CPSC 340: Machine Learning and Data Mining

Gradient Descent

Bonus slides

Beyond Gradient Descent

- There are many variations on gradient descent.
  - Methods employing a “line search” to choose the step-size.
  - “Conjugate” gradient and “accelerated” gradient methods.
  - Newton’s method (which uses second derivatives).
  - Quasi-Newton and Hessian-free Newton methods.
  - Stochastic gradient (later in course).

- This course focuses on gradient descent and stochastic gradient:
  - They’re simple and give reasonable solutions to most ML problems.
  - But the above can be faster for some applications.

Constraints, Continuity, Smoothness

- Sometimes we need to optimize with constraints:
  - Later we'll see "non-negative least squares".

$$\min_{w \geq 0} \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

  - A vector 'w' satisfying w ≥ 0 (element-wise) is said to be "feasible".
- Two factors affecting difficulty are continuity and smoothness.
  - Continuous functions tend to be easier than discontinuous functions.
  - Smooth/differentiable functions tend to be easier than non-smooth.
  - See the calculus review [here](#) if you haven't heard these words in a while.

Convexity, min, and argmin

- If a function is convex, then all critical points are global optima.

- However, convex functions don't necessarily have critical points:
  - For example, $f(x) = a*x$, $f(x) = \exp(x)$, etc.

- Also, more than one 'x' can achieve the global optimum:
  - For example, $f(x) = c$ is minimized by any 'x'.

Why use the negative gradient direction?

- For a twice-differentiable 'f', multivariable Taylor expansion gives:

$$f(w^{t+1}) = f(w^t) + \nabla f(w^t)^T(w^{t+1} - w^t) + \frac{1}{2}(w^{t+1} - w^t)\nabla^2 f(v)(w^{t+1} - w^t)$$

for some 'v' between w^{t+1} and w^t.

- If gradient can't change arbitrarily quickly, Hessian is bounded and:

$$f(w^{t+1}) = f(w^t) + \nabla f(w^t)^T(w^{t+1} - w^t) + \underbrace{O(\|w^{t+1} - w^t\|^2)}_{\text{becomes negigible as } w^{t+1} \text{ gets close to } w^t}$$

  – But which choice of w^{t+1} decreases 'f' the most?
    - As $||w^{t+1}-w^t||$ gets close to zero, the value of w^{t+1} minimizing $f(w^{t+1})$ in this formula converges to $(w^{t+1} – w^t) = - \alpha^t \nabla f(w^t)$ for some scalar α^t.
    - So if we're moving a small amount, the optimal w^{t+1} is: $w^{t+1} = w^t - \alpha_t \nabla f(w^t)$ for some scalar α_t.

Normalized Steps

Question from class: "can we use $w^{t+1} = w^t - \frac{1}{\|\nabla f(w^t)\|} \nabla f(w^t)$"

This will work for a while, but notice that

$$\|w^{t+1} - w^t\| = \left\| \frac{1}{\|\nabla f(w^t)\|} \nabla f(w^t) \right\|$$

$$= \frac{1}{\|\nabla f(w^t)\|} \|\nabla f(w^t)\|$$

$$= 1$$

So the algorithm never converges

Optimizer "findMin" Details

question

stop following 99 views

The minimizer function

Hi all,

I'm just curious how the minimizers given to us works. Are there any resources can give us more details about it?

the instructors' answer, *where instructors collectively construct a single answer*

It's just a basic gradient descent implementation with some clever guesses for the step-size.

The step-size on each iteration is initialized using the method from this classic paper (which works surprisingly well but we don't really know why except in two dimensions):
http://pages.cs.wisc.edu/~swright/726/handouts/barzilai-borwein.pdf

That step-size is evaluated using a standard condition ("Armijo condition") and then it fits a polynomial regression model based on the function and directional derivative values and tries the step-size minimizing this polynomial. Both these tricks are described in Nocedal and Wright's "Numerical Optimization" book.

edit · good answer | 3

Updated 7 months ago by Mark Schmidt