Machine Learning (CSE 446):
Gradient Descent and Stochastic Gradient Descent

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.
- A few comments on the course difficulty
- Today:
  New: GD and SGD

Course difficulty

Why is it difficult/what should we learn?

- homeworks
- exams
- grading

Review

Gradient Descent: Convergence

- Denote:
  $\mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}} F(\mathbf{z})$: the global minimum
  $\mathbf{z}^{(k)}$: our parameter after k updates.
- Thm: Suppose F is convex and "L-smooth" (e.g. works for square loss and the logistic loss). 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 $O(\frac{1}{k})$.
- **A constant learning rate means no parameter tuning!**

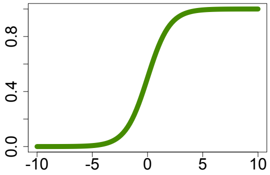
Probabilistic machine learning:

Probabilistic machine learning:

- **define a probabilistic model** relating random variables x to y
- **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:
  $$p_{\mathbf{w},b}(Y = +1 \mid \mathbf{x}) = \frac{1}{1 + \exp -a} = \frac{1}{1 + \exp -(\mathbf{w} \cdot \mathbf{x} + b)}$$



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

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 $y_1, \ldots y_n$ given the inputs $x_1, \ldots x_n$.
- The Maximum Likelihood Estimator (the **'MLE'**) is:

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

The MLE for Logistic Regression

- the MLE for the logistic regression model:

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

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

Derivation for Log loss for Logistic Regression: scratch space

Today

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!

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

Where did the variance go?

What is GD here?

Loss Minimization & Gradient Descent

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} \underbrace{\ell(\mathbf{x}_n, y_n, \mathbf{w})}_{\ell_n(\mathbf{w})} + R(\mathbf{w})$$

What is GD here?

What do we do if N is large?

Stochastic Gradient Descent (SGD): by example

$$\underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2$$

- Gradient descent:

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

Will it converge?

Stochastic Gradient Descent (SGD): by example

$$\underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2$$

- 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

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

- $\mathbf{w}^{(k)}$: 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!**