1 Multinomial MLE

Example 1.1: Suppose we have multiple classes $a_1, \ldots, a_n$ and m data samples D. We observe k_1 data for the first class, k_2 data for the second class and so on, $k_i > 0$. We seek to compute $P(\theta|D)$ where $\theta = (p_1, \ldots, p_n)$ with $\sum p_i = 1$. Please write the expression for MLE, i.e. $P(D|\theta)$.

Example 1.2: What is the MLE estimator for θ?

2 Bernoulli MAP

Now let's consider MAP instead. Recall that we're trying to maximize the posterior probability $P(\theta|D)$.

Example 2.1: Assume that the probability of the prior $P(\theta = 0.25) = 0.1$, $P(\theta = 0.55) = 0.6$, $P(\theta = 0.75) = 0.2$. Given the 2 samples with the observations ["head", "head"], which one of the three θ values is the most likely estimation for θ?

Example 2.2: Now we have 10 observations with 8 heads and 2 tails, which one of the three θ values gives you the best estimation for θ?

Example 2.3: How will the number of samples affects the estimation of θ?

3 Gradient Descent

Let f be a differentiable function. A gradient descent step update the value of x_t:

$$x_t \leftarrow x_{t-1} - \eta \cdot \nabla f(x_{t-1})$$

Visualization: https://suniljangirblog.wordpress.com/2018/12/03/the-outline-of-gradient-descent/

Why do we need gradient descent?

Example 3.1: Compute gradient descent for function $f(x) = \frac{1}{2}x^2$ with $x_0 = 1$ and learning rate $\eta = 0.1$. Please repeat for two steps.

4 Other Gradient Descent Methods

4.1 Stochastic Gradient Descent

Problem: Compute the gradient $\nabla f(x)$ for the full batch of data is expensive.

Solution: Randomly sample a data from the distribution D and compute the gradient.

- Fast
- Less accurate at each step
- But it will converge towards the optimal point in expectation

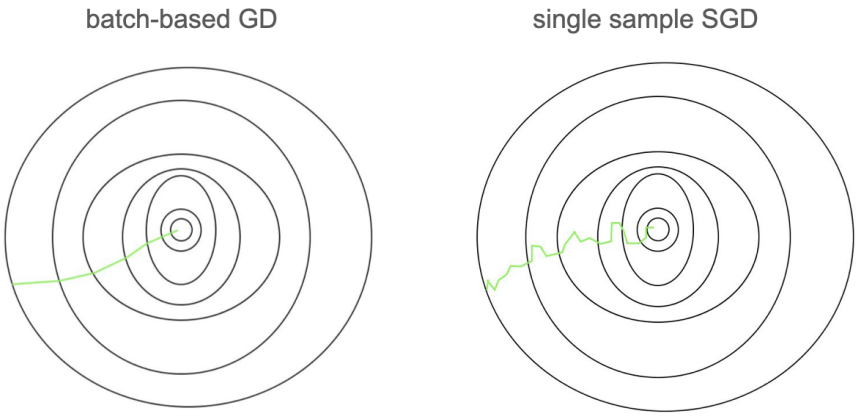

Figure 1: Cite: https://www.cs.cmu.edu/~aarti/Class/10315_Fall20/recs/rec2slides.pdf

4.2 Newton's Method

Idea: Quadratic approximation of $f(x)$

Let $f(x)$ be convex and twice differentiable. The quadratic approximation of $f(x)$ is

$$f(y) = f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}(y-x)^T\nabla^2 f(x)(y-x)$$

Minimize over y yields

$$x_t \leftarrow x_{t-1} - (\nabla^2 f(x))^{-1}\nabla f(x)$$

- Less optimization steps
- Each step is expensive to compute

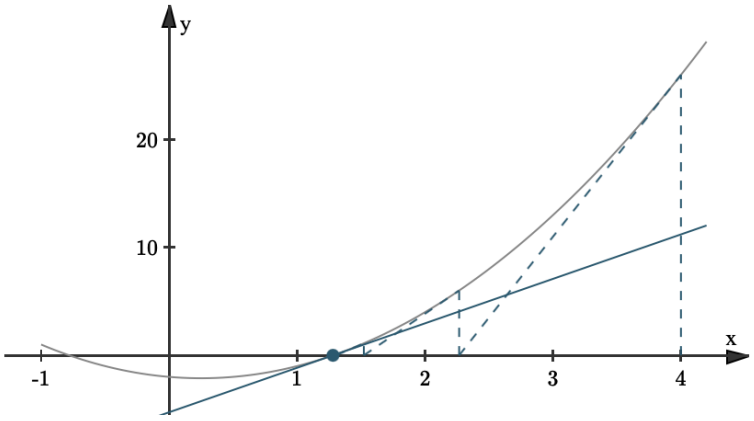

Figure 2: Cite: https://www.intmath.com/applications-differentiation/newtons-method-interactive.php

4.3 SGD with Momentum

Update x_t using gradients from previous steps

$$x_t \leftarrow x_{t-1} - v_t$$

where

$$v_t = \gamma \cdot v_{t-1} + \eta \cdot \nabla f(x_{t-1})$$

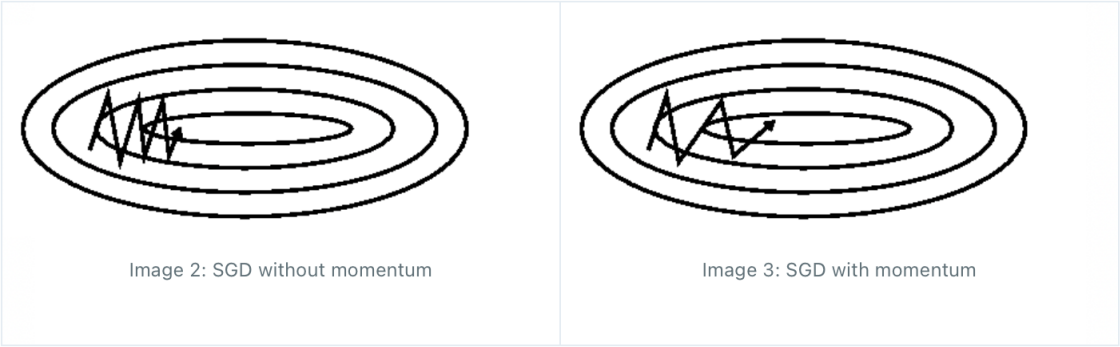

Image 2: SGD without momentum

Image 3: SGD with momentum

Figure 3: Cite: https://ruder.io/optimizing-gradient-descent/

4.4 Other Optimization Methods

- RMSProp
- AdaGrad
- AdaDelta
- Adam
- And many many more ...