Optimization on Discrete Probability Spaces and Applications to Probabilistic Control Design

Miguel Barão

Abstract—This paper addresses the iterative optimization of discrete probability distributions using a information geometry framework. Discrete probability distributions can be represented both as a mixture family or an exponential family. A Riemannian metric is introduced in these spaces given by the Fisher information matrix. The natural gradient is then computed with respect to this metric and is used in a iterative procedure for optimization. Properties of both formulations are given, and examples are presented. Finally, the formulation is illustrated in a probabilistic control design for a gene regulatory network problem.

I. INTRODUCTION

The problem considered in this paper is motivated by the design of probabilistic controllers [2], [6], [8] for controlled Markov chains which imply, in those solutions, the optimization of the Kullback-Leibler divergence. The optimization literature (e.g. [7]) refers many minimization algorithms that apply to smooth cost functions. Most methods are based on a cost function descent towards a local minimum. They provide, at each iteration, a direction computed from the gradient and, sometimes, the Hessian. Methods based on the gradient alone are usually slow, while methods using second order information, such as quasi-Newton and conjugate gradient methods, yield faster convergence rates. Quasi-Newton methods are also called variable-metric methods since the Hessian can be seen as a metric in the parameter space. This “metric” is, however, a property of the cost function and not a characteristic of the underlying parameter space. Amari suggests [1], [4], instead, the use of the Fisher metric to compute the natural gradient in probability spaces and to proceed in the resulting direction. It is shown in [3] that for a mixture parametrization, an optimization based on the natural gradient flow will satisfy the probability constraints, thus allowing the optimization to be dealt with as an unconstrained problem.

In the present paper two parametrizations are considered, namely the mixture and the exponential families. A metric and natural gradient are introduced for each one of them. The optimization following the natural gradient satisfies automatically the probability constraints. It is shown that this families are natural choices for Kullback-Leibler divergence optimization. With respect to [3], this paper derives a new formula to iterate the complete probability vector. Also, an exponential parametrization is considered.

The paper is organized as follows. Section II defines mixture and exponential families of probability mass functions, and defines the natural gradient and Fisher information in a general setting. Section III derives the natural gradient and shows how iterations can be computed both in parameter space and probability space. Section IV derives the same results for the exponential parametrization, and shows that a multiplicative update law is obtained in the probability space. Section V presents two examples, minimizing the Kullback-Leibler divergence $D(p\|q)$, and its dual $D(q\|p)$, using a mixture and an exponential parametrization, respectively. Section VI applies the mixture optimization to solve a gene regulatory network control problem. Finally, section VII draws conclusions.

II. PRELIMINARIES

Let $\mathcal{P}$ denote the set of probability mass functions (p.m.f.) $p(x)$, where $x \in \{0, \ldots, n\}$ is a finite alphabet. The p.m.f $p(x)$ can be parametrized in many ways, two such ways being the mixture and exponential families. These are the families considered in this paper.

Generally, a mixture family is defined by

$$p(x|\theta) = C(x) + \sum_{i=1}^{n} \theta^{i} F_{i}(x)$$  \hspace{1cm} (1)

while the exponential family is defined by

$$p(x|\theta) = \exp \left( C(x) + \sum_{i=1}^{n} \theta^{i} F_{i}(x) - \varphi(\theta) \right)$$  \hspace{1cm} (2)

where $\theta \in \Theta$ parametrizes $p(x|\theta)$ and works as a coordinate system for $\mathcal{P}$. $C(x)$ and $F_{i}(x)$ are selected basis...
functions, and $\varphi(\theta)$ is a normalization function enforcing $\int p(x|\theta) dx = 1$.

When performing optimization on $P$, first it is necessary to select a parametrization $\theta$, and then iterate these parameters until convergence is achieved. Here, the parameter iteration will follow a direction given by the natural gradient $\tilde{\nabla}f$ of the cost function $f : \Theta \to \mathbb{R}$. The natural gradient is defined as the solution of the equation

$$\langle \tilde{\nabla}f, v \rangle = df(v), \quad \forall v \neq 0, \tag{3}$$

where $v$ is an arbitrary vector with components $v^i$, $df$ is the differential of $f$, so that $df(v)$ gives the rate of change of $f$ along the direction $v$. The solution of equation (3) depends on the particular inner product used. Consider the following inner product:

$$\langle w, v \rangle = \sum_{i,j} w^i v^j g_{ij}, \tag{4}$$

where $g_{ij}$ is the metric tensor, and is a function of $\theta$ (changes from point to point). In Euclidean geometry, the metric is the identity $g_{ij} = \delta_{ij}$, where $\delta_{ij}$ is the Kronecker delta function, thus the natural gradient obtained from (3) is just the vector of partial derivatives. However, it has been suggested [1], [4], [5] that, when dealing with probabilities, the metric should be given by the Fisher information matrix (F.I.M.)

$$g_{ij}(\theta) = E_\theta \left[ \frac{\partial \log p(x|\theta)}{\partial \theta^i} \frac{\partial \log p(x|\theta)}{\partial \theta^j} \right]. \tag{5}$$

This implies that the natural gradient deviates from the standard one, this transformation depending on a particular point $\theta$ where the F.I.M. is computed. Hereafter, symbols $\tilde{\nabla}f$ and $\nabla f$ denote respectively the standard and natural gradients.

The next sections will specialize formulas for mixture and exponential families.

III. OPTIMIZATION OF A MIXTURE FAMILY

A mixture parametrization for $p(x|\theta)$ can be done by selecting as parameters $\theta^i = \Pr(X = i) = p(i)$, for $i = 1, \ldots, n$. Then, we have

$$p(x|\theta) = \begin{cases} 1 - \sum_{i=1}^n \theta^i & \text{if } x = 0, \\ \theta^x & \text{if } x \neq 0. \end{cases} \tag{6}$$

which is equivalent to selecting, in equation (1), the following basis functions:

$$C(x) = \delta_{0x}, \quad F_i(x) = \delta_{ix} - \delta_{0x}. \tag{7}$$

The Fisher information matrix $G = [g_{ij}]$ and its inverse $G^{-1}$ are given by

$$G = \begin{bmatrix} \theta^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1 - \sum_{k=1}^n \theta^k}{\theta^x} \end{bmatrix} + \frac{1}{1 - \sum_{k=1}^n \theta^k} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \tag{8}$$

and

$$G^{-1} = \begin{bmatrix} \theta^1 & 0 & \cdots & 0 \\ 0 & \theta^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \theta^n \end{bmatrix} \begin{bmatrix} \theta^1 \\ \vdots \\ \theta^n \end{bmatrix}. \tag{9}$$

The natural gradient $\tilde{\nabla}f$, defined in (3), is the solution of

$$(\nabla f)^T G v = [\frac{\partial f}{\partial \theta^1} \cdots \frac{\partial f}{\partial \theta^n}] v, \quad \forall v \neq 0. \tag{10}$$

Then

$$\tilde{\nabla}f = G^{-1} \begin{bmatrix} \frac{\partial f}{\partial \theta^1} \\ \vdots \\ \frac{\partial f}{\partial \theta^n} \end{bmatrix}. \tag{11}$$

The structure of $G^{-1}$ in equation (9) can be exploited (see [3]) to achieve a much faster implementation that avoids the explicit computation of $G^{-1}$. This is achieved with the formula $\tilde{\nabla}f = \theta \circ \nabla f - \theta (\theta \cdot \nabla f)$.

The natural gradient method is described by the following algorithm:

1) Compute the differential of $f$

$$df = [\frac{\partial f}{\partial \theta^1} \cdots \frac{\partial f}{\partial \theta^n}] = (\nabla f)^T. \tag{12}$$

2) Compute the natural gradient from the standard gradient $\nabla f = (df)^T$ using

$$\tilde{\nabla}f = \theta \circ \nabla f - \theta (\theta \cdot \nabla f) \tag{13}$$

where $\circ$ denotes the elementwise multiplication (Hadamard product).

3) Update $\theta$ using

$$\theta_{[k+1]} = \theta_{[k]} - \eta \tilde{\nabla}f \tag{14}$$

where the step size $\eta$ has to satisfy

$$\eta < \frac{1}{\max \alpha - \theta \cdot \nabla f} \tag{15}$$

$$\alpha = [0 \quad (\nabla f)^1 \cdots (\nabla f)^n]^T \tag{16}$$

to guaranty a feasible update.

4) Repeat the previous steps until convergence is attained. A test of convergence can be performed by checking if the rate of change $df$ of $f$ is lower than a threshold $\epsilon$, i.e.,

$$\nabla f \cdot \tilde{\nabla}f < \epsilon. \tag{17}$$

5) A probability vector $p$ can be obtained at any time from $\theta$ by

$$p = \begin{bmatrix} 1 - \sum_{i=1}^n \theta^i \\ \theta^1 \\ \vdots \\ \theta^n \end{bmatrix}. \tag{18}$$
Instead of iterating $\theta$ in the previous algorithm, it is possible to iterate the complete probability vector $p$, the main difference being the update of the additional element $p(0)$. In this case, the algorithm is the same as before with the difference that vectors are now $n+1$ dimensional. Defining
\[
p = \begin{bmatrix} p^0 \\ p^1 \\ \vdots \\ p^n \end{bmatrix}, \quad \nabla f = \begin{bmatrix} \partial f/\partial p^1 \\ \vdots \\ \partial f/\partial p^n \end{bmatrix},
\] then equations (13), (14) (15) and (17) become, respectively,
\[
\nabla f = p \circ \nabla f - p(p \cdot \nabla f) \\
p_{[k+1]} = p_{[k]} - \eta \nabla f \\
\eta < \frac{1}{\max \nabla f - p \cdot \nabla f} \\
\nabla f \cdot \nabla f < \epsilon
\]
It is known (see [3]) that using the natural gradient algorithm to iterate $\theta$ satisfies the constraints on probabilities. The same property is kept when working directly with the probability vector $p$. This is in contrast with standard optimization techniques, where this problem must be tackled as a constrained optimization problem. Here, the constraints are satisfied automatically.

IV. OPTIMIZATION OF AN EXPONENTIAL FAMILY

The discrete p.m.f. has the unusual property of being able to be parametrized either as a mixture or an exponential family. The previous section dealt with the mixture, while this section deals with the exponential parametrization.

A representation of $p(x|\theta)$ as an element from an exponential family is done by selecting as parameters and basis the following functions:
\[
C(x) = 0, \quad F_i(x) = \delta_{ix}, \\
\theta^i = \log \frac{p(i)}{1 - \sum_{k=1}^n p(k)}, \\
\varphi(\theta) = \log \left( 1 + \sum_{k=1}^n \exp \theta^k \right).
\]
Then, we have
\[
p(x|\theta) = \frac{\exp \theta^i}{1 + \sum_{k=1}^n \exp \theta^k}.
\]
The Fisher information matrix $G$ is, for this family, given by
\[
G = \begin{bmatrix} p_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & p_n \end{bmatrix} - \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \begin{bmatrix} p_1 & \cdots & p_n \end{bmatrix}
\] (28)
where $p_i = p(i|\theta)$. Note the similarity in structure between (28) and (9).

As before, the natural gradient in the parameter space is given by
\[
\tilde{\nabla} f = G^{-1} \nabla f
\]
and the update rule is
\[
\theta_{[k+1]} = \theta_{[k]} - \eta \tilde{\nabla} f.
\]
The probability vector $p$ can be obtained at any time from $\theta$ by
\[
p = \frac{1}{1 + \sum_{k=1}^n \exp \theta^k} \begin{bmatrix} \exp \theta^1 \\ \vdots \\ \exp \theta^n \end{bmatrix}
\]
(31)
The explicit use of parameters in the update rule forces the use of exponentiation to compute the probability vector $p$, which is then used to compute the natural gradient. An alternative approach is to perform the update directly over the probability vector $p$, eliminating the explicit use of $\theta$. In that formulation, the probability vector $p$ has a multiplicative update law
\[
p_{[k+1]} = p_{[k]} \beta \exp \left( - \eta \tilde{\nabla} f \right),
\]
where $\beta$ is a normalizing constant. This law is obtained by exponentiation of equation (30).

V. EXAMPLES

To illustrate the application of the natural gradient method, two low dimensional examples were built to illustrate the trajectories followed by standard and natural gradients in mixture and exponential families.

In the first example a p.m.f. $p(x|\theta)$ is sought to minimize the Kullback-Leibler divergence $D(p||q)$ to a specified target p.m.f. $q(x)$. In the second example, the minimization of the dual divergence $D(q||p)$ is sought instead. These are artificial problems as the answer $p(x) = q(x)$ is already known in both cases. Its interest stems from being a problem where the gradient and natural gradient methods can be easily compared.

A. Optimizing $D(p||q)$ in a mixture family

For the first problem, minimization of $D(p||q)$, $p$ has a mixture parametrization. The standard gradient is given by
\[
\nabla D = \begin{bmatrix} -1 & 1 & 0 \\ \vdots & \ddots & \vdots \\ -1 & 0 & 1 \end{bmatrix} (\log p - \log q),
\]
the logarithms being taken elementwise over the probability vectors $p$ and $q$. 
Let \( q = [0.2494; 0.0025; 0.7481] \) be the target p.m.f. and \( p = [\frac{1}{3}; \frac{1}{3}; \frac{1}{3}] \) the initial guess. Figure 1 shows the iterations of both standard and natural gradient methods with fixed step \( \eta \). The natural gradient performs much faster. In fact, with quadratic convergence since the Hessian of the K-L divergence approaches the metric \( G \) near the optimum.

### B. Optimizing \( D(q||p) \) in an exponential family

For the second problem, minimization of \( D(q||p) \), \( p \) is parametrized in an exponential family. Differentiating \( D(q||p) \) with respect to the parameters \( \theta_j \), yields

\[
\frac{\partial D}{\partial \theta_j} = q_j - p_j, \quad j = 1, \ldots, n.
\]

Therefore, the standard gradient is

\[
\nabla D = \begin{bmatrix} q_1 - p_1 \\ \vdots \\ q_n - p_n \end{bmatrix}.
\]

Using equations (29) and (32) yields results shown in figure 2.

### C. Remarks on the Kullback-Leibler optimization

The optimization of the Kullback-Leibler divergence is a particular problem for which the mixture and the exponential families are well suited. It is known that the Hessian of a K-L divergence converges to the Fisher information matrix \( G \) as \( p \) approaches \( q \). Thus, the natural gradient behaves as a quasi-Newton method near the optimum solution. This results applies to minimization of \( D(p||q) \) when \( p \) is a mixture, and to minimization of \( D(q||p) \) when \( p \) is an exponential p.m.f.. The convergence is therefore much faster that standard optimization techniques relying on the gradient alone. It should be emphasized that this faster convergence is obtained with linear computational complexity, while standard methods using second order information (Newton-like algorithms) have much higher computational and memory requirements.

The optimization of \( D(p||q) \) has been successfully used in probabilistic control formulations [6], [8], [2], [3], while optimization of the dual divergence \( D(q||p) \) has its applications in estimation problems.

### VI. CONTROL OF GENE REGULATORY NETWORKS

In this section, a probabilistic control problem is formulated and applied to the control of a gene regulatory network. In the problem considered here, three genes can be in one of two states – expressed and not expressed – and their state can influence the others through a probabilistic rule. The transition probabilities of each gene, depends on the state of the network and on the presence of an external substance \( U \). Figure 3 shows the interactions between them, where arrows mean that if one gene is expressed then it will tend to activate the other with some predefined probability, while bar terminated lines mean the opposite effect (one will tend to inhibit the other).

The network state is formed by the activation pattern of all of their genes \( g_0, g_1 \) and \( g_2 \). This activation pattern forms a binary word \( x = g_2 g_1 g_0 \) ranging from 000 to 111. The dynamics of the network is described by \( s(x_{t+1}|x_t, u_t) \), a
controlled Markov chain computed from the interactions between the individual genes, and depending on the exogenous variable $u$. Figure 4 shows the most likely transitions of the network, depending on the presence of substance $u$.

The purpose is to control the network by manipulation of $u$ such as to approach as best as possible a given distribution $q(x, u)$ for the network state. The problem is then formulated as follows: find the law $c(u|x)$ such that the closed loop $p(x, u)$ over some time horizon $T$ is as close as possible to $q(x, u)$. The K-L divergence is used here to measure the “distance” between these two p.m.f.. See [3] for more optimization details.

Figure 5 shows the histogram of a stochastic simulation using the optimized control law $c(u|x)$. In this figure, the symbol * represents the desired states and bars represent the actual percentage of time the network was at each state.

VII. CONCLUSIONS

This paper presented an application of the natural gradient method to optimize p.m.f. parametrized as a mixture or exponential families. Examples are presented to illustrate the natural gradient in both parametrizations. It is shown that the optimization can be performed as an unconstrained problem in both parametrizations. For the Kullback-Leibler divergence optimization problem, there is a natural choice of mixture or exponential families, depending on the order of the arguments. For the K-L problem, the algorithms have fast convergence with linear computational complexity, being practical for high dimensional problems.

An example is presented where the optimization on a mixture family is used to derive a control law for a gene regulatory network.

REFERENCES


