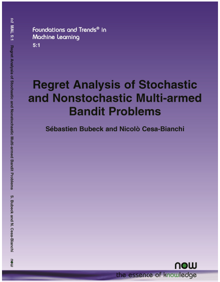
Lecture 4: Kernel-based methods for bandit convex optimization

Sébastien Bubeck
Machine Learning and Optimization group, MSR AI



Kernel-based methods

Notation: $\langle f, g \rangle := \int_{x \in \mathbb{R}^n} f(x)g(x)dx$. The expected regret with respect to point x can be written as $\sum_{t=1}^{T} \langle p_t - \delta_x, \ell_t \rangle$.

Kernel-based methods

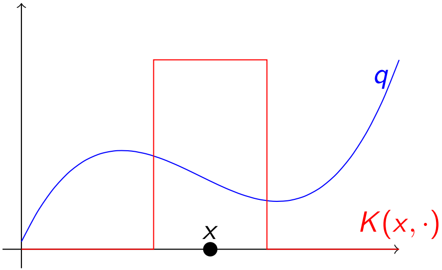
Notation: $\langle f, g \rangle := \int_{x \in \mathbb{R}^n} f(x) g(x) dx$. The expected regret with respect to point x can be written as $\sum_{t=1}^{T} \langle p_t - \delta_x, \ell_t \rangle$.

Kernel: $K : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_+$ which we view as a linear operator over measures via $Kq(x) = \int K(x, y) q(y) dy$. The adjoint K^* acts on functions: $K^* f(y) = \int f(x) K(x, y) dx$ (since $\langle Kq, f \rangle = \langle q, K^* f \rangle$).

Kernel-based methods

Notation: $\langle f, g \rangle := \int_{x \in \mathbb{R}^n} f(x)g(x)dx$. The expected regret with respect to point x can be written as $\sum_{t=1}^{T} \langle p_t - \delta_x, \ell_t \rangle$.

Kernel: $K : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_+$ which we view as a linear operator over measures via $Kq(x) = \int K(x,y)q(y)dy$. The adjoint K^* acts on functions: $K^* f(y) = \int f(x)K(x,y)dx$ (since $\langle Kq, f \rangle = \langle q, K^* f \rangle$).



Kernel-based methods

Notation: $\langle f, g\rangle := \int_{x \in \mathbb{R}^n} f(x) g(x) dx$. The expected regret with respect to point x can be written as $\sum_{t=1}^{T} \langle p_t - \delta_x, \ell_t \rangle$.

Kernel: $K : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_+$ which we view as a linear operator over measures via $Kq(x) = \int K(x,y) q(y) dy$. The adjoint K^* acts on functions: $K^* f(y) = \int f(x) K(x,y) dx$ (since $\langle Kq, f\rangle = \langle q, K^* f\rangle$).

Key point: canonical estimator of $K^* f$ based on bandit feedback on f:

$$\mathbb{E}_{x \sim q} \frac{f(x) K(x, \cdot)}{q(x)} = K^* f$$

Kernel-based methods

Notation: $\langle f, g \rangle := \int_{x \in \mathbb{R}^n} f(x) g(x) dx$. The expected regret with respect to point x can be written as $\sum_{t=1}^{T} \langle p_t - \delta_x, \ell_t \rangle$.

Kernel: $K : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_+$ which we view as a linear operator over measures via $Kq(x) = \int K(x, y) q(y) dy$. The adjoint K^* acts on functions: $K^* f(y) = \int f(x) K(x, y) dx$ (since $\langle Kq, f \rangle = \langle q, K^* f \rangle$).

Key point: canonical estimator of $K^* f$ based on bandit feedback on f:

$$\mathbb{E}_{x \sim q} \frac{f(x) K(x, \cdot)}{q(x)} = K^* f$$

Kernelized regret? Say p_t is full info strat with $\widetilde{\ell}_t = \frac{\ell_t(x_t) K_t(x_t, \cdot)}{q_t(x_t)}$ and $x_t \sim q_t$.

Kernel-based methods

Notation: $\langle f, g \rangle := \int_{x \in \mathbb{R}^n} f(x) g(x) dx$. The expected regret with respect to point x can be written as $\sum_{t=1}^{T} \langle p_t - \delta_x, \ell_t \rangle$.

Kernel: $K : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_+$ which we view as a linear operator over measures via $Kq(x) = \int K(x,y) q(y) dy$. The adjoint K^* acts on functions: $K^* f(y) = \int f(x) K(x,y) dx$ (since $\langle Kq, f \rangle = \langle q, K^* f \rangle$).

Key point: canonical estimator of $K^* f$ based on bandit feedback on f:

$$\mathbb{E}_{x \sim q} \frac{f(x) K(x, \cdot)}{q(x)} = K^* f$$

Kernelized regret? Say p_t is full info strat with $\widetilde{\ell}_t = \frac{\ell_t(x_t) K_t(x_t, \cdot)}{q_t(x_t)}$ and $x_t \sim q_t$. Then we can hope to control the regret with terms $\langle p_t - \delta_x, K_t^* \ell_t \rangle = \langle K_t(p_t - \delta_x), \ell_t \rangle$ while we want to control $\langle q_t - \delta_x, \ell_t \rangle$.

Kernel-based methods

Notation: $\langle f, g\rangle := \int_{x\in\mathbb{R}^n} f(x)g(x)dx$. The expected regret with respect to point x can be written as $\sum_{t=1}^T \langle p_t - \delta_x, \ell_t\rangle$.

Kernel: $K : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_+$ which we view as a linear operator over measures via $Kq(x) = \int K(x, y)q(y)dy$. The adjoint K^* acts on functions: $K^* f(y) = \int f(x)K(x, y)dx$ (since $\langle Kq, f\rangle = \langle q, K^* f\rangle$).

Key point: canonical estimator of $K^* f$ based on bandit feedback on f:

$$\mathbb{E}_{x\sim q} \frac{f(x)K(x, \cdot)}{q(x)} = K^* f$$

Kernelized regret? Say p_t is full info strat with $\widetilde{\ell}_t = \frac{\ell_t(x_t)K_t(x_t,\cdot)}{q_t(x_t)}$ and $x_t \sim q_t$. Then we can hope to control the regret with terms $\langle p_t - \delta_x, K_t^* \ell_t\rangle = \langle K_t(p_t - \delta_x), \ell_t\rangle$ while we want to control $\langle q_t - \delta_x, \ell_t\rangle$. Seems reasonable to take $q_t := K_t p_t$ and then we want:

$$\langle K_t p_t - \delta_x, \ell_t\rangle \lesssim \langle K_t(p_t - \delta_x), \ell_t\rangle$$

A good kernel for convex losses

$$\langle K_t p_t - \delta_x, \ell_t \rangle \lesssim \langle K_t(p_t - \delta_x), \ell_t \rangle$$

A good kernel for convex losses

$$\langle K_t p_t - \delta_x, \ell_t \rangle \lesssim \langle K_t(p_t - \delta_x), \ell_t \rangle$$

Thus for a given p we want a kernel K such that $\forall x$ and f convex one has (for some $\lambda \in (0,1)$)

$$\langle Kp - \delta_x, f \rangle \leq \frac{1}{\lambda} \langle K(p - \delta_x), f \rangle \Leftrightarrow K^* f(x) \leq (1-\lambda) \langle Kp, f \rangle + \lambda f(x)$$

A good kernel for convex losses

$$\langle K_t p_t - \delta_x, \ell_t \rangle \lesssim \langle K_t(p_t - \delta_x), \ell_t \rangle$$

Thus for a given p we want a kernel K such that $\forall x$ and f convex one has (for some $\lambda \in (0,1)$)

$$\langle Kp - \delta_x, f \rangle \leq \frac{1}{\lambda} \langle K(p - \delta_x), f \rangle \Leftrightarrow K^* f(x) \leq (1-\lambda)\langle Kp, f \rangle + \lambda f(x)$$

Natural kernel: $K\delta_x$ is the distribution of $(1-\lambda)Z + \lambda x$ for some random variable Z to be defined. Indeed in this case one has

$$K^* f(x) = \mathbb{E} f((1-\lambda)Z + \lambda x) \leq (1-\lambda)\mathbb{E} f(Z) + \lambda f(x)$$

A good kernel for convex losses

$$\langle K_t p_t - \delta_x, \ell_t \rangle \lesssim \langle K_t(p_t - \delta_x), \ell_t \rangle$$

Thus for a given p we want a kernel K such that $\forall x$ and f convex one has (for some $\lambda \in (0,1)$)

$$\langle Kp - \delta_x, f \rangle \leq \frac{1}{\lambda} \langle K(p - \delta_x), f \rangle \Leftrightarrow K^* f(x) \leq (1-\lambda)\langle Kp, f \rangle + \lambda f(x)$$

Natural kernel: $K\delta_x$ is the distribution of $(1-\lambda)Z + \lambda x$ for some random variable Z to be defined. Indeed in this case one has

$$K^* f(x) = \mathbb{E} f((1-\lambda)Z + \lambda x) \leq (1-\lambda)\mathbb{E} f(Z) + \lambda f(x)$$

Thus we would like Z to be equal to Kp, that is Z satisfies the following distributional identity, where $X \sim p$,

$$Z \stackrel{D}{=} (1-\lambda)Z + \lambda X$$

Generalized Bernoulli convolutions

$$Z \stackrel{D}{=} (1-\lambda)Z + \lambda X$$

Generalized Bernoulli convolutions

$$Z \overset{D}{=} (1-\lambda)Z + \lambda X$$

We say that Z is the *core* of p. It satisfies $Z = \sum_{k=0}^{+\infty} \lambda(1-\lambda)^k X_k$ with (X_k) i.i.d. sequence from p. We need to understand the "smoothness" of Z (which will translate in smoothness of the corresponding kernel).

Generalized Bernoulli convolutions

$$Z \stackrel{D}{=} (1-\lambda)Z + \lambda X$$

We say that Z is the *core* of p. It satisfies $Z = \sum_{k=0}^{+\infty} \lambda(1-\lambda)^k X_k$ with (X_k) i.i.d. sequence from p. We need to understand the "smoothness" of Z (which will translate in smoothness of the corresponding kernel).

Consider the core ν_λ of a random sign (this is a distinguished object introduced in the 1930's known as a Bernoulli convolution):

Generalized Bernoulli convolutions

$$Z \overset{D}{=} (1-\lambda)Z + \lambda X$$

We say that Z is the *core* of p. It satisfies $Z = \sum_{k=0}^{+\infty} \lambda(1-\lambda)^k X_k$ with (X_k) i.i.d. sequence from p. We need to understand the "smoothness" of Z (which will translate in smoothness of the corresponding kernel).

Consider the core ν_λ of a random sign (this is a distinguished object introduced in the 1930's known as a Bernoulli convolution):

- Wintner 1935: ν_λ is either absolutely continuous or singular w.r.t. Lebesgue. For $\lambda \in (1/2, 1)$ is it singular, and for $\lambda = 1/2$ it is a.c.

Generalized Bernoulli convolutions

$$Z \overset{D}{=} (1-\lambda)Z + \lambda X$$

We say that Z is the *core* of p. It satisfies $Z = \sum_{k=0}^{+\infty} \lambda(1-\lambda)^k X_k$ with (X_k) i.i.d. sequence from p. We need to understand the "smoothness" of Z (which will translate in smoothness of the corresponding kernel).

Consider the core ν_λ of a random sign (this is a distinguished object introduced in the 1930's known as a Bernoulli convolution):

- Wintner 1935: ν_λ is either absolutely continuous or singular w.r.t. Lebesgue. For $\lambda \in (1/2, 1)$ is it singular, and for $\lambda = 1/2$ it is a.c.
- Erdős 1939: $\exists \infty$ of singular $\lambda \in (0, 1/2)$.

Generalized Bernoulli convolutions

$$Z \overset{D}{=} (1-\lambda)Z + \lambda X$$

We say that Z is the *core* of p. It satisfies $Z = \sum_{k=0}^{+\infty} \lambda(1-\lambda)^k X_k$ with (X_k) i.i.d. sequence from p. We need to understand the "smoothness" of Z (which will translate in smoothness of the corresponding kernel).

Consider the core ν_λ of a random sign (this is a distinguished object introduced in the 1930's known as a Bernoulli convolution):

- Wintner 1935: ν_λ is either absolutely continuous or singular w.r.t. Lebesgue. For $\lambda \in (1/2, 1)$ is it singular, and for $\lambda = 1/2$ it is a.c.
- Erdős 1939: $\exists\infty$ of singular $\lambda \in (0, 1/2)$.
- Erdős 1940, Solomyak 1996: a.e. $\lambda \in (0, 1/2)$ is a.c.

Generalized Bernoulli convolutions

$$Z \overset{D}{=} (1-\lambda)Z + \lambda X$$

We say that Z is the *core* of p. It satisfies $Z = \sum_{k=0}^{+\infty} \lambda(1-\lambda)^k X_k$ with (X_k) i.i.d. sequence from p. We need to understand the "smoothness" of Z (which will translate in smoothness of the corresponding kernel).

Consider the core ν_λ of a random sign (this is a distinguished object introduced in the 1930's known as a Bernoulli convolution):

- Wintner 1935: ν_λ is either absolutely continuous or singular w.r.t. Lebesgue. For $\lambda \in (1/2, 1)$ is it singular, and for $\lambda = 1/2$ it is a.c.
- Erdős 1939: $\exists \infty$ of singular $\lambda \in (0, 1/2)$.
- Erdős 1940, Solomyak 1996: a.e. $\lambda \in (0, 1/2)$ is a.c.
- For any $k \in \mathbb{N}$, $\exists \lambda_k \approx 1/k$ s.t. ν_{λ_k} has a C^k density.

What is left to do?

Summarizing the discussion so far, let us play from $K_t p_t$, where K_t is the kernel described above (i.e., it "mixes in" the core of p_t) and p_t is the continuous exponential weights strategy on the estimated losses $\widetilde{\ell}_s = \ell_s(x_s)\frac{K_s(x_s,\cdot)}{K_s p_s(x_s)}$ (that is $dp_t(x)/dx$ is proportional to $\exp(-\eta \sum_{s<t} \widetilde{\ell}_s(x)))$.

What is left to do?

Summarizing the discussion so far, let us play from $K_t p_t$, where K_t is the kernel described above (i.e., it "mixes in" the core of p_t) and p_t is the continuous exponential weights strategy on the estimated losses $\widetilde{\ell}_s = \ell_s(x_s) \frac{K_s(x_s, \cdot)}{K_s p_s(x_s)}$ (that is $dp_t(x)/dx$ is proportional to $\exp(-\eta \sum_{s<t} \widetilde{\ell}_s(x))$).
Using the classical analysis of continuous exponential weights together with the previous slides we get for any q,

$$\begin{aligned}
\mathbb{E} \sum_{t=1}^{T} \langle K_t p_t - q, \ell_t \rangle &\leq \frac{1}{\lambda} \mathbb{E} \sum_{t=1}^{T} \langle K_t(p_t - q), \ell_t \rangle \\
&= \frac{1}{\lambda} \mathbb{E} \sum_{t=1}^{T} (\langle p_t - q, \widetilde{\ell}_t \rangle) \\
&\leq \frac{1}{\lambda} \mathbb{E} \left(\frac{\mathrm{Ent}(q \| p_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \langle p_t, \left(\frac{K_t(x_t, \cdot)}{K_t p_t(x_t)} \right)^2 \rangle \right) .
\end{aligned}$$

Variance calculation

All that remains to be done is to control the variance term $\mathbb{E}_{x \sim Kp} \langle p, \widetilde{\ell}^2 \rangle$ where $\widetilde{\ell}(y) = \frac{K(x,y)}{Kp(x)} = \frac{K(x,y)}{\int K(x,y')p(y')dy}$. More precisely if this quantity is $O(1)$ then we obtain a regret of $\widetilde{O}\left(\frac{1}{\lambda}\sqrt{nT}\right)$.

Variance calculation

All that remains to be done is to control the variance term $\mathbb{E}_{x \sim Kp} \langle p, \widetilde{\ell}^2 \rangle$ where $\widetilde{\ell}(y) = \frac{K(x,y)}{Kp(x)} = \frac{K(x,y)}{\int K(x,y')p(y')dy}$. More precisely if this quantity is $O(1)$ then we obtain a regret of $\widetilde{O}\left(\frac{1}{\lambda}\sqrt{nT}\right)$.

It is sufficient to control from above $K(x,y)/K(x,y')$ for all y, y' in the support of p and all x in the support of Kp (in fact it is sufficient to have it with probability at least $1 - 1/T^{10}$ w.r.t. $x \sim Kp$).

Variance calculation

All that remains to be done is to control the variance term $\mathbb{E}_{x\sim Kp}\langle p, \widetilde{\ell}^2\rangle$ where $\widetilde{\ell}(y) = \frac{K(x,y)}{Kp(x)} = \frac{K(x,y)}{\int K(x,y')p(y')dy}$. More precisely if this quantity is $O(1)$ then we obtain a regret of $\widetilde{O}\left(\frac{1}{\lambda}\sqrt{nT}\right)$.

It is sufficient to control from above $K(x,y)/K(x,y')$ for all y, y' in the support of p and all x in the support of Kp (in fact it is sufficient to have it with probability at least $1 - 1/T^{10}$ w.r.t. $x \sim Kp$).
Observe also that, with c denoting the core of p, one always has $K(x,y) = K\delta_y(x) = \mathrm{cst} \times c\left(\frac{x-\lambda y}{1-\lambda}\right)$. Thus we want to bound w.h.p w.r.t. $x \sim Kp$,

$$\sup_{y,y'\in\mathrm{supp}(p)} c\left(\frac{x-\lambda y}{1-\lambda}\right)/c\left(\frac{x-\lambda y'}{1-\lambda}\right).$$

Variance calculation heuristic

Control w.h.p w.r.t. $x \sim Kp$,

$$\sup_{y, y' \in \mathrm{supp}(p)} c\left(\frac{x - \lambda y}{1 - \lambda}\right) / c\left(\frac{x - \lambda y'}{1 - \lambda}\right) .$$

Variance calculation heuristic

Control w.h.p w.r.t. $x \sim Kp$,

$$\sup_{y, y' \in \mathrm{supp}(p)} c\left(\frac{x - \lambda y}{1-\lambda}\right) / c\left(\frac{x - \lambda y'}{1-\lambda}\right).$$

Let us assume

1. $p = \mathcal{N}(0, \mathrm{I}_n)$ (its core is $c = \mathcal{N}(0, \frac{\lambda}{2-\lambda} I_n)$).

Variance calculation heuristic

Control w.h.p w.r.t. $x \sim Kp$,

$$\sup_{y,y' \in \mathrm{supp}(p)} c\left(\frac{x-\lambda y}{1-\lambda}\right) / c\left(\frac{x-\lambda y'}{1-\lambda}\right) .$$

Let us assume

1. $p = \mathcal{N}(0, \mathrm{I}_n)$ (its core is $c = \mathcal{N}(0, \frac{\lambda}{2-\lambda} I_n)$).
2. $\mathrm{supp}(p) \subset \{y : |y| \leq R = \widetilde{O}(\sqrt{n})\}$

Variance calculation heuristic

Control w.h.p w.r.t. $x \sim Kp$,

$$\sup_{y, y' \in \mathrm{supp}(p)} c\left(\frac{x - \lambda y}{1 - \lambda}\right) / c\left(\frac{x - \lambda y'}{1 - \lambda}\right) .$$

Let us assume

1. $p = \mathcal{N}(0, \mathrm{I}_n)$ (its core is $c = \mathcal{N}(0, \frac{\lambda}{2-\lambda} I_n)$).
2. $\mathrm{supp}(p) \subset \{y : |y| \leq R = \widetilde{O}(\sqrt{n})\}$

Thus our quantity of interest is

$$\exp\left(\frac{2-\lambda}{2\lambda}\left(\left|\frac{x - \lambda y'}{1-\lambda}\right|^2 - \left|\frac{x - \lambda y}{1-\lambda}\right|^2\right)\right)$$
$$\leq \exp\left(\frac{1}{(1-\lambda)^2}(4R|x| + 2\lambda R^2)\right) .$$

Finally note that w.h.p. one has $|x| \lesssim \lambda R + \sqrt{\lambda n \log(T)}$, and thus with $\lambda = \widetilde{O}(1/n^2)$ we have a constant variance.

A reduction to the Gaussian case

We reduce to the Gaussian situation by observing that taking Z (in the definition of the kernel) to be the core of a measure convexly dominated by p is sufficient (instead of taking it to be directly the core of p), and furthermore one has:

A reduction to the Gaussian case

We reduce to the Gaussian situation by observing that taking Z (in the definition of the kernel) to be the core of a measure convexly dominated by p is sufficient (instead of taking it to be directly the core of p), and furthermore one has:

Lemma

Any isotropic log-concave measure p approximately convexly dominates a centered Gaussian with covariance $\widetilde{O}(\frac{1}{n})\mathrm{I}_n$.

A reduction to the Gaussian case

We reduce to the Gaussian situation by observing that taking Z (in the definition of the kernel) to be the core of a measure convexly dominated by p is sufficient (instead of taking it to be directly the core of p), and furthermore one has:

Lemma

Any isotropic log-concave measure p approximately convexly dominates a centered Gaussian with covariance $\widetilde{O}(\frac{1}{n})\mathrm{I}_n$.

Proof.

We show that p dominates any q supported on a small ball of cst radius. Pick a test function f, w.l.o.g. its minimum is 0 at 0 and the maximum on the ball is 1. By convexity f is above a linear function (maxed with 0) of constant slope. By light tails of log-concave, $\langle p, f \rangle$ is then at least a constant. □

A reduction to the Gaussian case

We reduce to the Gaussian situation by observing that taking Z (in the definition of the kernel) to be the core of a measure convexly dominated by p is sufficient (instead of taking it to be directly the core of p), and furthermore one has:

Lemma

Any isotropic log-concave measure p approximately convexly dominates a centered Gaussian with covariance $\widetilde{O}(\frac{1}{n})\mathrm{I}_n$.

Proof.

We show that p dominates any q supported on a small ball of cst radius. Pick a test function f, w.l.o.g. its minimum is 0 at 0 and the maximum on the ball is 1. By convexity f is above a linear function (maxed with 0) of constant slope. By light tails of log-concave, $\langle p, f \rangle$ is then at least a constant. □

What about assumption 2?

Restart and increasing learning rate

Unfortunately assumption 2 brings out a serious difficulty: it forces the algorithm to focus on smaller and smaller region of space. What if the adversary makes us focus on a region only to move the optimum far outside of it at a later time?

Restart and increasing learning rate

Unfortunately assumption 2 brings out a serious difficulty: it forces the algorithm to focus on smaller and smaller region of space. What if the adversary makes us focus on a region only to move the optimum far outside of it at a later time?

Idea: if the estimated optimum is too close to the boundary of the focus region then we restart the algorithm (similar idea appeared in Hazan and Li 2016).

Restart and increasing learning rate

Unfortunately assumption 2 brings out a serious difficulty: it forces the algorithm to focus on smaller and smaller region of space. What if the adversary makes us focus on a region only to move the optimum far outside of it at a later time?

Idea: if the estimated optimum is too close to the boundary of the focus region then we restart the algorithm (similar idea appeared in Hazan and Li 2016).

To be proved: negative regret at restart times (indeed the adversary must "pay" for making us focus and then move out the optimum). Technically this negative regret can come from a large relative entropy at some previous time.

Restart and increasing learning rate

Unfortunately assumption 2 brings out a serious difficulty: it forces the algorithm to focus on smaller and smaller region of space. What if the adversary makes us focus on a region only to move the optimum far outside of it at a later time?

Idea: if the estimated optimum is too close to the boundary of the focus region then we restart the algorithm (similar idea appeared in Hazan and Li 2016).

To be proved: negative regret at restart times (indeed the adversary must "pay" for making us focus and then move out the optimum). Technically this negative regret can come from a large relative entropy at some previous time.

Challenge: avoid the telescopic sum of entropies. For this we use a last idea: every time the focus region changes scale we also increase the learning rate.

Summary of the algorithm

- Compute the Gaussian N_t "inside" p_t, its associated core N'_t (when N_t is isotropic: $N'_t = \sqrt{\frac{\lambda}{2-\lambda}} N_t$), and the corresponding kernel: $K_t \delta_y = (1-\lambda) N'_t + \lambda y$ (i.e. $K_t(x, y) = N'_t(\frac{x-\lambda y}{1-\lambda}) \propto \exp(-\frac{n}{\lambda} \|x - \lambda y\|^2_{p_t})$).

Summary of the algorithm

- Compute the Gaussian N_t "inside" p_t, its associated core N'_t (when N_t is isotropic: $N'_t = \sqrt{\frac{\lambda}{2-\lambda}} N_t$), and the corresponding kernel: $K_t \delta_y = (1-\lambda) N'_t + \lambda y$ (i.e. $K_t(x,y) = N'_t(\frac{x-\lambda y}{1-\lambda}) \propto \exp(-\frac{n}{\lambda} \|x - \lambda y\|_{p_t}^2)$).
- Sample $X_t \sim p_t$ and play $x_t = (1-\lambda) N'_t + \lambda X_t \sim K_t p_t$.

Summary of the algorithm

- Compute the Gaussian N_t "inside" p_t, its associated core N_t' (when N_t is isotropic: $N_t' = \sqrt{\frac{\lambda}{2-\lambda}} N_t$), and the corresponding kernel: $K_t \delta_y = (1-\lambda) N_t' + \lambda y$ (i.e. $K_t(x,y) = N_t'(\frac{x-\lambda y}{1-\lambda}) \propto \exp(-\frac{n}{\lambda} \|x - \lambda y\|_{p_t}^2)$).
- Sample $X_t \sim p_t$ and play $x_t = (1-\lambda) N_t' + \lambda X_t \sim K_t p_t$.
- Update the exponential weights distribution: $p_{t+1}(y) \propto p_t(y) \exp(-\eta_t \widetilde{\ell}_t(y))$

Summary of the algorithm

- Compute the Gaussian N_t "inside" p_t, its associated core N'_t (when N_t is isotropic: $N'_t = \sqrt{\frac{\lambda}{2-\lambda}N_t}$), and the corresponding kernel: $K_t\delta_y = (1-\lambda)N'_t + \lambda y$ (i.e. $K_t(x,y) = N'_t(\frac{x-\lambda y}{1-\lambda}) \propto \exp(-\frac{n}{\lambda}\|x-\lambda y\|^2_{p_t})$).
- Sample $X_t \sim p_t$ and play $x_t = (1-\lambda)N'_t + \lambda X_t \sim K_t p_t$.
- Update the exponential weights distribution: $p_{t+1}(y) \propto p_t(y)\exp(-\eta_t \widetilde{\ell}_t(y))$ where

$$\widetilde{\ell}_t(y) = \frac{\ell_t(x_t)}{K_t p_t(x_t)} K_t(x_t, y) \propto \exp(-n\lambda\|y - x_t/\lambda\|^2_{p_t})$$

Summary of the algorithm

- Compute the Gaussian N_t "inside" p_t, its associated core N'_t (when N_t is isotropic: $N'_t = \sqrt{\frac{\lambda}{2-\lambda}} N_t$), and the corresponding kernel: $K_t \delta_y = (1-\lambda) N'_t + \lambda y$ (i.e. $K_t(x, y) = N'_t(\frac{x-\lambda y}{1-\lambda}) \propto \exp(-\frac{n}{\lambda} \|x - \lambda y\|^2_{p_t})$).
- Sample $X_t \sim p_t$ and play $x_t = (1-\lambda) N'_t + \lambda X_t \sim K_t p_t$.
- Update the exponential weights distribution: $p_{t+1}(y) \propto p_t(y) \exp(-\eta_t \widetilde{\ell}_t(y))$ where

$$\widetilde{\ell}_t(y) = \frac{\ell_t(x_t)}{K_t p_t(x_t)} K_t(x_t, y) \propto \exp(-n\lambda \|y - x_t/\lambda\|^2_{p_t})$$

(note that $\|x_t/\lambda\| \approx 1/\sqrt{\lambda}$ and the standard deviation of the above Gaussian is $\approx 1/\sqrt{n\lambda}$).

Summary of the algorithm

- Compute the Gaussian N_t "inside" p_t, its associated core N_t' (when N_t is isotropic: $N_t' = \sqrt{\frac{\lambda}{2-\lambda} N_t}$), and the corresponding kernel: $K_t \delta_y = (1-\lambda) N_t' + \lambda y$ (i.e. $K_t(x, y) = N_t'(\frac{x - \lambda y}{1-\lambda}) \propto \exp(-\frac{n}{\lambda} \|x - \lambda y\|_{p_t}^2)$).
- Sample $X_t \sim p_t$ and play $x_t = (1-\lambda) N_t' + \lambda X_t \sim K_t p_t$.
- Update the exponential weights distribution: $p_{t+1}(y) \propto p_t(y) \exp(-\eta_t \widetilde{\ell}_t(y))$ where

$$\widetilde{\ell}_t(y) = \frac{\ell_t(x_t)}{K_t p_t(x_t)} K_t(x_t, y) \propto \exp(-n\lambda \|y - x_t/\lambda\|_{p_t}^2)$$

  (note that $\|x_t/\lambda\| \approx 1/\sqrt{\lambda}$ and the standard deviation of the above Gaussian is $\approx 1/\sqrt{n\lambda}$).
- Restart business: check if adversary is potentially moving out of focus region (if so restart the algorithm), check if updating the focus region would change the problem's scale (if so make the update and increase the learning rate multiplicatively by $(1 + \frac{1}{\widetilde{O}(\mathrm{poly}(n))})$).