Possible issues with gradient methods

Contents

Introduction

We are given some data $x \in X$, and we are trying to find p a probability distribution on X such that p is "typical" for x and "captures some information". More precisely, we are trying to find a p such that the bound given in the first talk (we recall that the symbol $\overset{+}{\leqslant}$ means "inferior to, up to a constant"),

$$K(x) \overset{+}{\leqslant} K(p) - \log_2(p(x)), \tag{1}$$

is "tight" (Not exactly in the usual sense, since K is not computable).

When p is a member of a parametric family ($p = p_\theta, \theta \in \Theta$), (1) becomes:

$$K(x) \overset{+}{\leqslant} K(p) + K(\theta|p) - \log_2(p_\theta(x)). \tag{2}$$

The complexity of the model $K(p)$ is usually rather small. Now, we have the problem of choosing which θ we will encode. Formally, we are trying to find:

$$\min_\theta \left(K(\theta|p) - \log p_\theta(x) \right), \tag{3}$$

or, if we use Jeffreys' prior (we will denote it by J):

$$\min_\theta \left(J(\theta) - \log p_\theta(x) \right). \tag{4}$$

To do this, we can start from some θ and make the following gradient update:

$$\theta \leftarrow \theta + \eta \frac{\partial}{\partial \theta} \left(\log J(\theta) + \log p_\theta(x) \right). \tag{5}$$

In the (frequent) case where the length of x is much larger than the dimension of Θ, we discard the $\log J(\theta)$, and our new goal is now just maximizing $\log p_\theta(x)$:

$$\theta \leftarrow \theta + \eta \frac{\partial}{\partial \theta} \left(\log p_\theta(x) \right). \tag{6}$$

1 Gradient methods

A gradient ascent (or descent) can be either continuous (for an easier theory):

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = \frac{\partial f}{\partial \theta}, \tag{7}$$

or with discrete time (in practice, all gradient ascents are discrete):

$$\theta \leftarrow \theta + \eta \frac{\partial f}{\partial \theta}. \tag{8}$$

Notice that (8) is only a first order expansion of the continuous time equation, with time step η.

It is obvious that the only guarantee we can have for a gradient ascent is that it will reach a *local* maximum of f. However, as we are going to see, there is a more fundamental problem: the gradient ascent in (7) and (8) is *not* well-defined.

1.1 Why gradient methods are not well-defined

1.1.1 Example

Consider a businessman selling electronic devices, who can change the size and the frequency of the devices: he is trying to maximize the function $\mathrm{profit}(\mathrm{size}, \mathrm{frequency})$.

Suppose now that this function has two local maximums (small size and small frequency (s_1, f_1), for mobile phones ; and larger size and frequency (s_2, f_2), for computers), and that he is trying to optimize his profit with a gradient ascent, starting from (s_0, f_0).[1]

Consider the two following possibilities:

- The businessman measures the frequency in megahertz, the size in inches and the profit in dollars.
- The businessman measures the frequency in picohertz, the size in inches and the profit in dollars.

The gradient update is:

$$\theta \leftarrow \theta + \eta \frac{\partial f}{\partial \theta}, \tag{9}$$

where f is the profit, and θ is the vector $(\mathrm{size}, \mathrm{frequency})$, and η is the "gradient step size".

In the second case, the term $\frac{\partial f}{\partial \theta}$ is much (10^{12} factor) smaller than in the first case, and moreover, the updated quantity is measured in picohertz, for a

[1] We leave the problem of obtaing the derivative of the profit with respect to the size and the frequency to the businessman.

factor 10^{24} to the final change: the factors linked to a change of unit do not cancel out, they end up *squared*.

Consequently, the trajectories of the two gradient ascents will be different, and even worse: they might end up at different solutions (the second ascent essentially starts by optimizing the size at fixed frequency, and *then* modifies the frequency).

The fact that the trajectory of the optimization depends on arbitrary choices by the user is not acceptable, but in this case, even the *result* of the optimization algorithm can change, which is even worse.

1.1.2 Homogeneity

The problem with equation (9) ($\theta \leftarrow \theta + \eta\frac{\partial f}{\partial \theta}$) is that if we consider that η is a unitless real number, it is not *homogeneous*: in the example before, $\frac{\partial f}{\partial \theta}$ is measured in euros by hertz (or some multiple of euros by hertz).

It *does not make sense* to add this quantity ($\frac{\partial f}{\partial \theta}$) to a frequency ($\theta$).

In other words, η cannot be a number like 0.01: it should be for example $0.01\text{Hz}^2.\$^{-1}$.

This extremely important remark does solve the problem above, and the homogeneity of any expression should be checked. However, two different coefficients are now needed: one for the frequency and the other for the size), and the continuous formulation does not seem to make sense anymore.

Consequently, we need to look at gradient ascents more closely with a mathematical point of view.

1.2 Mathematically correct description

We are trying to maximize $f : V \to \mathbb{R}$, where V is a vector space.

The problem with the gradient ascent $v \leftarrow v + \eta\frac{\partial f}{\partial v}$, is that "$\frac{\partial f}{\partial v}$" is *not* a column vector (as the gradient should be), it is a *linear form* (or a row vector): $\frac{\partial f}{\partial v}$ takes a (column) vector as an argument, and returns a real number: by definition $\frac{\partial f}{\partial v}$ satisfies

$$f(v + \varepsilon w) = f(v) + \varepsilon\frac{\partial f}{\partial v}(w) + O(\varepsilon^2), \tag{10}$$

and the correct definition of the gradient of f is:

$$f(v + \varepsilon w) = f(v) + \varepsilon\langle\nabla f, w\rangle + O(\varepsilon^2). \tag{11}$$

Consequently, to define our gradient correctly, we need to choose a scalar product, or equivalently, a basis of V (which would be orthonormal for the scalar product): if we fix a scalar product, then the gradient ascent

$$v^{n+1} \leftarrow v^n + \eta\nabla f \tag{12}$$

is well defined: from equations (11) and (12), we can deduce that if M is such that $\langle u, v\rangle = u^T M v$, then

$$\nabla f = M^{-1}\frac{\partial f}{\partial v}. \tag{13}$$

Remark. For simpler notation, we will sometimes write $\frac{\partial f}{\partial v}$ for the column vector defined by $(\frac{\partial f}{\partial v})_i = \frac{\partial f}{\partial v_i}$ (as above[2]). With this notation, if we are using the canoncial scalar product (i.e. $M = I$), we do have $\nabla f = \frac{\partial f}{\partial v}$. But as we are going to see below, we usually do not want to use the canonical scalar product.

Indeed, with no constraints on how we can choose our scalar product, for a given time step η, *any* point w in the half-space $\frac{\partial f}{\partial v}(w) > 0$ can be the end of the gradient step. It could therefore be useful to have guidelines for choosing a scalar product.

1.3 Choosing a relevant scalar product

The following lemma shows that a gradient ascent step can be seen as the maximisation of f, with a penalty for going far away from the initial point.

Lemma 1. *The gradient ascent $v^{n+1} \leftarrow v^n + \eta \nabla f$ can be rewritten, up to $O(\eta^2)$*

$$v^{n+1} \leftarrow \text{argmax}_v \{f(v) - \frac{1}{2\eta}\|v - v^n\|^2\rangle\} \tag{14}$$

Proof. Let us rewrite the right side of (14) by replacing v by $v^n + \eta w$:

$$\begin{aligned} f(v) - \frac{1}{2\eta}\langle v - v^n, v - v^n\rangle =& f(v^n + \eta w) - \frac{1}{2\eta}\langle \eta w, \eta w\rangle & (15)\\ =& f(v^n) + \eta\langle \nabla f, w\rangle - \frac{\eta}{2}\langle w, w\rangle + O(\eta^2) & (16)\\ =& f(v^n) + \eta\langle \nabla f - \frac{w}{2}, w\rangle + O(\eta^2). & (17)\end{aligned}$$

Consequently, we want to maximize (in w) $\phi(w) := \langle \nabla f - \frac{w}{2}, w\rangle$, and it is easy to check that $w = \nabla f$ is the maximum of ϕ.

□

The penalization for the "vanilla" gradient ascent is the numerical change in our parameters: it would be better if our penalty had an intrinsic meaning. We will discuss a solution to this problem later: let us give an example first to show that we need to go a bit further.

1.3.1 Example

Suppose we are trying fo fit n observations $x_1, ..., x_n$ to a Gaussian with mean 0 and variance v.

$$\begin{aligned} \log p_v(x_1, ..., x_n) =& \log(\frac{1}{\sqrt{2\pi v^n}}) - \frac{1}{2}\frac{\sum x_i^2}{v} & (18)\\ =& -\log(\sqrt{2\pi}) - \frac{n}{2}\log v - \frac{n}{2}\frac{\hat{v}}{v}, & (19)\end{aligned}$$

where $\hat{v}$ is the observed variance.

[2] Actually, this remark is misleading: $\frac{\partial f}{\partial v}$ in $M^{-1}\frac{\partial f}{\partial v}$ really is a row vector, and $M^{-1}\frac{\partial f}{\partial v}$ is a column vector, for reasons that will not be discussed here. The right formalism is tensor calculus.

Let us compute the vanilla gradient ascent over v:

$$\frac{\partial p_v}{\partial v} = -\frac{n}{2v} + \frac{n}{2}\frac{\hat{v}}{v^2}. \tag{20}$$

The gradient ascent is therefore:

$$v \leftarrow v + \eta\frac{n}{2}(\frac{\hat{v}}{v^2} - \frac{1}{v}) = v + \eta\frac{n}{2}\frac{\hat{v} - v}{v^2}. \tag{21}$$

This gradient descent has a major problem with the step size: if $v \gg 0$, then the steps will be very small, while if $v \sim 0$, they will be too large.

This problem can be solved by optimizing over $\rho := \log v$ instead of v, and we can rewrite the corresponding gradient step:

$$\rho \leftarrow \rho + \eta\frac{\partial f}{\partial \rho}, \tag{22}$$

or, using the variable v:

$$v \leftarrow v\exp(\eta\frac{\partial f}{\partial v}e^v) \approx v + \eta e^v\frac{\partial f}{\partial v}. \tag{23}$$

In higher dimension, we would have, for some matrix M:

$$v \leftarrow v + \eta M^{-1}(v)\frac{\partial f}{\partial v}. \tag{24}$$

(24) is the most general example of gradient ascent, where the scalar product, given by $M(v)$ is allowed to depend on the point where we are, and M is called a *metric*. The update can also rewritten in a way similar to (14):

$$v^{n+1} = \text{argmax}_v\{f(v) - \frac{1}{2\eta}\|v - v^n\|^2_{v^n}\}, \tag{25}$$

and if $\|.\|_{v^n}$ is defined in a intrinsic way, then our gradient ascent (or descent) will be well-defined, in the sense that it will be insensitive to any change of variables in the continuous case (in the discrete case, this is true up to $O(\eta^2)$, but it is still exact if the change of variables is linear).

The reason why we want our gradient ascent to be insensitive to changes of variables is that the choice of a parametrization for a problem is not meaningful: the result of an optimization algorithm should *not* depend on arbitrary choices of the user.

2 Return to machine learning

Let us return to machine learning, and more precisely, equation (6), slightly modified to take Section 2 into account

$$\theta \leftarrow \theta + \eta M^{-1}(\theta)\frac{\partial}{\partial \theta}\left(\log p_\theta(x)\right), \tag{26}$$

or equivalently up to $O(\eta^2)$:

$$\theta^{n+1} = \text{argmax}_\theta\{\log p_\theta(x) - \frac{1}{2\eta}\|\theta - \theta^n\|^2_{\theta^n}\} \tag{27}$$

The naive gradient ascent ($M = I$, or equivalently, $\|\theta' - \theta\|_\theta = \sum \theta_i^2$) corresponds to penalizing a change in *the numerical values of the parameter* θ, which has no practical meaning: we should try to build a metric such that $\|\theta' - \theta\|$ depends *only* on P_θ and P_θ, and as we will see in the next talk, the Fisher metric satisfies this condition.

References

[1] Peter D. Grünwald. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning).* The MIT Press, 2007.

[2] Marcus Hutter. On universal prediction and Bayesian confirmation. *Theoretical Computer Science*, 384(1):33–48, 2007.

[3] Ming Li and Paul M.B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer Publishing Company, Incorporated, 3 edition, 2008.

[4] Ray J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7, 1964.

[5] Joel Veness, Kee Siong Ng, Marcus Hutter, and Michael H. Bowling. Context tree switching. *CoRR*, abs/1111.3182, 2011.

[6] Frans M. J. Willems. The context-tree weighting method: Extensions. *IEEE Transactions on Information Theory*, 44:792–798, 1994.

[7] Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J. Tjalkens. The context tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41:653–664, 1995.