Homework 6
Stochastic Gradient Descent on Support Vector Machines

CS-449 System for data science

March 25, 2019

This homework will provide a short tutorial for deriving stochastic gradient descent (SGD) for a binary classification support vector machine (SVM), with batch-size equal to 1. It would be perfectly fine if you implemented your previous project I in another format, but make sure goes through this homework and fully understand what are the "formal" way of building the SGD. Note that the formulation of trainable weights, data are slightly different from the *Hogwild!* paper.

The original optimization problem for the SMV is given by

$$\min_{\mathbf{w}\in\mathcal{R}^D} \sum_{n=1}^{N} \ell(y_n \mathbf{x}_n^T \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{1}$$

where $\ell\, \mathcal{R} \to \mathcal{R}, \ell(z) = \max 0, 1 - z$ is the *hinge* loss. For any $n, 1 \leq n \leq N$, the vector of $\mathbf{x}_n \in \mathcal{R}^D$ is the n^{th} data example, and $y_n \in \pm 1$ is the corresponding label.

You are asked to implement a single thread stochastic gradient descent (SGD) for the SVM formulation 1. In every iteration, pick one data example $n \in [N]$ uniformly at random, and perform an update on the trainable weights $\mathbf{w}$ based on the (sub)gradient of the objective. Then iterate by picking the next n. You should implement the algorithm in Python, and we provide a template adapted from previous course of EPFL, pattern recognition and machine learning.

[Hints for this tutorial and the project milestone II: You should think the following questions while doing this exercise. How do you compute the gradient? How to determine the loss is fully converged? Can you pick more than one sample during each update? What is a good way to measure the model you trained so far?]