Recitation 3

Gradient Descent and Logistic Regression

10-701: Introduction to Machine Learning

Feb 6, 2026

1 Gradient Descent

Gradient descent (GD) is one of the most commonly used optimization algorithms in machine learning. Here we will go over 1) why we are using gradients in the first place and 2) why stochastic gradient descent (SGD) works.

1.1 Gradient Points in the Direction of Steepest Ascent

In class, we saw a pictorial sketch of why gradient descent makes sense; we will discuss it more formally here.

One way of thinking about this is that for a differentiable multivariate function $f : \mathbb{R}^d \to \mathbb{R}$, the gradient at point $\boldsymbol{x} \in \mathbb{R}^d$ (i.e. $\nabla f(\boldsymbol{x})$) points in the direction of *steepest ascent* at $\boldsymbol{x}$.

Recall from calculus that we define the directional derivative with respect to some unit vector $\boldsymbol{u} \in \mathbb{R}^d$ as

$$D_{\boldsymbol{u}} f(\boldsymbol{x}) = \lim_{h \to 0} \frac{f(\boldsymbol{x} + h\boldsymbol{u}) - f(\boldsymbol{x})}{h}$$

In words, the directional derivative describes how the function value instantaneously changes if we step along the direction of $\boldsymbol{u}$ from point $\boldsymbol{x}$. With some calculation, one can show that

$$\begin{aligned} D_{\boldsymbol{u}} f(\boldsymbol{x}) &= \boldsymbol{u}^T \nabla f(\boldsymbol{x}) \\ &= ||\boldsymbol{u}|| ||\nabla f(\boldsymbol{x})|| \cos\theta \ (\because \text{inner product}) \end{aligned}$$

where θ is the angle between $\boldsymbol{u}$ and $\nabla f(\boldsymbol{x})$. We know $||\boldsymbol{u}|| = 1$, and it is easy to see that when $\theta = 0$, $D_{\boldsymbol{u}} f(\boldsymbol{x})$ is maximized.

Thus, at each point, we should step in the direction of the gradient to maximally *increase* the function value (gradient ascent). Meanwhile, we should step in the direction opposite of the gradient to maximally *decrease* the function value (gradient descent). Whether we want to use gradient ascent or descent will depend on whether we want to maximize or minimize some objective function.

1.2 Stochastic Gradient Descent

In lecture, you are introduced with the concept of gradient descent (GD). However, the *stochastic* gradient descent (SGD) is more common in practice than the vanilla GD, as the

gradient updates in SGD are computationally cheaper when working with large datasets but still lead to good, generalizable solutions (often even better). An in-depth discussion of SGD is beyond the scope of this course, but here we will see why it makes sense at a high level.

Suppose we have some ML model (e.g., linear regression, logistic regression, neural network) and we want to optimize the parameters $\boldsymbol{w}$ of that model using data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$. And let's say that our loss function is

$$\mathcal{L}_{\text{total}} = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

which is an average of the loss $\mathcal{L}$ for each data point $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ across our dataset.

Now, we want to use the following SGD algorithm to iteratively update $\boldsymbol{w}_t$:

1. Randomly sample (without replacement) m indices from $\{1, \ldots, n\}$. Call the set of sampled indices $\mathcal{B}$.
2. Calculate the loss using the sampled data points
   $\tilde{\mathcal{L}} = \frac{1}{m} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i) \underbrace{1[i \in \mathcal{B}]}_{\text{indicator}}$
3. Update $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \frac{\partial \tilde{\mathcal{L}}}{\partial \boldsymbol{w}}$

Based on this setup, answer the question below.

1. In statistics, the bias of an estimator (or bias function) is the difference between this estimator's expected value and the true value of the parameter being estimated.

   $$Bias(\hat{\theta}, \theta) = \mathbb{E}_{x|\theta}[\hat{\theta} - \theta]$$

   An estimator or decision rule with zero bias is called unbiased. Show that the stochastic gradient is an unbiased estimator of the gradient, i.e. show that:

   $$\mathbb{E}\left[\frac{\partial \tilde{\mathcal{L}}}{\partial \boldsymbol{w}}\right] = \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}$$

   (Hint: $\mathbb{E}\big[1[i \in \mathcal{B}]\big] = p(i \in \mathcal{B})$. Also check out the remark below if you need better

intuition.)

$$\begin{aligned}\mathbb{E}\left[\frac{\partial \tilde{\mathcal{L}}}{\partial \boldsymbol{w}}\right] &= \frac{1}{m}\sum_{i=1}^{n}\frac{\partial \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{w}}\mathbb{E}\big[1[i \in \mathcal{B}]\big] \\ &= \frac{1}{m}\sum_{i=1}^{n}\frac{\partial \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{w}}\frac{m}{n} \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{w}} \\ &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}\end{aligned}$$

The fact that the stochastic gradient is an unbiased estimator of the full-batch gradient is an important justification for using SGD. In fact, showing unbiasedness is generally important in many stochastic algorithms.

Remark: Randomly choosing the data points for updating the parameters at each iteration is what makes SGD *stochastic*. Try comparing $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}$ and $\frac{\partial \tilde{\mathcal{L}}}{\partial \boldsymbol{w}}$; you should notice that since we are only using a random subset of all of our training points to calculate the gradient, $\frac{\partial \tilde{\mathcal{L}}}{\partial \boldsymbol{w}}$ can be thought of as an approximation of $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}$.

2 Logistic Regression

1. For a probability value $p \in (0, 1)$, what is the range of the odds, $\frac{p}{1-p}$, and the log-odds, $\log\left(\frac{p}{1-p}\right)$? Explain why this makes the log-odds a desirable transformation of our data to fit with our affine model.

   The range of the odds is $(0, +\infty)$. The range of the log-odds is $(-\infty, +\infty)$. Since the range of an affine function is $(-\infty, +\infty)$, the log-odds is a reasonable choice to fit with a simple affine model. If we fit the probabilities or odds directly with a linear model, the range of the linear model would be a superset of the range of the data, requiring a constraint on the regression coefficients to ensure meaningful outputs for all data points. Fitting the log-odds avoids such issues.

   (as a reminder)

$$w^T x + b = \log\left(\frac{p}{1-p}\right)$$

$$e^{w^T x+b} = \frac{p}{1-p}$$

$$p = \frac{e^{w^T x+b}}{1+e^{w^T x+b}}$$

$$p = \frac{1}{1+e^{-(w^T x+b)}}$$

2. We consider the following models of logistic regression for a binary classification with a sigmoid function $g(z) = \frac{1}{1+e^{-z}}$:

   - Model 1: $P(Y = 1 \mid X, w_1, w_2) = g(w_1 X_1 + w_2 X_2)$
   - Model 2: $P(Y = 1 \mid X, w_0, w_1, w_2) = g(w_0 + w_1 X_1 + w_2 X_2)$

   We have three training examples:

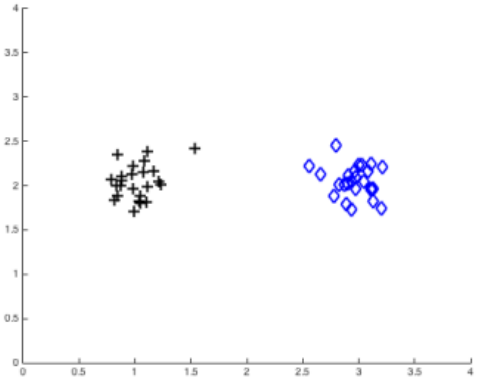
$$\begin{array}{lll} x^{(1)} = [1, 1]^T & x^{(2)} = [1, 0]^T & x^{(3)} = [0, 0]^T \\ y^{(1)} = 1 & y^{(2)} = 0 & y^{(3)} = 1 \end{array}$$

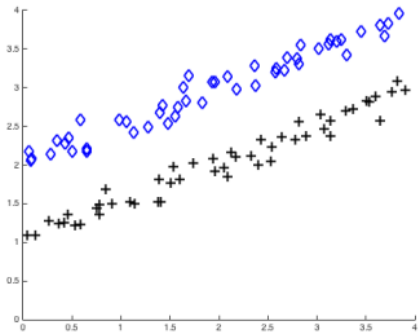
   Does it matter how the third example is labeled in Model 1? i.e., would the learned value of $w = (w1, w2)$ be different if we change the label of the third example to 0? Does it matter in Model 2? Briefly explain your answer.

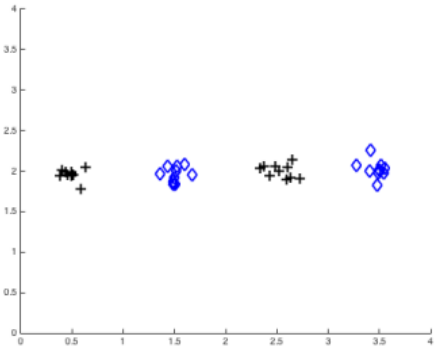
   It does not matter in Model 1 because $x^{(3)} = (0, 0)$ makes $w_1 x_1 + w_2 x_2$ always zero, and hence the likelihood of the model (i.e., $p = 0.5$) does not depend on the value of w.

   It does matter in Model 2, influencing the bias term w_0 depending on the label.

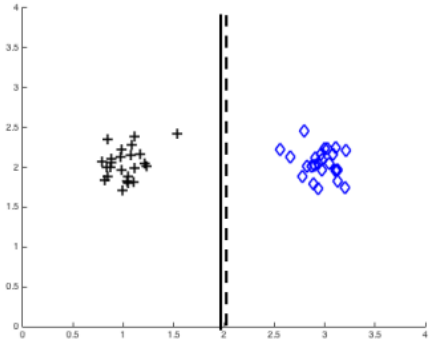
3. For each of the following figures, evaluate the performance of a logistic regression model. To be more precise, if logistic regression is able to classify the given points, indicate the appropriate decision boundary. If not, provide a reasonable justification as to why the model is unable to classify the particular set of points.



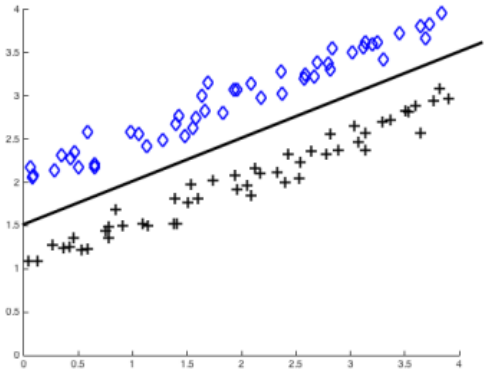




In figure 1, Logistic Regression can separate the data.



In figure 2, Logistic Regression can separate the data.



Logistic Regression cannot separate the points in figure 3, since it can only solve for a linear decision boundary.