CS 229: Assignment 4

Santiago de Buen

November 21, 2020

1 Principal Component Analysis

$f_u(x)$ is the projection of point x onto the direction given by u. We begin by examining this equation

$$f_u(x) = \arg\min_{v \in \mathcal{V}} ||x - v||_2^2$$

We can replace v with αu, and we optimize the expression for a specific $x^{(i)}$. Note that we can now optimize over α given that u is taken as a given in this expression.

$$f_u(x^{(i)}) = \arg\min_{\alpha} ||x^{(i)} - \alpha u||_2^2$$

From the lecture, we know that the α that solves this equation is

$$\alpha = \langle u, x^{(i)} \rangle$$

We want to obtain v:

$$v = \alpha u = \langle u, x^{(i)} \rangle u \tag{1}$$

Now we can examine the mean squared error between the projection and original points, and we will show that the unit-length vector u that minimizes MSE corresponds to the first principal component.

$$\arg\min_u \sum_{i=1}^{n} ||x^{(i)} - f_u(x^{(i)})||_2^2$$

$$\arg\min_u \sum_{i=1}^{n} ||x^{(i)} - u^T x^{(i)} u||_2^2$$

$$\arg\min_u \sum_{i=1}^{n} \left[(x^{(i)} - (u^T x^{(i)})u)^T (x^{(i)} - (u^T x^{(i)})u) \right]$$

$$\arg\min_u \sum_{i=1}^{n} \left[x^{(i)^T} x^{(i)} - 2(u^T x^{(i)})^2 + (u^T x^{(i)})^2 u^T u \right]$$

$$\arg\min_u \sum_{i=1}^{n} \left[-(u^T x^{(i)})^2 \right]$$

Because the first term does not depend on u and $u^T u = 1$

$$\arg\max_u \sum_{i=1}^{n} (u^T x^{(i)})^2$$

$$\arg\max_u u^T \left(\sum_{i=1}^{n} x^{(i)} {x^{(i)}}^T \right) u$$

Realize that $\sum_{i=1}^{n} x^{(i)} {x^{(i)}}^T$ is the covariance of our data, which means that $x^T x$ it is a symmetric and square matrix. Therefore, we can represent it with the following decomposition, where Λ contains the eigenvalues of the matrix

$$\arg\max_u u^T U \Lambda U^T u$$

For any $u_k \in u$, where k represents the number of principal components, the term $u_k^T U$ is 0 for every term except for column k, where it has a value of 1. Analogously, $U^T u_k$ is 0 for every term except for row k. Therefore, the u that maximizes this expression is the one that selects the largest element of Λ.

Λ contains the eigenvalues, so maximizing this expression is equivalent to finding the largest principal component for the data. Because we order the principal components from largest to smallest, we are finding the first principal component.

2 Independent Component Analysis

(a) **Gaussian source**

In this problem, we assume that sources come from a standard normal distribution: $s_j \sim \mathcal{N}(0, 1)$. We can express the log-likelihood function as

$$\ell(W) = \sum_{i=1}^{n} \left(\log |W| + \sum_{j=1}^{d} \log g'(w_j^T x^{(i)}) \right) \tag{2}$$

where $g'(w_j^T x^{(i)})$ is the standard normal distribution. Our goal is to deduce the simplest expression for the relationship between W and X. Expressing the normal distribution,

$$g'(s_j^{(i)}) = g'(w_j^T x^{(i)}) = \frac{1}{(2\pi)^{d/2} I} \exp\left(-\frac{1}{2} (w_j^T x^{(i)})^T I (w_j^T x^{(i)}) \right)$$

We want to maximize the log-likelihood function:

$$\max_W \ell(W) = \max_W \sum_{i=1}^{n} \left[\log |W| + \sum_{j=1}^{d} \left(\log \frac{1}{(2\pi)^{d/2} I} - \frac{1}{2} (w_j^T x^{(i)})^T I (w_j^T x^{(i)}) \right) \right]$$

$$= \max_W \; n \log |W| - \frac{1}{2} \sum_{i=1}^{n} {x^{(i)}}^T W W^T x^{(i)}$$

Using the property that $\nabla_W |W| = |W| {(W^{-1})}^T$ and setting equal to zero

$$n {(W^{-1})}^T - \left(\sum_{i=1}^{n} {x^{(i)}}^T x^{(i)} \right) W = 0$$

$$n(W^{-1})^T - XX^TW = 0$$

$$n(W^{-1})^T = XX^TW$$

$$n(WW^T)^{-1} = XX^T$$

$$WW^T = n(XX^T)^{-1} \tag{3}$$

This is the simplest form of the relationship between W and X, and it is not a closed form solution. In particular, if we choose an arbitrary orthogonal matrix R, and the data is generated according to ARs instead of As, we would obtain the same expression. Therefore, we cannot identify a unique W matrix that unmixes the data.

(b) **Laplace source**

Our objective is to find the update rule for W when sources are distributed according to a standard Laplace distribution, $s_i \sim \mathcal{L}(0,1)$, where $f_{\mathcal{L}}(s) = \frac{1}{2}\exp(-|s|)$.

We begin by substituting the definition of the Laplace distribution onto (2).

$$\ell(W) = \sum_{i=1}^{n}\left(\log|W| + \sum_{j=1}^{d}\log\left(\frac{1}{2}\exp(-|w_j^T x^{(i)}|)\right)\right)$$

$$= \sum_{i=1}^{n}\left(\log|W| - \sum_{j=1}^{d}|w_j^T x^{(i)}|\right)$$

Our objective is to maximize with respect to W

$$\max_W \ell(W) = \max_W \sum_{i=1}^{n}\left(\log|W| - \sum_{j=1}^{d}|w_j^T x^{(i)}|\right)$$

Note that $\nabla_W|W| = |W|(W^{-1})^T$ and $\nabla_W|W| = \text{sign}(W)$

$$\max_W \ell(W) = \sum_{i=1}^{n}\left((W^T)^{-1} - \begin{bmatrix}\text{sign}(W_1^T x^{(i)})x^{(i)} \\ \text{sign}(W_2^T x^{(i)})x^{(i)} \\ \vdots \\ \text{sign}(W_d^T x^{(i)})x^{(i)}\end{bmatrix}\right)$$

Therefore, the update rule for the entire training set is

$$W := W + \alpha \sum_{i=1}^{n} \left(\left(W^T\right)^{-1} - \begin{bmatrix} \text{sign}(W_1^T x^{(i)}) x^{(i)} \\ \text{sign}(W_2^T x^{(i)}) x^{(i)} \\ \vdots \\ \text{sign}(W_d^T x^{(i)}) x^{(i)} \end{bmatrix} \right) \tag{4}$$

And the update rule for a single example i can be expressed as

$$W := W + \alpha \left(\left(W^T\right)^{-1} - \begin{bmatrix} \text{sign}(W_1^T x^{(i)}) x^{(i)} \\ \text{sign}(W_2^T x^{(i)}) x^{(i)} \\ \vdots \\ \text{sign}(W_d^T x^{(i)}) x^{(i)} \end{bmatrix} \right) \tag{5}$$

(c) **Cocktail Party Problem**

The full unmixing matrix is

$$W = \begin{bmatrix} 52.83512054 & 16.79594361 & 19.94115104 & -10.1985153 & -20.89752402 \\ -9.92869812 & -0.97683438 & -4.67732108 & 8.04342263 & 1.78792785 \\ 8.31150961 & -7.47676167 & 19.3154481 & 15.17437116 & -14.3261898 \\ -14.66745004 & -26.64459003 & 2.44058135 & 21.38205443 & -8.42074783 \\ -0.26912908 & 18.37411935 & 9.31234188 & 9.1025964 & 30.59424603 \end{bmatrix}$$

3 Markov Decision Process

(a) **Proof that Bellman update operator is contraction in the max-norm**

Our objective is to prove the following

$$||B(V_1) - B(V_2)||_\infty \leq \gamma ||V_1 - V_2||_\infty \tag{6}$$

Before we begin, let's prove that $|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|$. For any functions $f(x)$ and $g(x)$, the following must be true

$$\begin{aligned} |f(x)| &\leq |f(x) - g(x)| + |g(x)| \\ \max |f(x)| &\leq \max(|f(x) - g(x)| + |g(x)|) \leq \max |f(x) - g(x)| + \max |g(x)| \\ \max |f(x)| - \max |g(x)| &\leq \max |f(x) - g(x)| \end{aligned} \tag{7}$$

Going back to the original problem, We begin by expressing the infinity norm as the maximum of the absolute value, and by replacing $B(V)$. We focus on the left-hand-side term.

$$||B(V_1) - B(V_2)||_\infty = \max_{s \in S} \left| R(s) + \gamma \max_{a \in A} \sum_{s' \in S} P_{sa}(s')V_1(s') - R(s) + \gamma \max_{a \in A} \sum_{s' \in S} P_{sa}(s')V_2(s') \right|$$

$$= \gamma \max_{s \in S} \left| \max_{a \in A} \sum_{s' \in S} P_{sa}(s')V_1(s') - \max_{a \in A} \sum_{s' \in S} P_{sa}(s')V_2(s') \right|$$

$$\leq \gamma \max_{s \in S} \max_{a \in A} \left| \sum_{s' \in S} P_{sa}(s')V_1(s') - \sum_{s' \in S} P_{sa}(s')V_2(s') \right| \tag{8}$$

where we used the property derived in (7).

$$= \gamma \max_{s \in S} \max_{a \in A} \left| \sum_{s' \in S} P_{sa}(s')[V_1(s') - V_2(s')] \right|$$

$$\leq \gamma \max_{s \in S} \max_{a \in A} \sum_{s' \in S} P_{sa}(s')|V_1(s') - V_2(s')|$$

where we are using the property $|\sum x| \leq \sum |x|$

$$= \gamma \max_{s \in S} |V_1(s') - V_2(s')|$$

$$= \gamma ||V_1 - V_2||_\infty \tag{9}$$

where we use the definition of P_{sa} as a probability distribution, $\sum_{s' \in S} P_{sa}(s') = 1$

Therefore, we have proven (6).

(b) **Proof that V is a fixed point of B**

We proof this point by contradiction. Assume that there are two fixed points such that $B(V_1) = V_1$ and $B(V_2) = V_2$. Using, the results from (a):

$$||B(V_1) - B(V_2)||_\infty \leq \gamma ||V_1 - V_2||_\infty$$

$$||V_1 - V_2||_\infty \leq \gamma ||V_1 - V_2||_\infty$$

$$1 \leq \gamma$$

However, $\gamma < 1$ by definition. Therefore, there can be no more than a single fixed point of B.

4 Reinforcement Learning: Policy Gradient

(a) **Policy Gradient**

Our goal is to derive the expression for $\nabla_\theta \ln \pi_\theta(0|s)$ and $\nabla_\theta \ln \pi_\theta(1|s)$.

We have a *logistic* policy, which can be expressed as $\pi_\theta(0|s) = \sigma(\theta^T s)$ and $\pi_\theta(1|s) = 1 - \sigma(\theta^T s)$.

First, lets obtain the gradient of the logistic function:

$$\nabla\sigma(z) = (-1) \cdot (1 + e^{-z})^{-2} \cdot e^{-z} \cdot (-1)$$

$$= \frac{e^{-z}}{(1 + e^{-z})^2}$$

$$= \frac{e^{-z}}{1 + e^{-z}} \cdot \frac{1}{1 + e^{-z}}$$

$$= (1 - \sigma(z)) \cdot \sigma(z)$$

Therefore, we can express the gradient of the policy function as

$$\nabla_\theta \ln \pi_\theta(0|s) = \frac{1}{\sigma(\theta^T s)}(1 - \sigma(\theta^T s)) \cdot \sigma(\theta^T s)s$$

$$= (1 - \sigma(\theta^T s))s \tag{10}$$

$$\nabla_\theta \ln \pi_\theta(1|s) = \frac{1}{\sigma(-\theta^T s)}(\sigma(-\theta^T s) - 1) \cdot \sigma(-\theta^T s)s$$

$$= ((1 - \sigma(\theta^T s)) - 1)s$$

$$= -\sigma(\theta^T s)s \tag{11}$$

(b) **Implementation**

The plot generated by `full_trajectory` is shown below.

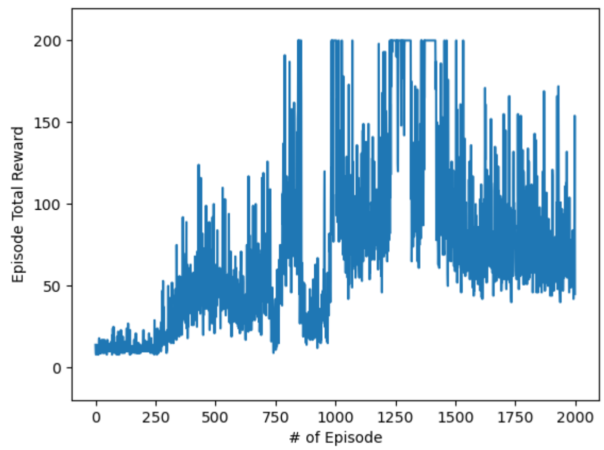

Figure 1: Reward under Full Trajectory Gradient

(c) **Reward-To-Go**

The plot generated by `reward_to_go` is shown below.

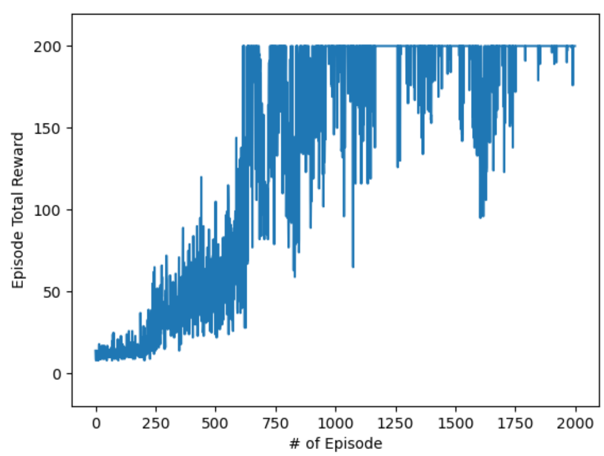

Figure 2: Reward under Reward To Go Gradient

The implementation using full trajectory is much more volatile than the reward-to-go implementation. Not only is the reward-to-go algorithm more stable, it converges more quickly as well. Therefore, the reward-to-go algorithm is preferable. Theoretically, we are taking advantage of the fact that the expectation of the gradient of the policy function is zero for fixed t in the past, removing a term that is only adding noise to the training process.