Tracking Objects

Sebastian Damrich
Heidelberg University
sebastian.damrich@gmx.de

Lukas Bold
Heidelberg University
lukasbold92@gmail.com

Abstract

The abstract paragraph should be indented ½ inch (3 picas) onboth the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Differentiate through KKT Conditions

The OptNet Paper [] introduced a method for differentiating through a quadratic problem given by

$$\min_f \frac{1}{2} f^T Q f + c^T f$$

$$\text{subject to } \; Af - b = 0 \; \text{ and } \; Gf \leq h,$$

where $f \in \mathbb{R}^n$ is the optimization variable, $Q \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $G \in \mathbb{R}^{p \times n}$ and $h \in \mathbb{R}^p$ are the data describing the minimization problem.

Since our first purpose was the application of this process to a minimum-cost flow problem (MCFP) through a graph associated to a video, we could restrict ourselves to a linear optimization problem of the following form:

$$\min_f c^T f$$

$$\text{subject to } \; Af = b, \; f \geq 0 \text{ and } f \leq \kappa,$$

where f is the flow through the graph, c is the cost vector and κ is the capacity vector related the graph's edges. Together with suitable A, b and the inequalities $f \geq 0$, $f \leq \kappa$, this case encodes the MCFP. Now we pursue the OptNet Paper and use the definitions

$$G \coloneqq \begin{pmatrix} -I \\ I \end{pmatrix} \; \text{ and } \; h \coloneqq \begin{pmatrix} 0 \\ h \end{pmatrix}$$

to get the both restrictions $Af = b$ and $Gf \leq h$. Then we obtain the Lagrangian function

$$L(f, \nu, \lambda) = c^T f + \nu^T (Af - b) + \lambda^T (Gf - h)$$

which leads us to the KKT conditions

$$\begin{aligned} \nabla_f L(f^*, \nu^*, \lambda^*) &= c + A^T \nu^* + G^T \lambda^* = 0 \\ \nabla_\nu L(f^*, \nu^*, \lambda^*) &= Af^* - b = 0 \\ D(\lambda^*)(\nabla_\lambda L(f^*, \nu^*, \lambda^*)) &= D(\lambda^*)(Gf^* - \kappa) = 0 \end{aligned}$$

where D(-) creates a diagonal matrix consisting of the entries of a vector and f^*, ν^* and λ^* are the optimal primal and dual variables. After taking the differentials of the KKT conditions we receive

$$\begin{aligned} \mathrm{d}c + \mathrm{d}A^T\nu^* + A^T\mathrm{d}\nu + \mathrm{d}G^T\lambda^* + G^T\mathrm{d}\lambda &= 0 \\ \mathrm{d}Af^* + A\mathrm{d}f - \mathrm{d}b &= 0 \\ D(Gf^* - h)\mathrm{d}\lambda + D(\lambda^*)(\mathrm{d}Gf^* + G\mathrm{d}f - \mathrm{d}h) &= 0. \end{aligned}$$

Indeed the following matrix form is equivalent to the upper equations:

$$\begin{pmatrix} 0 & G^T & A^T \\ D(\lambda^*)G & D(Gf^* - h) & 0 \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathrm{d}f \\ \mathrm{d}\lambda \\ \mathrm{d}\nu \end{pmatrix} = \begin{pmatrix} -\mathrm{d}c - \mathrm{d}G^T\lambda^* - \mathrm{d}A^T\nu^* \\ -D(\lambda^*)\mathrm{d}Gf^* + D(\lambda^*)\mathrm{d}h \\ -\mathrm{d}Af^* + \mathrm{d}b \end{pmatrix} \tag{1}$$

where the left hand matrix is from now on called M. In the next step, we are interested in the derivation $\frac{\partial \ell}{\partial c} \in \mathbb{R}^n$, where $\ell : \mathbb{R}^n \to \mathbb{R},\ f \mapsto \ell(f)$ is our loss function. So we want to know how the loss varies with respect to the costs of the graph. Later, we also discuss the approach of considering $\frac{\partial \ell}{\partial h} \in \mathbb{R}^n$, where h carries the information about the capacities κ. After differentiating the equation (1) by ∂c we obtain

$$M \begin{pmatrix} \frac{\partial f}{\partial c} \\ \frac{\partial \lambda}{\partial c} \\ \frac{\partial \nu}{\partial c} \end{pmatrix} = \begin{pmatrix} -I \\ 0 \\ 0 \end{pmatrix}.$$

If the derivation by h is desired, the right hand side has to be replaced by $(0, I, 0)^T$. For the next point, it is necessary to consider the construction of A in a few words. One axis is indexed by the number of pixels in the video plus one vertex for the sink and the other by the edges in the video's graph in lexicographic order. The most entries of A are zeros. In each column are one 1 and one -1 indicating the outgoing and incomming edges regarding the two pixels. So A is the incidence matrix of the graph and it's rank is therefore the number of rows minus 1.

In our case the top left entry of M is the zero matrix and A does not have full rank in contrast to the positive definite matrix Q and full rank matrix A in the OptNet paper in Theorem 1. Therefore, it might happen, that the above equation has no solution at all or a non-unique one, because our matrix M is not invertible. However, we can continue and try to compute a solution (if it exists), which is of course not unique. With this solution for $\frac{\partial f}{\partial c}$ and derivative $\frac{\partial \ell}{\partial f}$, computed above, we could compute the desired derivation

$$\frac{\partial \ell}{\partial c} = \frac{\partial \ell}{\partial f}\frac{\partial f}{\partial c}$$

to insert it as the gradient for the neural network.

1.1 Theoretical issues

The first issue was already mentioned above. We don't know if there exists an analytic solution (x, y, z) for the equation (1). Additionally there would be infinite many solutions $(x, y, z) + \ker M$ satisfying this equation. Exactly this could become a problem if iterated computations of these solutions with some varied parameters are necessary. The iterated solutions might not go in the same "direction", so that this could lead to unexpected effects.
One possible idea might be to add a "small" quadratic term Q and a full rank matrix A that don't change M in a too dramatically way to obtain a uniquely solvable problem.
If the system has no solution, one can use for example the least square method to get results which we can continue working with.

The feasable set of a linear program is a polyhedron, which is why the function $f : \mathbb{R}^n \to \mathbb{R}^n$, which yields the solution for the optimization problem, is hard to differentiate. Maybe it is possible to add an regularisation term to guarantee the smoothness of the boundaries of the feasable set. A similar

approach using an entropic regularisation term is mentioned in the Sinkhorn Distances Paper [].
In contrast to the method above, differentiating by capacities has much more continuous properties. If the cost vector changes, the optimal flow could jump in another edge of the polyhedron. Otherwise, if the capacities are changed, the edges of the feasable polyhedron are shifted, what is a continuous process.

It should also be refered to the fact that f itself is not a map in general. Since f is a solution of a linear program, there might be more than one optimal flow. However, in practice f is considered as the flow through a video's graph, so we hope that the different optimal flows don't cause too diverging results.

1.2 Practical issues

The main practical issue is the huge size of the problem. While the OptNet paper states that more than 1000 parameters are infeasable, we also had realized that without shrinking and cropping the original videos there is no possibility for a acceptable computation time. To be precise, the matrix A has shape

$$\big(\text{number of pixels in the video + 1, number of edges of the graph}\big)$$

where the number of edges in the graph corresponds to

$$[(\text{window size})^2 * \text{ number of frames in the video} + 1] * \text{ width } * \text{ height .}$$

This was a way too much for our computational capabilities, even if we used an extremely shrunk video. Maybe working with sparse matrices could improve the performance, since A is a very sparse matrix. We have not explored it yet, but we expect the problem size still being too big.

Additionally the QP Solver from the OptNet authors did not work, so we were not able to compare their performance and computational results with ours. It also could be possible, that their QP Solver is not even capable of solving our problem without adding a positive definite matrix Q.