Gradient, Subgradient and how they may affect your grade(ient)

David Sontag & Yoni Halpern

February 7, 2016

1 Introduction

The goal of this note is to give background on convex optimization and the Pegasos algorithm by Shalev-Schwartz et al. The Pegasos algorithm is a *Stochastic sub-gradient descent method* for solving SVM problems which takes advantage of the *structure* and *convexity* of the SVM loss function. We describe each of these terms in the upcoming sections.

2 Convexity

A **set** $X \subseteq \mathbb{R}^d$ is a **convex** set if for any $\vec{x}, \vec{y} \in X$ and $0 \leq \alpha \leq 1$,

$$\alpha\vec{x} + (1-\alpha)\vec{y} \in X$$

Informally, if for any two points $\vec{x}$, $\vec{y}$ that are in the set every point on the line connecting $\vec{x}$ and $\vec{y}$ is also included in the set, then the set is convex. See Figure 1 for examples of non-convex and convex sets.

A **function** $f : \mathcal{X} \to \mathbb{R}$ is **convex** for a convex set $\mathcal{X}$ if $\forall \vec{x}, \vec{y} \in \mathcal{X}$ and $0 \leq \alpha \leq 1$,

$$f(\alpha\vec{x} + (1-\alpha)\vec{y}) \leq \alpha f(\vec{x}) + (1-\alpha) f(\vec{y}) \tag{1}$$

Informally, a function is convex if the line between any two points on the curve always upper bounds the function (see Figure 3). We call a function **strictly convex** if the inequality in Eq. 1 is a strict inequality. See See Figure 2 for examples of non-convex and convex functions.

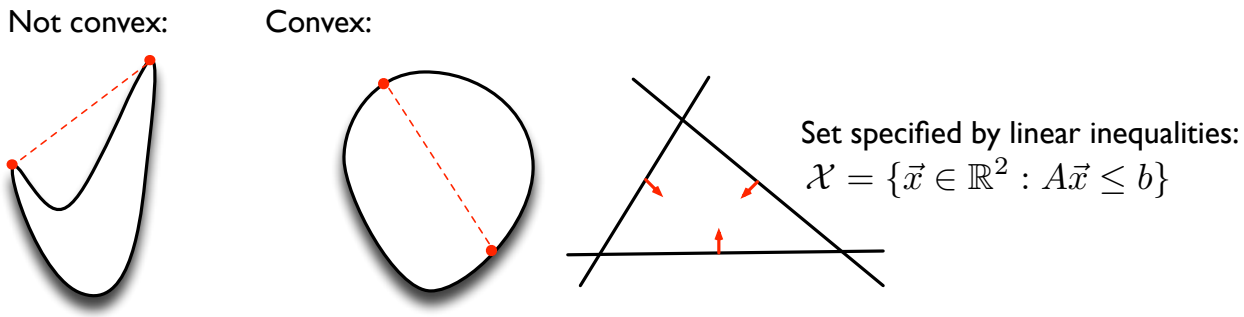

Figure 1: Illustration of a non-convex and two convex sets in $\mathbb{R}^2$.

A function $f(x)$ is **concave** is $-f(x)$ is convex. Importantly, it can be shown that strictly convex functions always have a unique minima.

For a function $f(x)$ defined over the real line, one can show that $f(x)$ is convex if and only if $\frac{d^2}{dx^2}f \geq 0 \ \forall x$. Just as before, strict convexity occurs when the inequality is strict. For example, consider $f(x) = x^2$. The first derivative of $f(x)$ is given by $\frac{d}{dx}f = 2x$ and its second derivative by $\frac{d^2}{dx^2}f = 2$. Since this is always strictly greater than 0, we have proven that $f(x) = x^2$ is strictly convex. As a second example, consider $f(x) = \log(x)$. The first derivative is $\frac{d}{dx}f = \frac{1}{x}$, and its second derivative is given by $\frac{d^2}{dx^2}f = -\frac{1}{x^2}$. Since this is negative for all $x > 0$, we have proven that $\log(x)$ is a *concave* function over $\mathbb{R}_+$.

This matters because optimization for convex functions is easy. In particular, one can show that nearly any reasonable optimization method, such as gradient descent (where one starts at arbitrary point, moves a little bit in the direction opposite to the gradient, and then repeats), is **guaranteed** to reach a global optimum of the function. Note that whereas the minimization of convex functions is easy, likewise, the maximization of concave functions is easy.

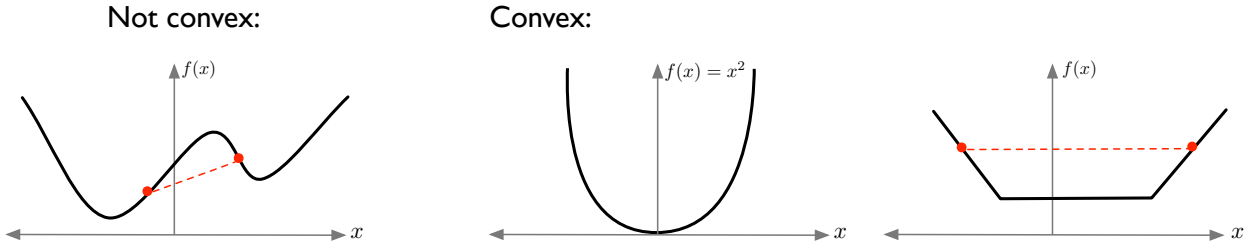

Figure 2: Illustration of a non-convex and two convex functions over $\mathcal{X} = \mathbb{R}$.

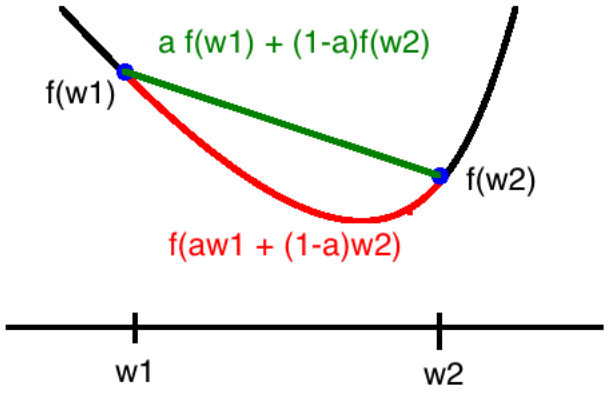

Figure 3: Convexity: Taking a weigted average of the values of the function (green) is greater than or equal to evaluating the function at a point which is the weighted average of the arguments (red).

2.1 Combining convex functions

Certain methods of combining convex functions preserves their convexity. These combination rules are useful to be able to look at a function composed of simple parts and determine quickly that it is also convex. For our purposes now, we note that a non-negative weighted sum of convex functions is convex:

$$g = \sum_i c_i f_i \text{ is convex if all } f_i \text{ are convex and } c_i \geq 0. \tag{2}$$

Additionally, a pointwise maximum of convex functions is convex.

$$h = \max\{f_i(w), f_j(w)\} \text{ is convex if all } f_i \text{ and } f_j \text{ are convex.} \tag{3}$$

For more detail on convexity preserving operations see Chapter 3 of Boyd and Vandenberghe's Convex Optimization (link).

2.2 The primal SVM objective is convex in w and b

Recall that primal SVM objective can be written as:

$$f(w,b) = \frac{1}{2}||w||^2 + \sum_i \max\{0, 1 - (y_i w \cdot x_i - b)\}.$$

Check for yourself that the above rules can be used to quickly prove that the above function is convex by breaking it into simpler parts with recognizable convexity.

2.3 Convex functions and greedy descent

Convex functions have an important property that allows us to search for a minimum using a *greedy* search method. If f is convex, with a global minimizer w^* (i.e. $f(w^*) \leq f(w)\ \forall w$) then from any point w_0, there is a connected path $w_0 \to w^*$ such that for every point w_i along the path, we have that $f(w_i) \leq f(w_0)$. This property allows us to use a greedy search method which always takes steps to reduce $f(w)$ without worrying that it will lead to a sub-optimal solution. This property follows from the definition of convexity (i.e. Equation 1). This brings us to our next section on greedy descent methods.

3 Greedy descent methods

Let's say we want to minimize some differentiable function $f(w)$. The following greedy descent algorithm is a template for many optimization techniques. We will give some more detail about each of these steps in the upcoming sections.

Algorithm 1 Greedy Descent

Input: a convex, differentiable function f.
Output: w_t, a minimizer of f.
initialize $w_0 = 0$, $t = 0$
while not converged **do**
  Choose a direction p_t
  Choose a stepsize α_t
  Update $w_{t+1} \leftarrow w_t + \alpha_t p_t$
  Test for convergence
  $t \leftarrow t + 1$
end while

3.1 Choosing a direction p

We can choose $p = -\nabla f$ (steepest descent). Later in this note we will see the stochastic gradient descent algorithm which uses an approximation to the gradient. Many other methods exist for choosing a direction, but we will not discuss them here.

3.2 Choosing a stepsize

Many options here:

- Constant stepsize: $\alpha_t = c$.
- Decaying stepsize: $\alpha_t = c/t$ (can also use different rates of decay (e.g. $\frac{1}{\sqrt{t}}$).
- Backtracking linesearch: Start with $\alpha_t = c$. Check for a decrease: Is $f(w_t + \alpha_t p)$ lower than $f(w_t)$? If the decrease condition is not met, multiply α_t by a decaying factor γ, (common choice is $\gamma = 0.5$) and repeat until the decrease condition is met. (Prove to yourself that this method will not terminate until it reaches a local minimum.)

The ipython notebook linked to on the course website gives interactive examples of the effects of choosing a stepsize rule.

3.3 Test for convergence

Again, many options exist here:

- Fixed number of iterations: Terminate if $t \geq T$.
- Small increase: Terminate if $f(w_{t+1}) - f(w_t) \leq \epsilon$.
- Small change: Terminate if $||w_{t+1} - w_t|| \leq \epsilon$.

4 Sub-gradient descent methods

Notice that the SVM objective is not continously differentiable. We cannot directly apply gradient descent but we can apply *subgradient* descent.

The subgradient of a convex function f at w_0 is formally defined as *all vectors* v such that for any other point w

$$f(w) - f(w_0) \geq v \cdot (w - w_0) \tag{4}$$

If f is differentiable at w_0, then the subgradient contains only one vector which is the gradient $\nabla f(w_0)$. However, when f is *not differentiable*, there may be many different values for v that satisfy this inequality (Figure 4).

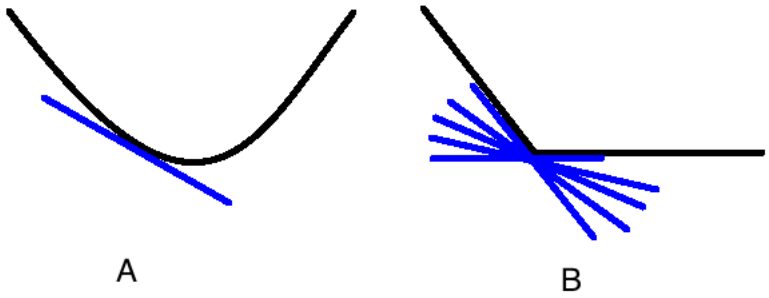

Figure 4: Subgradient: Function A is differentiable and only has a single subgradient at each point Function B is not differentiable at a single point. At that point it has *many* subgradients. (subtangent lines are shown in blue.)

5 Stochastic sub-gradient descent methods

Notice that the SVM objective contains an average over data points. We can approximate this average by looking at a single data point at a time. This serves as the basis for stochastic gradient descent methods. At each iteration we randomly choose a single data point to look at, uniformly from the set of all data points.

Instead of calculating the exact gradient by summing over all data points, we *approximate* it by looking at only a single data point. (Mini-batch methods exist which interpolate between the two extremes of stochastic gradient descent and traditional gradient descent. In a mini-batch method, the sum is approximated by looking at a small set of training points.)

6 Putting it all together

The Pegasos algorithm by Shalev-Schwartz et al. is a *stochastic subgradient descent method on the primal objective of the SVM function* (by now you should know what each of those terms means).

Recall once more the SVM objective with M data points:

$$f(w) = \frac{1}{2}||w||^2 + C\sum_{i=0}^{M-1} \max\{0, 1 - (y_i w \cdot x_i)\}.$$

Here we assume that $b = 0$, so we are solving the SVM with an *unbiased hyperplane*. The

following is a subgradient with respect to w (verify for yourself):

$$g(w) = w - C \sum_{i=1}^{M-1} \mathbf{1} \left[y_i w \cdot x_i < 1 \right] y_i x_i$$

A stochastic version of the algorithm uses the following *approximation* to the subgradient:

$$\tilde{g}(w) = w - CM\mathbf{1} \left[y_i w_t \cdot x_i < 1 \right] y_i x_i$$

where i is chosen uniformly at random from [0... M-1] at each step.

The subgradient gives a *direction* of movement, we also need to choose a stepsize. The Pegasos algorithm uses a *decaying* stepsize of $\alpha_t = \frac{CM}{t}$.

We can now put the full pesasos algorithm together:

Algorithm 2 Pegasos algorithm

Output: w_t, an approximate minimizer of f, the SVM primal objective function.
initialize $w_1 = 0$, $t = 1$
while not converged **do**
  Choose a direction: $p_t \leftarrow -\tilde{g}(w_t, b_t)$
  Choose a stepsize: $\alpha_t \leftarrow \frac{CM}{t}$
  Update $w_{t+1} \leftarrow w_t + \alpha_t p_t$
  $t \leftarrow t + 1$
  Test for convergence
end while