}}(y_n \mid \mathbf{x}_n) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{n=1}^{N} \log\left(1+\exp(-y_n \mathbf{w}\cdot\mathbf{x}_n)\right)$$

∇

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

Do G.D.

$$\sum_{n=1}^{N} \frac{e^{-y_n(w\cdot x_n)}}{1+e^{-y_n(w\cdot x_n)}}\left(-y_n \vec{x}_n\right)$$

Derivation for Log loss for Logistic Regression: scratch space

Linear Regression as a Probabilistic Model

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

1. Observe the feature vector $\mathbf{x}$; transform it via the activation function:

$$\mu = \mathbf{w} \cdot \mathbf{x}$$

2. Let μ be the mean of a normal distribution and define the density:

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

3. Sample Y from $p_{\mathbf{w}}(Y \mid \mathbf{x})$.

Linear Regression-MLE is (Unregularized) Squared Loss Minimization!

$$\operatorname*{argmin}_{\mathbf{w}} \sum_{n=1}^{N} -\log p_{\mathbf{w}}(y_n \mid \mathbf{x}_n) \equiv \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \underbrace{(y_n - \mathbf{w} \cdot \mathbf{x}_n)^2}_{SquaredLoss_n(\mathbf{w},b)}$$

Where did the variance go?