Notes

MSc Thesis

Saurav Shekhar, (16-947-921)
shekhars@student.ethz.ch

November 22, 2018

Abstract

All notes regarding my masters thesis project

1 External links

- Thesis pre proposal
- Tasks document
- Gitlab repository

2 Math notes

Taylor series multivariate.

$$f(\boldsymbol{r}_0 + \boldsymbol{a}t) = f(\boldsymbol{r}_0) + [\boldsymbol{a}\cdot\nabla f(\boldsymbol{r})]\Big|_{\boldsymbol{r}=\boldsymbol{r}_0} t + \frac{1}{2!}[\boldsymbol{a}\cdot\nabla][\boldsymbol{a}\cdot\nabla] f(\boldsymbol{r})\Big|_{\boldsymbol{r}=\boldsymbol{r}_0} t^2 + \dots \tag{1}$$

Radon Nikodym derivative Derivative between probability distributions.
Shannon Entropy.
Self information of event x = x is defined as $I(x) := -\log P(x)$

$$H(x) = \mathbb{E}_{x\sim P}\left[I(x)\right] = -\mathbb{E}_{x\sim P}\left[\log P(x)\right]$$

Cramer-Rao lower bound. [Balabdaoui and van de Geer, 2016] Suppose θ is an unknown deterministic parameter which is to be estimated from measurements x, distributed according to some pdf $f(x;\theta)$. The variance of any *unbiased estimator* $\hat{\theta}$ of θ is then bounded by reciprocal of Fischer Information $I(\theta)$:

$$\begin{aligned}\text{var}(\hat{\theta}) &\geq \frac{1}{I(\theta)} \text{ where}\\ I(\theta) &= \mathbb{E}\left[\left(\frac{\partial l(x;\theta)}{\partial\theta}\right)^2\right]\\ &= -\mathbb{E}\left[\frac{\partial^2 l(x;\theta)}{\partial\theta^2}\right]\end{aligned}$$

Note: See Wikipedia for other more general versions

Lipschitz continuity, smoothness, etc.. f is $M-$*Lipschitz continuous* given M if

$$|f(x) - f(y)| \leq M|x-y| \forall x, y \in \mathbb{R}$$

. If f is differentiable, Lipschitz continuity says that f has bounded derivative.

f is L-*Lipschitz smooth* if its derivatives are Lipschitz continuous with L. This is called smoothness type $C^{1,1}$ i.e

$$\forall x, y \in \mathbb{R}, ||\nabla f(x) - \nabla f(y)|| \leq L||x-y||$$

The definition does not assume convexity of f. Some other equivalent conditions are: on here.

$$g(x) = \frac{L}{2}x^\intercal x - f(x) \text{ is convex} \tag{2}$$

$$f(y) \leq f(x) + \nabla f(x)^\intercal (y-x) + \frac{L}{2}||y-x||^2, \forall x, y \tag{3}$$

$$(\nabla f(x) - \nabla f(y))^\intercal (x-y) \leq L||x-y||^2, \forall x, y \tag{4}$$

$$f(\alpha x + (1-\alpha)y) \geq \alpha f(x) + (1-\alpha)f(y) - \frac{\alpha(1-\alpha)L}{2}||x-y||^2 \forall x, y \in \mathbb{R}, \alpha \in [0,1] \tag{5}$$

$$\cdots \tag{6}$$

Strong Convextiy: f is α-strongly convex if

$$\nabla^2 f(x) \succeq \alpha I \forall x$$

Also

$$f(x+y) \geq f(x) + y^\intercal \nabla f(x) + \frac{\alpha}{2}||x-y||^2 \tag{7}$$

7 is equivalent to saying $g(x) = f(x) - \frac{\alpha}{2}||x||^2$ is convex.. Latter is equivalent to $\nabla^2 g \succeq 0 \equiv \nabla^2 f \succeq \alpha I$. more info here

Note:l- strong convexity can be written as

$$f(y) \geq f(x) + \nabla f(x)^\intercal (y-x) + \frac{l}{2}||y-x||^2 \tag{8}$$

Equation 8 is direct contrast to 3. $\frac{L}{l}$ is called condition number of matrix.

Function Space Theory, Functional Analysis and Functional Optimization. this is a good reference on Function spaces (Banach Spaces, Hilbert spaces, etc).

Functional Optimization on KL divergence.

in progress

In function space, dot product is extended as $\langle \cdot, \cdot \rangle$

$$\langle f, g \rangle = \int f(\theta) g(\theta) d\theta$$

Now,

$$\begin{aligned} \text{KL}\left(q||p\right) &= \mathbb{E}_q\left[\log q\right] - \mathbb{E}_q\left[\log p\right] \\ &= \langle q, \log \frac{q}{p} \rangle \\ &=: f(q) \end{aligned} \tag{9}$$

For functional f, derivative defined as

$$f(q + \epsilon d) = f(q) + f'(q)\epsilon + O(\epsilon^2) \tag{10}$$

$$f'(q) = \int \frac{\partial f(q)}{\partial q(\theta)} d(\theta) d\theta \tag{11}$$

$$\equiv \langle \nabla f(q), d \rangle \tag{12}$$

From equation 9,

$$\nabla f(q) := \frac{\partial f(q)}{\partial q} = \log \frac{q}{p} \tag{13}$$

In finite dimensional optimization, we look for candidates for optima by looking for $\mathbf{x}^*$ where $\nabla f(\mathbf{x}^*) = 0$. This is equivalent to asking every component of $\nabla f \stackrel{!}{=} 0$. For functions we look for q^* s.t

$$\frac{\partial f(q^*)}{\partial q(\theta)} \stackrel{!}{=} 0 \forall x \in \mathbb{R}$$

Frank Wolfe for Variational Inference Optimization.

in progress

Algorithm 5 where optimization is over probability distributions. q_t, s in [Locatello et al., 2018] is $\equiv \mathbf{x}_t, \mathbf{s}$ in all discussion on Frank-Wolfe. One is a function while the other is a vector. $\mathbf{x}_t$ can also be seen as parameters of the probability distributions but we do not use the same distance functions and gradients Objective function so it would be better to stay in function space.

Objective function f is $f(\mathbf{x}_t) \equiv f(q_t) = \mathcal{D}^{kl} q_t || p$ where p is the target distribution. $\mathbf{d}_t = \mathbf{s}_t - \mathbf{x}_t$. $||\mathbf{d}_t||$ can be replaced by a divergence metric between s and q_t like $\text{KL}\left(s_t || q_t\right)$. We can also try $\text{KL}\left(q_t || s\right)$ or others metrics. Gap g_t is always scalar.

is there an analog for this in functional space?

$$\begin{aligned} g_t &= -\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t \rangle \geq 0 \\ &= \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t \rangle - \langle \nabla f(\mathbf{x}_t), \mathbf{s}_t \rangle \end{aligned} \tag{14}$$

For probability distribution q,

$$\langle h, q \rangle = \langle q, h \rangle = \mathbb{E}_q \left[h\right] \tag{15}$$

$$\approx \frac{1}{S} \sum_{\theta_i \sim q}^{i=1..S} f(\theta_i) \tag{16}$$

i.e expectation of f under q. 14 then becomes

$$g_t = \mathbb{E}_{q_t} \left[\nabla f(q_t)\right] - \mathbb{E}_{s_t} \left[\nabla f(q_t)\right] \tag{17}$$

$$= \mathbb{E}_{q_t} \left[\log \frac{q_t}{p}\right] - \mathbb{E}_{s_t} \left[\log \frac{q_t}{p}\right] \tag{18}$$

3 Code Notes

Normal distribution edward

```
from edward.models import Normal
from keras.layers import Dense

hidden = Dense(256, activation='relu')(x_ph)
qz = Normal(loc=Dense(10)(hidden),
scale=Dense(10, activation='softplus')(hidden))
```

`Edward` has issues with `Tensorflow versions > 1.7` see #893. Issue is non-trivial and it doesn't seem like there is a plan to fix since `Edward2` is part of `tensorflow_probability`.

4 Web pages

1. Zen of gradient descent with intro to Nexterov Method

2. I am a bandit Nesterov accelerated

3. CMU Stats FW lecture

5 Literature notes

Variational inference.: [Blei et al., 2016]

in progress

$$\overbrace{\mathrm{KL}\left(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})\right)}^{\downarrow \text{Objective} \geq 0} := \mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z})\right] - \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{z}|\mathbf{x})\right] \tag{19}$$

$$\mathrm{ELBO}(q) := \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{z},\mathbf{x})\right] - \mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z})\right] \tag{20}$$

$$\mathrm{KL}\left(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})\right) + \underbrace{\mathrm{ELBO}(q)}_{\uparrow \text{Optimize}} = \underbrace{\log p(\mathbf{x})}_{\text{constant w. } q} \tag{21}$$

$$\mathrm{ELBO}(q) = \mathbb{E}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left(q(\mathbf{z})||p(\mathbf{z})\right) \tag{22}$$

$$\mathrm{ELBO}(q) = \mathcal{Q}(\theta,\theta_t) - \mathcal{H}(\mathbf{z}|\mathbf{x}) \text{ //Entropy} \tag{23}$$

Algorithm 1: Coordinate Ascent for VI

Input: A model $p(\mathbf{x},\mathbf{z})$, a data set $\mathbf{x}$
Output: A variational density $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$

Initialize: Variational factors $q_j(z_j)$
while *the ELBO has not converged* **do**
  for $j \in \{1,\ldots,m\}$ **do**
    Set $q_j(z_j) \propto \exp\{\mathbb{E}_{-j}[\log p(z_j \,|\, \mathbf{z}_{-j},\mathbf{x})]\}$
  end
  Compute $\mathrm{ELBO}(q) = \mathbb{E}\left[\log p(\mathbf{z},\mathbf{x})\right] - \mathbb{E}\left[\log q(\mathbf{z})\right]$
end
return $q(\mathbf{z})$

Exponential families conditional conjugacy. [Bauer, 2018]

define conditional conjugacy properly

Algorithm 2: VI with conjugate family assumption

Input: A model p, variational family $q_{\phi(z)}, q_\lambda(z)$
while *ELBO is not converged* **do**
  for *each data point* i **do**
    Update $\varphi_i \leftarrow \mathbb{E}_\lambda\left[\eta_l(\beta, x_i)\right]$
  end
  Update $\lambda \leftarrow \mathbb{E}_\varphi\left[\eta_g(x,z)\right]$
end

Gradient Optimization for ELBO. We will try to solve the optimization problem from Gradient ascent perspective. This will open up opportunity for stochastic optimization [Robbins and Monro, 1951] [Robbins and Monro, 1985].

Moving from Gradient Opt to Stochastic VI

1. subsample a data point t from full data
2. use current global param λ to update local param φ_t
3. update λ

Gradient optimization step $\lambda_{t+1} = \lambda_t + \delta \nabla_\lambda f(\lambda_t)$. An equivalent formulation (for small $d\lambda$) is

$$\arg\max_{d\lambda} f(\lambda + d\lambda) \text{ st. } ||d\lambda||^2 \leq \epsilon \tag{24}$$

Here we have euclidean distance metric, which is not the best choice for probability distributions. For ex - $q_\lambda \sim \mathcal{N}(0, 1000)$ is much closer distribution to $q_{\lambda''} \sim \mathcal{N}(10, 10000)$ than $q_{\lambda'} \sim \mathcal{N}(0, 0.001)$ is to $q_{\lambda'''}\mathcal{N}(0.1, 0.001)$ even though $||\lambda - \lambda''|| \geq ||\lambda' - \lambda'''||$

Natural gradient of ELBO: *natural gradient* accounts for geometric structure of probability parameters (λ). They wrap the parameter space in a sensible way such that moving in same direction in different directions amounts to equal change in symmetrized KL divergence.

$$\begin{aligned} &\arg\max_{d\lambda} f(\lambda + d\lambda) \text{ st.} \\ &D_{KL}^{sym}(q_\lambda, q_{\lambda+d\lambda}) \leq \epsilon \text{ where} \\ &D_{KL}^{sym}(q,p) = KL(q||p) + KL(p||q) \end{aligned} \tag{25}$$

We need to find Riemannian metric [1] $G(\lambda)$ which transforms euclidean distance to symmetrized KL divergence:

$$d\lambda^\intercal d\lambda = D_{KL}^{sym}(q_\lambda(\beta), q_{\lambda+d\lambda}(\beta)) \tag{26}$$

Using information geometry [2], we can also rescale the gradients in the right space:

$$\hat{\nabla_\lambda} ELBO = G^{-1}(\lambda)\nabla_\lambda ELBO \text{ where} \tag{27}$$

$$G(\lambda) = \mathbb{E}\left[\Big(\nabla_\lambda \log q_\lambda(\beta)\Big)\Big(\nabla_\lambda \log q_\lambda(\beta)\Big)^\intercal\right] \tag{28}$$

$G(\lambda)$ is the Fisher information matrix. For our model class (conjugate exponential...) We've

$$\nabla_\lambda \log q_\lambda(\beta) = t(\beta) - \mathbb{E}\left[t(\beta)\right] \tag{29}$$

Combining 29 and 28

$$G(\lambda) = \nabla_\lambda^2 a(\lambda) = a''(\lambda) \tag{30}$$

From [Hoffman et al., 2013], equation of Euclidean gradient

$$\nabla_\lambda ELBO = a''(\lambda)\Big(\mathbb{E}\left[\eta(\mathbf{x},\mathbf{z})\right] - \lambda\Big) \tag{31}$$

refresh 31 with value in [Blei et al., 2

Combining 27, 31 and 30

$$\begin{aligned} g(\lambda) = \widehat{\nabla_\lambda} ELBO &= \mathbb{E}\left[\eta(\mathbf{x},\mathbf{z})\right] - \lambda \text{ and} \\ \lambda_t &= \lambda_{t-1} + \delta_t g(\lambda_{t-1}) \\ \Rightarrow \lambda_t &= (1-\delta_t)\lambda_{t-1} + \delta_t \mathbb{E}\left[\eta(\mathbf{x},\mathbf{z})\right] \end{aligned} \tag{32}$$

Algorithm 3: VI with conjugate family assumption

Input: A model p, variational family $q_{\phi(z)}, q_\lambda(z)$

1 **while** *ELBO is not converged* **do**
2 **for** *each data point i* **do**
3 Update $\varphi_i \leftarrow \mathbb{E}_\lambda\left[\eta_l(\beta, x_i)\right]$
4 **end**
5 Update $\lambda \leftarrow (1-\delta_t)\lambda + \delta_t \mathbb{E}_{q(\varphi)}\left[\eta_g(x,z)\right]$
6 **end**

Stochastic Variational inference. in Algorithm 3 line 2-4, we have to iterate over all data to compute the new set of local variables φ. This does not scale well to large datasets. [Hoffman et al., 2013] So we have to use stochastic gradients. Noisy gradients H of f will converge to a local optimum as long as

- $\mathbb{E}\left[H\right] = \nabla f$
- Step size δ_t st: $\sum_1^\infty \delta_t = \infty$ and $\sum_1^\infty \delta_t^2 < \infty$

Now,

$$\mathbb{E}\left[\eta(\mathbf{x},\mathbf{z})\right] = \left(\alpha_1 + \sum_1^n \mathbb{E}_q\left[t(z_i, x_i)\right], n + \alpha_2\right)$$

Noisy gradient by sampling

[1] seems to be some kind of transformation
[2] Hope so

1. Sample $t \sim Uniform(1, \dots, n)$
2. Rescale

$$\begin{aligned} g(\lambda) &= \left(\alpha_1 + n\mathbb{E}_q\left[t(z_t, x_t)\right], n + \alpha_2\right) - \lambda \\ &=: \hat{\lambda} - \lambda \end{aligned}$$

Algorithm 4: Stochastic VI

Input: A model $p(\mathbf{x}, \mathbf{z})$, data $\mathbf{x}$
Initialize:variational family $q_{\phi(z)}, q_\lambda(z)$ with params λ_0
Result: Global variational densities $q_\lambda(\beta)$
while *Stopping criteria not met* **do**
  Sample $t \sim Uniform(1, \dots, n)$
  Update $\phi_t \leftarrow \mathbb{E}_\lambda\left[\eta_l(\beta, x_t)\right]$
  Compute global param estimate $\hat{\lambda} = \mathbb{E}_\varphi\left[\eta_g(z_t, x_t)\right]$
  Update $\lambda \leftarrow (1 - \delta_t)\lambda + \delta_t\hat{\lambda}$
end
return λ

Research on optimizing difficult variational objectives with Monte Carlo (MC) estimates. Write gradient of ELBO as expectation, compute MC estimates, use stochastic optimization with MC estimates. New approaches avoid any model-specific derivations, and are called 'Black-box' inference techniques. As examples, see - [Kingma and Welling, 2013] [Rezende et al., 2014] [Ranganath et al., 2014] [Ranganath et al., 2016] [Titsias and Lázaro-[Kucukelbir et al., 2017]

$$\text{ELBO} = \mathbb{E}_{q_\nu}\left[\log p_\theta(z, x)\right] - \mathbb{E}_q\left[\log q_\nu z\right]$$

ν params of variational family, θ params of model. We need unbiased estimates of $\nabla_{\nu,\theta} ELBO$ to maximize ELBO.

Black Box variational inference.

in progress

From [Ranganath et al., 2014]

> We will form the derivative of the objec- tive as an expectation with respect to the variational approximation and then sample from the variational ap- proximation to get noisy but unbiased gradients, which we use to update our parameters. For each sample, our noisy gradient requires evaluating the joint distribution of the observed and sampled variables, the variational distribution, and the gradient of the log of the varia- tional distribution. This is a black box method in that the gradient of the log of the variational distribution and sampling method can be derived once for each type of variational distribution and reused for many models and applications.
>
> We will form the $\nabla ELBO$ as an $\mathbb{E}_{q_\lambda}[...]$ and then sample S samples from the q_λ to get noisy but unbiased gradients (w.r.t λ), which we use to update λ. For each sample, our noisy gradient requires evaluating the $p(\mathbf{x}, \mathbf{z}_S), q(\mathbf{z}_S)$, and $\nabla \log q(\mathbf{z}_S)$. This is a black box method in that the $\nabla \log q(\mathbf{z}_S)$ and sampling method can be derived once for each type of variational distribution and reused for many models and applications.

Equation (2) of [Ranganath et al., 2014]

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_q\left[\nabla_\lambda \log q(z|\lambda)\Big(\log p(x, z) - \log q(z|\lambda)\Big)\right] \text{ where} \tag{33}$$

$$\mathcal{L}(\lambda) \triangleq \mathbb{E}_{q_{\lambda_z}}\left[\log p(x, z) - \log q(z)\right] \text{ (ELBO)}$$

here it says that Equation 2/3 can be derived simply using the log trick but the authors use a complicated method in paper. Also derived in [Jalil Taghia and Schn, 2018] and [Bauer, 2018]
the gradient $\nabla_\lambda \log q(z|\lambda)$ of the log of a probability distribution is called the score function or REINFORCE

Basic algorithm

$$z_s \sim q(z|\lambda) \text{ for } s \in 1..S$$
$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda \log q(z_s|\lambda)\Big(\log p(x, z_s) - \log q(z_s|\lambda)\Big) \tag{34}$$

Rao-Blackwellization and smart Control Variates to control variance
Variance still very high. Reparameterization and amortization come to rescue (See this tutorial from David Blei)

Good notes on Stochastic VI and Black Box VI from [Jalil Taghia and Schn, 2018]

Reparameterization trick.

todo

Boosting Variational inference. [Guo et al., 2016]

in progress

Iterative boosting by $q_{i+1} = (1-\gamma)q_i + \gamma h_i$. Very similar to Frank-Wolfe. Optimization is to find optimal γ and h_i at every step. γ is very similar to line search method for [Locatello et al., 2018] and the method is exactly same (stochastic gradient descent by taking expectations). For h_i a *Laplacian Gradient Boosting* technique is used.

Frank-Wolfe.

in progress

[Jaggi, 2013] [Pedregosa, 2018] [Pedregosa et al., 2018] [Demyanov and Rubinov, 1970] *Idea:* Approximate the objective funciton f at iterate $\mathbf{x}_t$ using a linear function:

$$\tilde{f}(\mathbf{s}) := f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{s} - \mathbf{x}_t \rangle$$

Find $\mathbf{s}$ which minimizes this Linear problem (LMO) and then move in that direction by step size γ. Approximate solutions to the linear problem also suffice. Here $x, \mathcal{D} \equiv q, \mathcal{A}$

Algorithm 5: Frank-Wolfe

Constrained Optimization: $\min_{x \in \mathcal{D}} f(\mathbf{x})$
f is Convex, differentiable with L-Lipschitz gradient and domain $\mathcal{D}$ is Convex and compact
for $t \in \{0, \ldots, T\}$ **do**
  $s^t \leftarrow \arg\min_{s \in \mathcal{D}} \langle \mathbf{s}, \nabla f(\mathbf{x}^t) \rangle$
  $\mathbf{x}^{t+1} \leftarrow$ **UpdateRule**$(\mathbf{x}^t, s^t, t, f)$
end

UpdateRule can be

Constraint $\mathcal{D}$	LMO problem
norm $\|\|x\|\| \leq 1$	$-\partial\|\|\cdot\|\|_*$ Subgradients of corresponding dual norm
l_1 norm $\|\|x\|\|_1 \leq 1$	$-\partial\|\|\nabla f(\mathbf{x}_t)\|\|_\infty$
Trace norm $\underbrace{\|\|X\|\|_{tr} \leq 1}_{\text{sum of singular values}}$	Operator norm $\underbrace{s_t \in -\|\|\nabla f(X_t)\|\|_{op}}_{\text{Largest singular value}}$

Table 1: LMO problem for well known constraints

$$q^{t+1} \leftarrow (1-\gamma)q^t + \gamma s^t = q^t + \gamma \overbrace{(s^t - q^t)}^{d_t} \text{ where}$$

$$\textbf{Variant0} : \gamma \leftarrow \frac{2}{t+2} \tag{35}$$

$$\textbf{Variant1} : \gamma \leftarrow \arg\min_{\gamma \in [0,1]} f((1-\gamma)q^t + \gamma s^t) \tag{36}$$

$$g_t \leftarrow -\langle \nabla f(\mathbf{x}_t), d_t \rangle\rangle$$

$$\textbf{Exitcondition} : g_t < \delta$$

$$\textbf{Variant2} : \gamma \leftarrow \min\Big(\frac{g_t}{L||d_t||^2}, 1\Big) \tag{37}$$

Variant3 :

$$q^{t+1} \in \arg\min_{q \in conv\{x^0, s^0, s^1, \ldots, s^t\}} f(q) \tag{38}$$

(39)

37 has variants [Pedregosa, 2018] [Demyanov and Rubinov, 1970] add more

$$\gamma \leftarrow \min\Big\{\frac{g_t}{L \text{ diam}(\mathcal{D})^2}, 1\Big\}$$

Frank-Wolfe Convergence.

in progress

$$\overbrace{g_t}^{Gap} := \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{s}_t \rangle$$
$$\geq \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^* \rangle$$
$$\geq \underbrace{f(\mathbf{x}_t) - f(\mathbf{x}^*)}_{\epsilon_t} \quad Convexity \tag{40}$$

(41)

Using quadratic upper bound 3 L-continous gradient can be relaxed to this, We get,

$$f(\mathbf{y})) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2}||\mathbf{y} - \mathbf{x}||^2 \tag{42}$$

$$\Rightarrow f(\mathbf{x}_{t+1}) = f\Big((1-\gamma)\mathbf{x}_t + \gamma \mathbf{s}_t\Big) \leq f(\mathbf{x}_t) - \gamma g_t + \frac{L\gamma^2}{2}||\mathbf{s}_t - \mathbf{x}_t||^2$$
$$\leq f(\mathbf{x}_t) - \gamma \epsilon_t + \frac{L\gamma^2}{2}||\mathbf{s}_t - \mathbf{x}_t||^2$$
$$\Rightarrow \epsilon_{t+1} \leq (1-\gamma)\epsilon_t + \frac{L\gamma^2}{2}\mathcal{D}_t^2 \tag{43}$$
$$\leq (1-\gamma_t)\epsilon_t + \gamma_t^2 C \Big(= \frac{L}{2} diam(\mathcal{D})\Big) \tag{44}$$

(45)

Goal: To show, for $\gamma_t = \frac{2}{t+2}$

$$\epsilon_t \leq \frac{4C}{t+2} \tag{46}$$

Using induction, for $t = 0$, $\epsilon_0 = C \leq \frac{4C}{2}$. At step t

$$\epsilon_{t+1} \leq \Big(1 - \frac{2}{t+2}\Big)\epsilon_t + \Big(\frac{2}{t+2}\Big)^2 C$$
$$\leq \Big(\frac{t}{t+2}\Big) \cdot \Big(\frac{4C}{t+2}\Big) + \Big(\frac{1}{t+2}\Big) \cdot \Big(\frac{4C}{t+2}\Big)$$
$$= \frac{4C}{t+2}\frac{t+1}{t+2}$$
$$\leq \frac{4C}{t+2}\frac{t+2}{t+3}$$
$$= \frac{4C}{t+1+2}$$

Revisiting Frank-Wolfe: Projection Free Sparse Convex Optimization. [Jaggi, 2013]

in progress

Variants of FW in 6

- **Approximate LMO**: $\acute{\epsilon} := \frac{1}{2}\delta\gamma C_f$ additive approximate error
- **Fully Corrective**: in Equation 38, is only a degenerate boosting as only one atom $\mathbf{s}$ is chosen at each time. If we change the search space to

$$q^{t+1} \leftarrow \underset{q \in conv(\bigcup_{i=1}^{t} s^t)}{\arg\min} f(q)$$

Then the progress made per iteration would be more but the search problem would not be much easier than the original problem

Curvature Constant C_f of a convex and differentiable f:

$$C_f := \sup_{\mathbf{x},\mathbf{s}\in\mathcal{D},\gamma\in[0,1],\mathbf{y}=\mathbf{x}+\gamma(\mathbf{s}-\mathbf{x})} \frac{2}{\gamma^2}\Big(f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x})\rangle\Big)$$

Note that $C_f = \frac{2}{\gamma^2}\Big(f - \tilde{f}\Big)$ means that for bounded C_f, deviation of f from $\tilde{f}$ will also be bounded. $f - \tilde{f}$ is also called *Bregman divergence*. If ∇f is L-Lipschitz continuous on $\mathcal{D}$ w.r.t some norm $||\cdot||$, then $C_f \leq \text{diam}_{||\cdot||}(\mathcal{D}^2)L$

Boosting Black Box Variational inference.

in progress

Boosting introduced in [Guo et al., 2016], connection with FW in [Locatello et al., 2017]. define a Linear Minimization Problem (LMO) as $\mathbf{LMO}_{\mathcal{A}}(y) := \arg\min_{s\in\mathcal{A}}\langle y, s\rangle$ In line 3 of 5, rewrite it as

$$s^t \leftarrow (\delta - \text{Approx-})\mathbf{LMO}_{\mathcal{A}}(\nabla f(q^t))$$

Algorithm for LMO in section 4 of [Locatello et al., 2018]. In Theorem 2, Curvature $\mathcal{C}_{f,\mathcal{A}}$ is bounded for D^{KL} if param. space of densities in $\mathcal{A}$ is bounded. In section 3, a bounded curvature for D^{KL} is obtained.

Black box LMO:
In this case $f(q^t) = \text{KL}\left(q^t(\mathbf{z})||p(\mathbf{x},\mathbf{z})\right)$. Assuming θ are the parameters defining variational family $\mathcal{Q} \equiv \mathcal{A}$ We've to find $\nabla_\theta f(q^t)$, more specifically, we've to find

$$s^t \leftarrow (\delta - \text{ Approx.}) \arg\min_{s\in\mathcal{A}} \langle \nabla \text{KL}\left(q^t(\mathbf{z})||p(\mathbf{x},\mathbf{z})\right), s\rangle$$

Convergence of SGD not fully understood. To guarantee convergence of FW, solution of LMO should not be degenerate. This translates to a constraint on $||s||_\infty$ which is not practical. Every pdf with bounded $||\cdot||_\infty$ has bounded entropy and the converse holds true in most cases of interest. (Gaussian, Laplacian, ...). Assume $\mathcal{A}$ is such a family and $\bar{\mathcal{A}}$ is $\mathcal{A}$ w/o l_∞ norm constraint.

add how [Guo et al., 2 [Locatello et deal with optimization of LMO. Also add the part about conv($\mathcal{A}$) being sufficient instead of $\mathcal{A}$.

$$\arg\min_{s\in\bar{\mathcal{A}},\mathcal{H}(s)\geq -M} \langle \nabla \text{KL}\left(q^t(\mathbf{z})||p(\mathbf{x},\mathbf{z})\right), s\rangle \stackrel{?}{=} \arg\min_{s\in\bar{\mathcal{A}},\mathcal{H}(s)\geq -M} \left\langle s, \log\frac{q^t}{p}\right\rangle$$

Using Lagrange multiplier λ

$$\left\langle s, \log\left(\frac{s}{\sqrt[\lambda]{\frac{p}{q^t}}}\right)\right\rangle$$

$$\equiv \arg\min_{s\in\bar{\mathcal{A}}} \text{KL}\left(s||\sqrt[\lambda]{\frac{p}{q^t}}Z\right)$$

$$\text{RELBO}(s,\lambda) := \mathbb{E}_s\left[\log p\right] - \mathbb{E}_s\left[\log q^t\right] - \lambda\mathbb{E}_s\left[\log s\right] \tag{47}$$

ask

For true LMO solution, will need to maximize for λ. Might end in saddle, fix or slowly decrease with time $\frac{1}{\sqrt{t+1}}$

Adaptive step size.

Read paper and add summary

6 Ideas

6.1 Line search

Line search in 36 is not working very well.

line search

$$\gamma' = \underset{\gamma \in [0,1]}{\operatorname{argmin}} \underbrace{KL\left(q^t + \gamma(s-q^t) \,\|\, p\right)}_{\tilde{D}_{KL}(\gamma)} = \nabla_\gamma f(\tilde{q}_t)$$

grad descent, $\gamma \leftarrow \gamma - \eta \nabla_\gamma \tilde{D}_{KL}(\gamma)$

func. grad, $\nabla_{x_0} f(x)$

$$= \log \frac{x}{p}$$

$$-g_n = \langle \nabla f(x), s \rangle - \langle \nabla f(x), x \rangle$$

$$\nabla_\gamma KL\left(q^t + \gamma(s-q^t) \,\|\, p\right)$$

$$\int \underbrace{\left(q^t + \gamma(s-q^t)\right)}_{\tilde{q}_t} \log\left(\frac{q^t + \gamma(s-q^t)}{p}\right) dx$$

$$= \nabla_\gamma \int \tilde{q}_t \log \frac{\tilde{q}_t}{p}$$

$$= \int \left(\nabla_\gamma \tilde{q}_t\right) \log \frac{\tilde{q}_t}{p} + \tilde{q}_t \nabla_\gamma \log \frac{\tilde{q}_t}{p}$$

$\nabla_\gamma \tilde{q}_t \perp s - q^t$

$\nabla_\gamma \log \tilde{q}_t = \frac{1}{\tilde{q}_t} \nabla_\gamma \tilde{q}_t$, $\nabla_\gamma \tilde{q}_t \propto (s-q^t)$

$$= \int (s-q^t) \log \frac{\tilde{q}_t}{p} + (s-q^t)$$

$$= \int (s-q^t)\left(1 + \log \frac{\tilde{q}_t}{p}\right)$$

$$= \int s \log \frac{\tilde{q}_t}{p} - \int q^t \log \frac{\tilde{q}_t}{p}$$

$$= \langle s, \log \frac{\tilde{q}_t}{p} \rangle - \langle q^t, \log \frac{\tilde{q}_t}{p} \rangle$$

$$= \mathbb{E}_s\left[\log \tilde{q}_t - \log p\right] - \mathbb{E}_{q^t}\left[\log \tilde{q}_t - \log p\right]$$

$$= \mathbb{E}_s\left[\nabla_{\tilde{q}_t} f(\tilde{q}_t)\right] - \mathbb{E}_{q^t}\left[\nabla_{\tilde{q}_t} f(\tilde{q}_t)\right]$$

changes for measuring variance of $\mathbb{E}_s[\cdot]$ and $\mathbb{E}_{q^{\gamma}_{t+1}}[\cdot]$.

```
...
grad_gamma = []
for it in range(n_steps):
...
# Samples w.r.t s
rez_s = np.asarray([
    px_qx_ratio_log_prob(sample_s[ss]) for ss in range(len(sample_s))
])
# Samples w.r.t q_{t+1}
rez_q = np.asarray([
    px_qx_ratio_log_prob(sample_q[ss]) for ss in range(len(sample_q))
])
grad_gamma.append({'E_s': rez_s, 'E_q': rez_q, 'gamma': gamma})
...
# Write grad_gamma to outdir/line_search_samples_<n_samples>.npy.<fw_iter>
```

Metrics on original version.

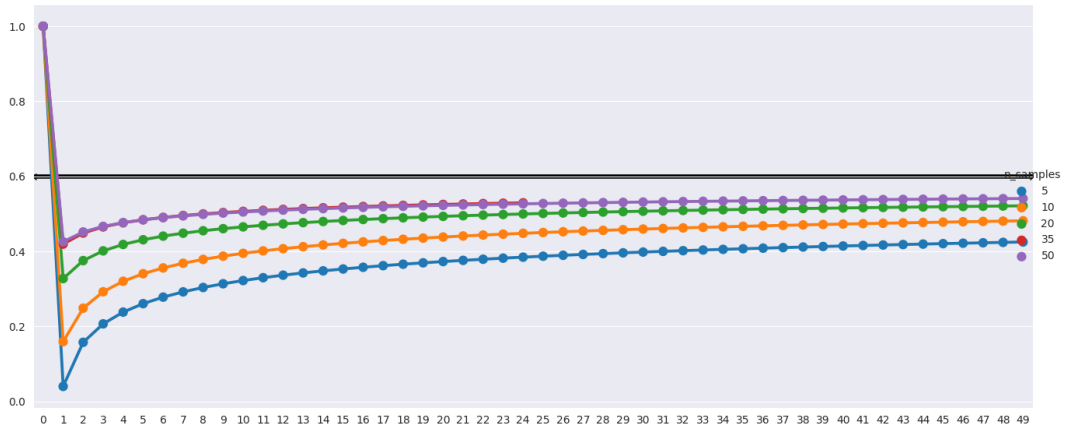

Figure 1: gamma with iterations for different n_samples

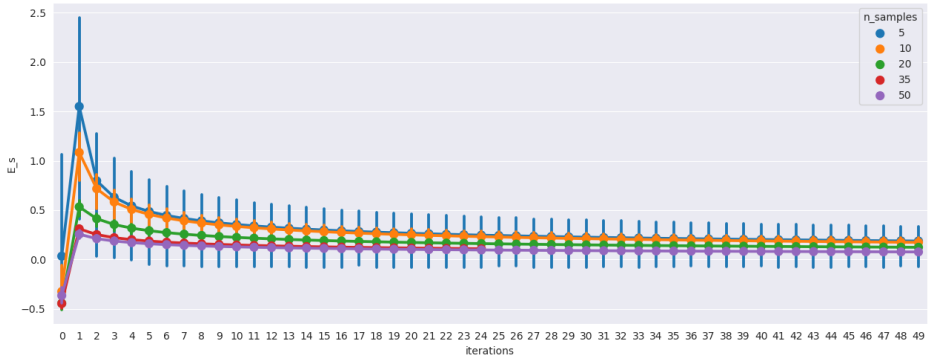

Figure 2: E_s with different n_samples

6.2 Adaptive Frank-Wolfe from smoothness estimators

Algorithm 1 of [Pedregosa et al., 2018]

Code changes.

6.3 Measuring smoothness

in progress

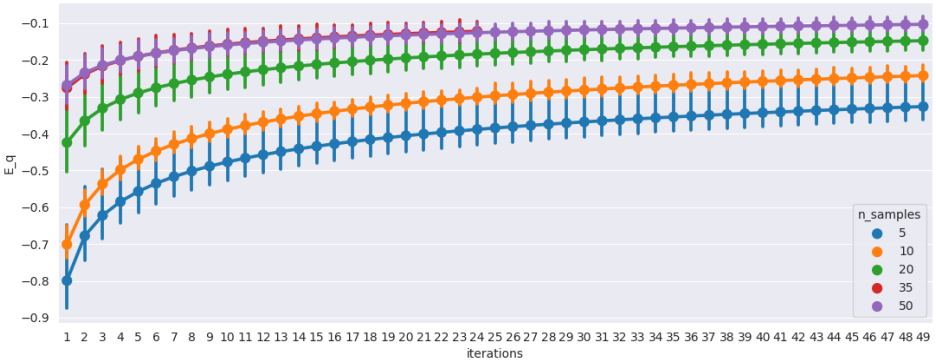

Figure 3: E_q with different n_samples
?? begins with iter 1 as iter 0 has very high variance

Algorithm 6: Adaptive Frank-Wolfe for Boosting BBVI

Input: $q_0 \in \mathcal{D}$, initial Lipschitz estimate L_{-1}, line search parameters $\tau > 1, \eta \in (0, 1]$

for $t = 1 \ldots T$ **do**
- $s_t \leftarrow LMO_{\mathcal{A}}(\nabla f(q^t))$
- $g_t \leftarrow \langle \nabla f(q_t), q_t \rangle - \langle \nabla f(q_t), s_t \rangle$ `// Gap` ≥ 0
- Find smallest integer i s.t
- $f(q_t + \gamma_t(s_t - q_t)) \leq Q_t(\gamma_t)$ Where
- $Q_t(\gamma) := f(q_t) - \gamma g_t + \frac{\gamma^2 L_t}{2} d(s_t, q_t)$ `// Quadratic upper bound 3`
- $L_t \leftarrow \tau^i \eta L_{t-1}$ and $\gamma_t \leftarrow \min\left(\frac{g_t}{L_t d(s_t, q_t)}, 1\right)$

end

return q_T

Computing optimal γ directly from eqn 1 of [Pedregosa et al., 2018]

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) + \gamma \langle \nabla f(\mathbf{x}_t), \mathbf{s}_t - \mathbf{x}_t \rangle + \frac{\gamma^2}{2} L_t ||\mathbf{s}_t - \mathbf{x}_t||^2$$

$$\Rightarrow L_t \geq \frac{f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) + \gamma \langle \nabla f(\mathbf{x}_t), \mathbf{s}_t - \mathbf{x}_t \rangle}{\frac{\gamma^2}{2} ||\mathbf{s}_t - \mathbf{x}_t||^2}$$

$$\frac{f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) + \gamma \langle \nabla f(\mathbf{x}_t), \mathbf{s}_t - \mathbf{x}_t \rangle}{\frac{\gamma^2}{2} \mathrm{KL}\left(s_t || q_t\right)}$$

Code changes.

```
def grad_kl(q, p, theta):
    # Functional Gradient w.r.t q ∇KL(q||p) = log q − log p
    ...


def lmo(y, p, ...):
    # f = D^{KL}(y||p)
    # ⟨∇f, y⟩ = E_y ∇f
    ...
```

6.4 Other optimization algorithm

todo

As shown in link 3, Frank-Wolfe converges slower than Projected Gradient Descent in Practice. See [Locatello et al., 2017] to see why we use FW and if it can be replaced. (will have to derive new convergence proofs and boosting won't be as integrated into the optimization algorithm as before).

6.5 Entropy Regularization and Noise addition using Optimal Transport

In LMO, [Locatello et al., 2018] uses Entropy Regularization in place of norm constrained optimization. It can be replaced with something simpler See [Tolstikhin et al., 2017] [Dong Liu, 2018] [Bernton et al., 2017] [Jordan et al., 1998] here And [Peyré et al., 2017] part 4.

6.6 Port code to Tensorflow Probability

see 3. Issue is if `tfp.edward2` will have support for Variational Inference and ELBO etc.

References

[Balabdaoui and van de Geer, 2016] Balabdaoui, F. and van de Geer, S. (2016). Fundamentals of mathematical statistics.

[Bauer, 2018] Bauer, S. (2018). Probabilistic graphical models for image analysis.

[Bernton et al., 2017] Bernton, E., Jacob, P. E., Gerber, M., and Robert, C. P. (2017). Inference in generative models using the wasserstein distance. *arXiv preprint arXiv:1701.05146*.

[Blei et al., 2016] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2016). Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*.

[Demyanov and Rubinov, 1970] Demyanov, V. F. and Rubinov, A. M. (1970). Approximate methods in optimization problems. *(Modern Analytic and Computational Methods in Science and Mathematics. IX.*

[Dong et al., 2018] Dong, L., Minh, T. V., Chatterjee, S., and Rasmussen, L. K. (2018). Entropy-regularized optimal transport generative models. *arXiv preprint arXiv:1811.06763*.

[Guo et al., 2016] Guo, F., Wang, X., Fan, K., Broderick, T., and Dunson, D. B. (2016). Boosting variational inference. *arXiv preprint arXiv:1611.05559*.

[Hoffman et al., 2013] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

[Jaggi, 2013] Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435.

[Jalil Taghia and Schn, 2018] Jalil Taghia, L. M. and Schn, T. (2018). Probabilistic machine learning.

[Jordan et al., 1998] Jordan, R., Kinderlehrer, D., and Otto, F. (1998). The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Kucukelbir et al., 2017] Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474.

[Locatello et al., 2018] Locatello, F., Dresdner, G., Khanna, R., Valera, I., and Rätsch, G. (2018). Boosting black box variational inference. *arXiv preprint arXiv:1806.02185*.

[Locatello et al., 2017] Locatello, F., Khanna, R., Ghosh, J., and Rätsch, G. (2017). Boosting variational inference: an optimization perspective. *arXiv preprint arXiv:1708.01733*.

[Pedregosa, 2018] Pedregosa, F. (2018). Notes on the frank-wolfe algorithm, part i. [Online; posted 21-March-2018].

[Pedregosa et al., 2018] Pedregosa, F., Askari, A., Negiar, G., and Jaggi, M. (2018). Step-size adaptivity in projection-free optimization. *arXiv preprint arXiv:1806.05123*.

[Peyré et al., 2017] Peyré, G., Cuturi, M., et al. (2017). Computational optimal transport. Technical report, École Normale Supérieure.

[Ranganath et al., 2014] Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

[Ranganath et al., 2016] Ranganath, R., Tran, D., and Blei, D. (2016). Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333.

[Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286.

[Robbins and Monro, 1951] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.

[Robbins and Monro, 1985] Robbins, H. and Monro, S. (1985). A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer.

[Titsias and Lázaro-Gredilla, 2014] Titsias, M. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. In *International Conference on Machine Learning*, pages 1971–1979.

[Tolstikhin et al., 2017] Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2017). Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*.