

# 1 An online expectation minimization algorithm

Define  $\mu = \alpha \exp(f)$ ,  $\nu = \beta \exp(g)$ ,  $x = (\mu, \nu)$ . We will change variables without warning in the following. Define the Bregman divergence

$$\begin{aligned} D_\alpha(\mu, \mu_0) &= \langle \alpha, \exp(f_0 - f) - 1 - (f_0 - f) \rangle \\ D_\beta(\nu, \nu_0) &= \langle \beta, \exp(g_0 - g) - 1 - (g_0 - g) \rangle \\ D_{\alpha, \beta}(x, x_0) &= D_\alpha(\mu, \mu_0) + D_\beta(\nu, \nu_0) \end{aligned}$$

We want to solve the objective

$$\min_x \mathcal{F}(x) \triangleq \text{KL}(\alpha, \mu) + \text{KL}(\beta, \nu) + \langle \mu \otimes \nu, \exp(-C) \rangle - 1$$

Define the prox objective

$$\begin{aligned} \mathcal{L}(x, x_t) &= 2\mathcal{F}(x_t) + \langle \nabla \mathcal{F}(x_t), x - x_t \rangle + D_{\alpha, \beta}(x, x_t) \\ &= \mathbb{E}_{\hat{\alpha} \sim \alpha, \hat{\beta} \sim \beta} \left[ 2F(x_t) + \langle \nabla \mathcal{F}(x_t), x - x_t \rangle + D_{\hat{\alpha}, \hat{\beta}}(x, x_t) \right] \end{aligned}$$

The Sinkhorn iterations then rewrites as

$$x_{t+1} = \underset{x}{\operatorname{argmin}} \mathbb{E}_{\hat{\alpha}, \hat{\beta}} \mathcal{L}_{\hat{\alpha}, \hat{\beta}}(x, x_t)$$

and online Sinkhorn

$$x_{t+1} = (1 - \eta_t)x_t + \eta_t \underset{x}{\operatorname{argmin}} \mathcal{L}_{\hat{\alpha}_t, \hat{\beta}_t}(x, x_t)$$

Probably useless ?

# 2 Variable mirror descent point of view

Consider the objective

$$\max_{f, g} \mathcal{F}(f, g) = \langle \alpha, f \rangle + \langle g, \beta, - \rangle \langle \alpha \otimes \beta, \exp(f \oplus g - C) \rangle + 1$$

The gradient reads

$$\nabla \mathcal{F}(f, g) = \left( \alpha(1 - \exp(f - T_\beta(g))), \beta(1 - \exp(g - T_\alpha(f))) \right) \in \mathcal{M}^+(\mathcal{X}^2)$$

Using the local Bregman divergence

$$\omega_t(f, g) = \langle \alpha, \exp(f_t - f) \rangle + \langle \beta, \exp(g_t - g) \rangle,$$

online Sinkhorn iterations rewrites as

$$\nabla \omega_t(f_{t+1}, g_{t+1}) = \nabla \omega_t(f_t, g_t) + \eta_t \tilde{\nabla} \mathcal{F}(f_t, g_t),$$

where

$$\tilde{\nabla} \mathcal{F}(f, g) = \left( \hat{\alpha}_t(1 - \exp(f - T_\beta(g))), \hat{\beta}_t(1 - \exp(g - T_\alpha(f))) \right) \in \mathcal{M}^+(\mathcal{X}^2)$$

is an unbiased estimate of  $\nabla \mathcal{F}(f, g)$ .

### 3 An EM point of view

The simultaneous Sinkhorn updates can be rewritten as

$$f_t, g_t = \underset{f, g}{\operatorname{argmax}} Q_t^*((f, g), (f_t, g_t)) \triangleq \mathbb{E}_{Y \sim \beta} \left[ \mathbb{E}_{X \sim \alpha} \left[ f(X) - e^{f(X) + g_t(Y) - C(X, Y)} \right] \right] \\ + \mathbb{E}_{X \sim \alpha} \left[ \mathbb{E}_{Y \sim \beta} \left[ g(Y) - e^{f_t(X) + g(Y) - C(X, Y)} \right] \right].$$

This is similar to the EM algorithm: the first expectation is on data, the second on hidden random variables. We now define the approximate functions

$$Q_t((f, g), (f_t, g_t)) = \mathbb{E}_{Y \sim \hat{\beta}_t} \left[ \mathbb{E}_{X \sim \alpha} \left[ f(X) - e^{f(X) + g_t(Y) - C(X, Y)} \right] \right] \\ + \mathbb{E}_{X \sim \hat{\alpha}_t} \left[ \mathbb{E}_{Y \sim \beta} \left[ g(Y) - e^{f_t(X) + g(Y) - C(X, Y)} \right] \right] \\ = \mathbb{E}_{X \sim \alpha} [f(X)] + \mathbb{E}_{X \sim \alpha} \left[ \sum_{i=n_t}^{n_{t+1}} b_i e^{f(X) + g_t(y_i) - C(X, y_i)} \right] \\ + \mathbb{E}_{Y \sim \beta} [g(Y)] + \mathbb{E}_{Y \sim \beta} \left[ \sum_{i=n_t}^{n_{t+1}} a_i e^{g(Y) + f_t(x_i) - C(x_i, Y)} \right]$$

Running the iterations

$$f_t, g_t = \underset{f, g}{\operatorname{argmax}} Q_t((f, g), (f_t, g_t))$$

amounts to set

$$f_t(\cdot) = -\log \sum_{i=n_t}^{n_{t+1}} b_i e^{g_t(y_i) - C(\cdot, y_i)} \quad g_t(\cdot) = -\log \sum_{i=n_t}^{n_{t+1}} a_i e^{f_t(x_i) - C(x_i, \cdot)},$$

which is the randomized Sinkhorn algorithm. Setting

$$\bar{Q}_t = (1 - \eta_t) \bar{Q}_{t-1} + \eta_t Q_t$$

and running the iterations

$$f_t, g_t = \underset{f, g}{\operatorname{argmin}} \bar{Q}_t((f, g), (f_t, g_t))$$

gives online Sinkhorn:

$$f_t(\cdot) = -\log \sum_{i=1}^{n_{t+1}} e^{q_i - C(\cdot, y_i)} \quad g_t(\cdot) = -\log \sum_{i=1}^{n_{t+1}} e^{p_i - C(x_i, \cdot)},$$

with the update rule on  $q_i, p_i$  as : see paper. Every function  $Q_t$  is parametrized by  $(p_i, q_i, x_i, y_i)_{i=(n_t, n_{t+1}]}$ , and  $\bar{Q}_t$  by  $(p_i, q_i, x_i, y_i)_{i=(0, n_{t+1}]}$ . Thus the parametrization of  $f_t, g_t$  is encoded using an argmax trick, and we recover the structure of a stochastic expectation-maximization algorithm (less the probabilistic point of view).

### 4 Stochastic approximation

Online EM: in finite dimension: Olivier Cappé and Eric Moulines (2009). “Online EM Algorithm for Latent Data Models”. In: *Journal of the Royal Statistical Society: Series B* 71.3, pp. 593–613

Applications + better explanation: Christophe Dupuy and Francis Bach (2017). “Online but Accurate Inference for Latent Variable Models with Local Gibbs Sampling”. In: *Journal of Machine Learning Research*, p. 45

Random fixed point iterations: Ya. I. Alber et al. (2012). “Stochastic Approximation Method for Fixed Point Problems”. In: *Applied Mathematics* 03.12, pp. 2123–2132

Non-asymptotic rates for SGD + Polyak-Ruppert averaging: Eric Moulines and Francis R. Bach (2011). “Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning”. In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., pp. 451–459

## 4.1 The Robbins Monroe-Algorithm

Overall, everything can be rewritten as looking for the zero of some function

$$\text{Find } x^* \text{ such that } h(x) = 0,$$

with access to an oracle  $\hat{h}(x)$  s.t.  $\mathbb{E}[\hat{h}(x)] = h(x)$  for all  $x \in \mathcal{X}$ . Then the algorithm

$$x_{n+1} = x_n - \eta_n h(x_n)$$

gives a sequence converging to  $x^*$ , provided that

$$\sum_n \eta_n = \infty, \quad \sum_n \eta_n^2 < \infty, \quad h \text{ non decreasing} \quad \mathbb{E}[h(x_n)^2 | \mathcal{F}_{n-1}] \leq \sigma^2$$

When looking for  $\min f(x)$ , we can use  $h(x) = \nabla f(x)$ . When looking for a fixed point equation

$$x = Tx,$$

we may use  $h(x) = x - T(x)$ , in which case the algorithm writes

$$x_{n+1} = (1 - \eta_n)x_n + \eta_n S(x_n),$$

where  $\mathbb{E}[S(x_n)] = x - T(x)$ , which is our case. In a Hilbert space, assuming  $T$  is contracting for the norm, i.e.

$$\|Tx - Ty\| \leq \kappa \|x - y\|,$$

it is easy to obtain convergence of  $\mathbb{E}[\|x_n - x^*\|^2]$  + rates on the mean-square convergence rate + almost sure convergence of the iterate (Alber et al., 2012).

## 4.2 Proof: basic inequality

Overall, all these references use at some point exhibits a sequence  $(\delta_n)_n$  such that

$$\delta_{n+1} \leq (1 - \eta_n)\delta_n + C\gamma_n$$

with  $\sum \eta_n = \infty$  and  $\sum \gamma_n < \infty$ . Typically  $\gamma_n = \eta_n^2$ .

E.g. from SGD, setting  $\delta_n = \mathbb{E}[\|\theta_n - \theta\|^2]$ , we have, if objective is  $L$ -smooth and  $\mu$ -strongly convex:

$$\delta_n \leq (1 - 2\mu\gamma_n + 2L^2\gamma_n^2)\delta_{n-1} + 2\sigma^2\gamma_n$$

**Problem.** We do not have access to such an equality:

- The contraction of the Sinkhorn operator is for a non-Euclidean distance
- Therefore we need to increase the sampling size with time

What we have at hand

$$0 \leq e_{t+1} \leq (1 - \tilde{\eta}_t)e_t + \tilde{\eta}_t(\|\varepsilon_{\hat{\beta}_t}\|_{\text{var}} + \|\iota_{\hat{\alpha}_t}\|_{\text{var}}).$$

with

$$\varepsilon_{\hat{\beta}}(\cdot) \triangleq f^* - T_{\hat{\beta}}g^*, \quad \iota_{\hat{\alpha}}(\cdot) \triangleq g^* - T_{\hat{\alpha}}f^*,$$

## References

- Alber, Ya. I., C. E. Chidume, and Jinlu Li (2012). “Stochastic Approximation Method for Fixed Point Problems”. In: *Applied Mathematics* 03.12, pp. 2123–2132.
- Cappé, Olivier and Eric Moulines (2009). “Online EM Algorithm for Latent Data Models”. In: *Journal of the Royal Statistical Society: Series B* 71.3, pp. 593–613.
- Dupuy, Christophe and Francis Bach (2017). “Online but Accurate Inference for Latent Variable Models with Local Gibbs Sampling”. In: *Journal of Machine Learning Research*, p. 45.
- Moulines, Eric and Francis R. Bach (2011). “Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning”. In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Curran Associates, Inc., pp. 451–459.