Projected Stochastic Primal-Dual Method for Constrained Online Learning with Kernels

Kaiqing Zhang[§], Tame Başar[§], Hao Zhu[†],Alec Koppel[⋆]
[§]University of Illinois Urbana-Champaign [†]UT Austin
[⋆]CISD, U.S. Army Research Laboratory

Large-Scale Distributed Optimization and Computation on Graphs
IEEE Conference on Decision and Control
Miami, FL, Dec. 18, 2018

Constrained Nonlinear Interpolation

- Want to find $f^* \in \mathcal{H}$ to minimize some expected cost $R(f)$

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(f(\mathbf{x}), y)] + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}$$
$$\text{such that } \mathbf{G}(f) \leq \mathbf{0}$$

  $\Rightarrow$ Loss $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ penalize deviations between $f(\mathbf{x})$, $\mathbf{y}$

  $\Rightarrow$ interpret $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ as features/state variables

  $\Rightarrow$ $y \in \mathcal{Y}$ $\Rightarrow$ targets, e.g., reference trajectory or binary labels

  $\Rightarrow$ expected risk $L(f) := \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(f(\mathbf{x}), \mathbf{y})]$
- Examples:

  $\Rightarrow$ learning with risk constraints

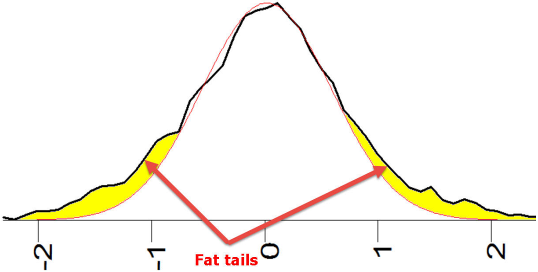
  $\Rightarrow$ trajectory planning with obstacle avoidance

  $\Rightarrow$ wireless utility maximization with quality of service guarantees

  $\Rightarrow$ wireless beamforming with robustness constraints

Examples

- Learning with nonlinear statistical models and risk constraints
  - $\Rightarrow$ when distribution $\mathbb{P}(\mathbf{x}, \mathbf{y})$ has heavy tails
  - $\Rightarrow$ then learning $f(\mathbf{x})$ by minimizing **average** loss will "overfit"



- Impose risk constraint, such as CVaR (Rockafeller '2000)

$$
\begin{aligned}
G(f) &= \mathrm{CVaR}_\alpha(f) - \gamma \\
&= \min_{z \in \mathbb{R}} \left\{ z + \frac{1}{1-\alpha} \mathbb{E}_{\mathbf{x},\mathbf{y}} \{ [\ell(f(\mathbf{x}), \mathbf{y}) - z]_+ \} \right\} - \gamma
\end{aligned}
$$

Examples

- Trajectory planning with obstacle avoidance
  $\Rightarrow$ obstacles define region of state space to be avoided

- Let $g(f(x)) > 0$ represent safe area in $\mathbb{R}^p$
  $\Rightarrow$ upper bound $\mathbb{P}(g(f(x)) > 0) \leq \gamma$ for a given γ
- Convexified chance constraint:

$$\inf_{\lambda > 0} \left[\Psi(f, \lambda) - \lambda\gamma\right] \leq 0,$$

here $\Psi(f, \lambda) = \lambda \mathbb{E}_x[\phi(\lambda^{-1} g(f(x)))]$ $\Rightarrow$ $\phi(\cdot)$: MGF of $\mathbb{P}(x)$

Some Context

- Want to find $f^* \in \mathcal{H}$ to minimize some expected cost $R(f)$

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(f(\mathbf{x}), y)] + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}$$
$$\text{such that} \;\; \mathbf{G}(f) \leq \mathbf{0}$$

- Classically address using calculus of variations (Hamilton 1800s)
  - $\Rightarrow$ w/o special structure, can't solve Euler-Lagrange equations
  - $\Rightarrow$ with special structure on distribution $\Rightarrow$ variational inference
- Without hypotheses on distribution, "learning" approaches
  - $\Rightarrow$ parameterize f, then estimate parameters via samples of $\mathbf{x}$, y
- Choose parameterization s.t. solution close to original problem
  - $\Rightarrow$ "universal parameterizations:" Bayesian/**nonparametric**/DNN

Our Contribution

- Want to find $f^* \in \mathcal{H}$ to minimize some expected cost $R(f)$

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(f(\mathbf{x}), y)] + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}$$

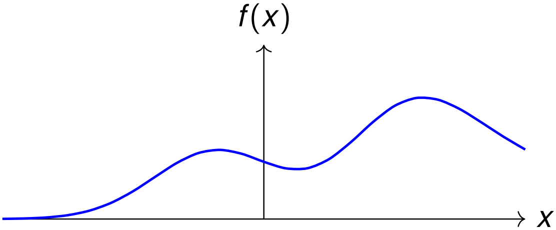
$$\text{such that } \mathbf{G}(f) \leq \mathbf{0}$$

- We adopt a **nonparametric** parameterization of f
  - $\Rightarrow$ using a reproducing kernel Hilbert space (RKHS)
  - $\Rightarrow$ motivated by the fact this param. **preserves convexity**
  - $\Rightarrow$ therefore Lagrange duality applies
- Extend Representer Theorem to constrained settings
  - $\Rightarrow$ for certain constraints, using augmented Lagrangian
- Propose a projected stochastic primal-dual method
  - $\Rightarrow$ custom projection trades off convergence and complexity
  - $\Rightarrow$ generalize existing convergence rates from vector case

Reproducing Kernel Hilbert Space

- Equip $\mathcal{H}$ with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that:

$$(i)\ \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X}\ ,$$
$$(ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}\ .$$



- Property (i) $\Rightarrow$ Will allow us to compute derivatives
- Kernel examples:

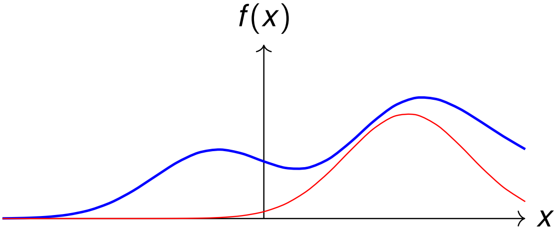
$\Rightarrow$ Gaussian/RBF $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2c^2}\right\}$

$\Rightarrow$ polynomial $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$

Reproducing Kernel Hilbert Space

- Equip $\mathcal{H}$ with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that:

$$(i)\ \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X}\ ,$$
$$(ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}\ .$$



- Property (i) $\Rightarrow$ Will allow us to compute derivatives
- Kernel examples:

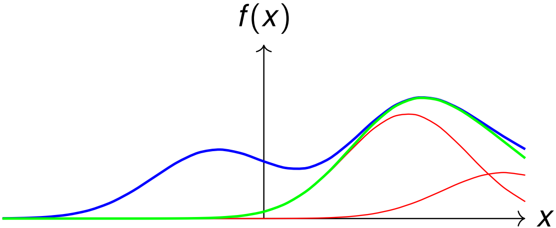
$\Rightarrow$ Gaussian/RBF $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2c^2}\right\}$

$\Rightarrow$ polynomial $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$

Reproducing Kernel Hilbert Space

- Equip $\mathcal{H}$ with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that:

$$(i)\ \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X}\ ,$$
$$(ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}\ .$$



- Property (i) $\Rightarrow$ Will allow us to compute derivatives
- Kernel examples:

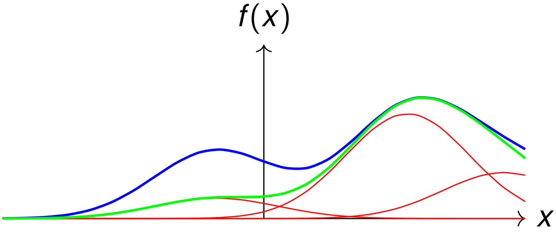
  $\Rightarrow$ Gaussian/RBF $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2c^2}\right\}$

  $\Rightarrow$ polynomial $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$

Reproducing Kernel Hilbert Space

- Equip $\mathcal{H}$ with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that:

$$(i)\ \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X}\ ,$$
$$(ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}\ .$$



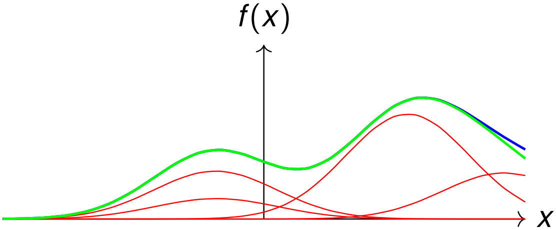
- Property (i) $\Rightarrow$ Will allow us to compute derivatives
- Kernel examples:

  $\Rightarrow$ Gaussian/RBF $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2c^2}\right\}$

  $\Rightarrow$ polynomial $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$

- Equip $\mathcal{H}$ with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that:

$$(i)\ \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X}\ ,$$
$$(ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}\ .$$



- Property (i) $\Rightarrow$ Will allow us to compute derivatives
- Kernel examples:

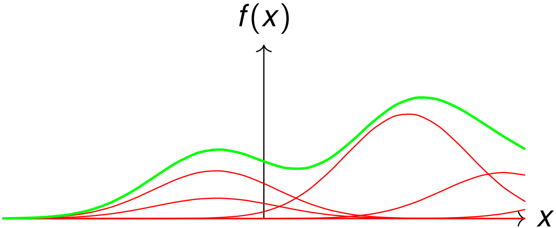
$\Rightarrow$ Gaussian/RBF $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2c^2}\right\}$

$\Rightarrow$ polynomial $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$

Reproducing Kernel Hilbert Space

- Equip $\mathcal{H}$ with a unique *kernel function*, $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that:

$$(i)\ \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X}\ ,$$
$$(ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}} \quad \text{for all } \mathbf{x} \in \mathcal{X}\ .$$



- Property (i) $\Rightarrow$ Will allow us to compute derivatives
- Kernel examples:

$\Rightarrow$ Gaussian/RBF $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2c^2}\right\}$

$\Rightarrow$ polynomial $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$

Function Representation

- Consider empirical risk minimization case: sample size $N < \infty$
- Classic Representer Theorem:

$$f^* = \underset{f}{\text{argmin}} \frac{1}{N} \sum_{n=1}^{N} \ell(f(\mathbf{x}_n), \mathbf{y}_n) \text{ takes the form } f(\mathbf{x}) = \sum_{n=1}^{N} w_n \, \kappa(\mathbf{x}_n, \mathbf{x}) \; .$$

$\Rightarrow$ $\mathbf{x}_n$ are feature vectors, and w_n is a scalar weight.

$\Rightarrow$ f is a kernel expansion over training set

$\Rightarrow$ dates to Riesz & Weiner, to ML by Scholkopf/Smola

- Does not apply to constrained settings . . .

Function Representation with Constraints

- Consider Lagrangian of constrained problem

$$\mathcal{L}^o(f, \boldsymbol{\mu}) = L(f) + \boldsymbol{\mu}^\top \mathbf{G}(f) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}},$$

Theorem

Suppose constraint takes form $\mathbf{G}(f) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{g}(f(\mathbf{x}), \mathbf{y})]$
Define saddle pt. prob: $(\check{f}^*, \check{\boldsymbol{\mu}}^*) = \arg\max_{\boldsymbol{\mu}\in\mathbb{R}^m_+} \min_{f\in\mathcal{H}} \quad \mathcal{L}^o(f, \boldsymbol{\mu}; \mathcal{S}),$
$\Rightarrow$ *Consider sample avg. approx. of Lagrangian w/* $\mathcal{S}_N = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$*:*

$$\mathcal{L}^o(f, \boldsymbol{\mu}; \mathcal{S}) := \frac{1}{N}\sum_{n=1}^{N}\Big[\ell(f(\mathbf{x}_n), \mathbf{y}_n) + \sum_{j=1}^{m}\mu_j g_j(f(\mathbf{x}_n), \mathbf{y}_n)\Big] + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}.$$

Then primal minimizer of takes form $f(\mathbf{x}) = \sum_{n=1}^N w_n \, \kappa(\mathbf{x}_n, \mathbf{x})$ *.*

Functional Stochastic Primal-Dual Method

- Formulate augmented Lagrangian of constrained prob:

$$\mathcal{L}(f, \boldsymbol{\mu}) = L(f) + \boldsymbol{\mu}^\top \mathbf{G}(f) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 - \frac{\delta\eta}{2}\|\boldsymbol{\mu}\|^2.$$

  $\Rightarrow$ $\boldsymbol{\mu}$ is Lagrange multiplier, δ is a regularization parameter

  $\Rightarrow$ Define stoch. approx. based on sample $(\mathbf{x}_t, \mathbf{y}_t)$ as $\hat{\mathcal{L}}_t(f, \boldsymbol{\mu})$

- Set $\tilde{\ell}(f(\mathbf{x}), \mathbf{y}, \boldsymbol{\mu}) = \ell(f(\mathbf{x}), \mathbf{y}) + \sum_{j=1}^m \mu_j g_j(f(\mathbf{x}), \mathbf{y})$.

  $\Rightarrow$ Then let's compute primal stochastic gradient:

$$\begin{aligned}\nabla_f \tilde{\ell}(f(\mathbf{x}_t), \mathbf{y}_t, \boldsymbol{\mu})(\cdot) &= \frac{\partial \tilde{\ell}(f(\mathbf{x}_t), \mathbf{y}_t, \boldsymbol{\mu})}{\partial f(\mathbf{x}_t)} \frac{\partial f(\mathbf{x}_t)}{\partial f}(\cdot) \\ &= \tilde{\ell}'(f(\mathbf{x}_t), \mathbf{y}_t, \boldsymbol{\mu})\kappa(\mathbf{x}_t, \cdot)\end{aligned}$$

Functional Stochastic Primal-Dual Method

- Stochastic primal/dual descent/ascent steps:

$$\begin{cases} f_{t+1} = (1-\eta\lambda)f_t - \eta\Big[\ell'(f_t(\mathbf{x}_t),\mathbf{y}_t) + \sum_{j=1}^{m}\mu_j g_j'(f_t(\mathbf{x}_t),\mathbf{y}_t)\Big]\kappa(\mathbf{x}_t,\cdot), \\ \boldsymbol{\mu}_{t+1} = \big[(1-\eta^2\delta)\boldsymbol{\mu}_t + \eta\mathbf{g}(f_t(\mathbf{x}_t),\mathbf{y}_t)\big]_+, \end{cases}$$

- Via induction, can show $f_t(\mathbf{x}) = \sum_{t=1}^{t-1} w_t\kappa(\mathbf{x}_t,\mathbf{x}) = \mathbf{w}_t^\top \boldsymbol{\kappa}_{\mathbf{X}_t}(\mathbf{x})$

  $\Rightarrow$ hence f_t is parameterized by a growing matrix, weight vec.:

$$\mathbf{X}_{t+1} = [\mathbf{X}_t, \ \ \mathbf{x}_t],$$

$$\mathbf{w}_{t+1} = \Big[(1-\eta\lambda)\mathbf{w}_t, \ \ -\eta\ell'(f_t(\mathbf{x}_t),\mathbf{y}_t) - \eta\sum_{j=1}^{m}\mu_j g_j'(f_t(\mathbf{x}_t),\mathbf{y}_t)\Big] \ .$$

Online Subspace Projections

- Define un-projected/unsparsified iterate at step $t+1$

$$\tilde{f}_{t+1} = (1-\eta_t\lambda)f_t - \eta_t \nabla_f \tilde{\ell}(f_t; \mathbf{x}_t, \mathbf{y}_t; \boldsymbol{\mu}_t).$$

$\Rightarrow$ parameterized by dictionary and coefficients

$$\tilde{\mathbf{D}}_{t+1} = [\mathbf{D}_t, \;\; \mathbf{x}_t], \qquad \tilde{\mathbf{w}}_{t+1} = [(1-\eta_t\lambda)\mathbf{w}_t, \;\; -\tilde{\ell}'(f_t; \mathbf{x}_t, \mathbf{y}_t; \boldsymbol{\mu}_t)] \; .$$

- Our method: $(f_{t+1}, \mathbf{D}_{t+1}, \mathbf{w}_{t+1}) = \textbf{KOMP}(\tilde{f}_{t+1}, \tilde{\mathbf{D}}_{t+1}, \tilde{\mathbf{w}}_{t+1}, \epsilon_t)$
- This amounts to a certain orthogonal subspace projection

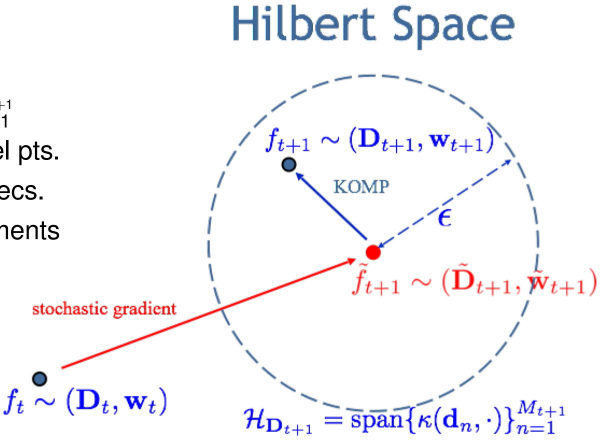
$$\begin{aligned} f_{t+1} &= \underset{f \in \mathcal{H}_{\mathbf{D}_{t+1}}}{\operatorname{argmin}} \left\| f - \Big((1-\eta_t\lambda)f_t - \eta_t \nabla_f \tilde{\ell}(f_t; \mathbf{x}_t, \mathbf{y}_t; \boldsymbol{\mu}_t) \Big) \right\|_{\mathcal{H}}^2 \\ &:= \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}} \Big[(1-\eta_t\lambda)f_t - \eta_t \nabla_f \tilde{\ell}(f_t(\mathbf{x}_t), y_t; \boldsymbol{\mu}_t) \Big]. \end{aligned}$$

- where we define Hilbert subspace $\mathcal{H}_{\mathbf{D}_{t+1}} = \text{span}\{\kappa(\mathbf{d}_n, \cdot)\}_{n=1}^{M_{t+1}}$

$\Rightarrow$ $\mathbf{d}_n$ are model points $\Rightarrow$ subset of past feature vectors $\{\mathbf{x}_u\}_{u \le t}$

Kernel Matching Pursuit

- Fix approx. error ϵ_t
- Define subspace $\mathcal{H}_{\mathbf{D}_{t+1}} = \text{span}\{\kappa(\mathbf{d}_n, \cdot)\}_{n=1}^{M_{t+1}}$
- $\{\mathbf{d}_n\} \subset \{\mathbf{x}_u\}_{u \leq t} \Rightarrow$ model pts.
  $\Rightarrow$ subset of past feat. vecs.
- Remove kernel dict. elements
- Stopping criterion: $\|\tilde{f}_{t+1} - f_{t+1}\|_{\mathcal{H}} \leq \epsilon_t$
- New model order: $M_{t+1} \leq M_t + 1$



Technical Setting

- The feature space $\mathcal{X} \subset \mathbb{R}^p$, target domain $\mathcal{Y} \subset \mathbb{R}$ are compact, and kernel is bounded $\sup_{\mathbf{x} \in \mathcal{X}} \sqrt{\kappa(\mathbf{x}, \mathbf{x})} = X < \infty$
- Instantaneous loss $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is uniformly C_1-Lipschitz continuous, and the constraints $g_i : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is C_2-Lipschitz, for all $z \in \mathbb{R}$ for a fixed $y \in \mathcal{Y}$.
- The primal loss $\ell(f(\mathbf{x}), y)$ is convex and differentiable with respect to its scalar argument $f(\mathbf{x})$, as are the constraints $g_i(f(\mathbf{x}), y)$ on $\mathbb{R}$ for all $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- There exists a strictly feasible pt: some $f \in \mathcal{H}$ satisfies $\mathbf{G}(f) < \mathbf{0}$.
- The output f_{t+1} of the KOMP update has Hilbert norm bounded by $R_{\mathcal{B}} < \infty$, and the optimal f^* lies in the ball $\mathcal{B}$ with radius $R_{\mathcal{B}}$

Main Result

Theorem

Denote the projected stochastic primal-dual sequence as $(f_t, \boldsymbol{\mu}_t)$. After T iterations with a constant step-size selected as $\eta = 1/\sqrt{T}$ and the approximation budget $\epsilon_t = \epsilon = P\eta^2$, where $P > 0$ is a fixed constant, we have

$$\sum_{t=1}^{T} \mathbb{E}[R(f_t) - R(f^*)] = \mathcal{O}(\sqrt{T})$$

Moreover, the time aggregation of the expected constraint violation of the algorithm grows sublinearly in T as

$$\sum_{j=1}^{m} \mathbb{E}\Big[\sum_{t=1}^{T} G_j(f_t)\Big]_+ \leq \mathcal{O}(T^{3/4}).$$

Main Result

Corollary

For $\bar{f}_T = \sum_{t=1}^{T} f_t/T$ as the functional formed by averaging the primal iterates f_t over time $t = 1, \cdots, T$, its objective function satisfies

$$\mathbb{E}[R(\bar{f}_T) - R(f^*)] \leq \mathcal{O}(1/\sqrt{T}).$$

In addition, the constraint violation evaluated at $\bar{f}_T$ satisfies

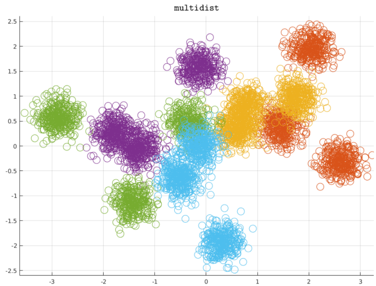
$$\sum_{j=1}^{m} \mathbb{E}\Big[\big(G_j(\bar{f}_T)\big)\Big]_+ \leq \mathcal{O}(T^{-1/4}).$$

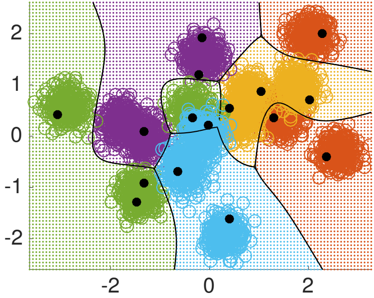
Discussion

- Comparable to existing results for vector-valued case
  - $\Rightarrow$ both in terms of primal sub-optimality and constraint violation
- Using a constant step-size and compression budget
  - $\Rightarrow$ yields fact that model complexity of RKHS function is finite
- Complexity depends on
  - $\Rightarrow$ parsimony constant P
  - $\Rightarrow$ kernel choice
  - $\Rightarrow$ data domain radius X

Online Multi-Class Kernel SVM

- Case where training examples for a fixed class
  $\Rightarrow$ drawn from a distinct Gaussian mixture
- 3 Gaussians per mixture, $C = 5$ classes total for this experiment
  $\Rightarrow$ 15 total Gaussians generate data

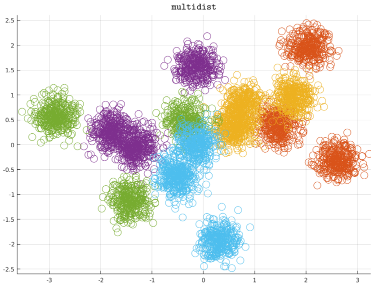


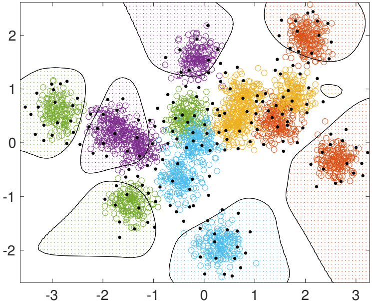


- Grid colors $\Rightarrow$ decision, bold black dots $\Rightarrow$ kernel dict. elements
- $\sim$ 96% accuracy

Online Multi-Class Kernel SVM

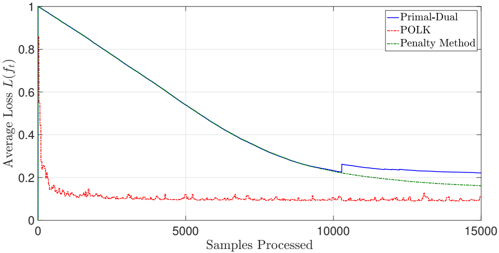
- Case where training examples for a fixed class
  $\Rightarrow$ drawn from a distinct Gaussian mixture
- 3 Gaussians per mixture, $C = 5$ classes total for this experiment
  $\Rightarrow$ 15 total Gaussians generate data





- Grid colors $\Rightarrow$ decision, bold black dots $\Rightarrow$ kernel dict. elements
- risk constraint prevents confidence in areas of class overlap

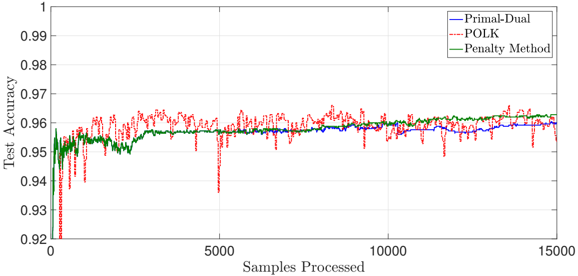
Online Multi-Class Kernel SVM



- Comparison with unconstrained (POLK) and penalty approach

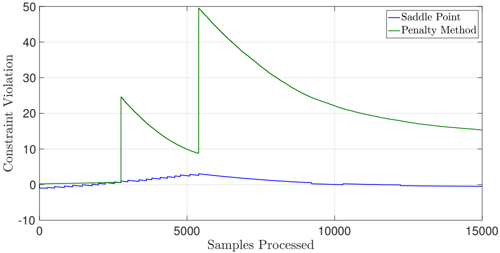
  $\Rightarrow$ Constrained optimizer has large objective than unconstrained

Online Multi-Class Kernel SVM



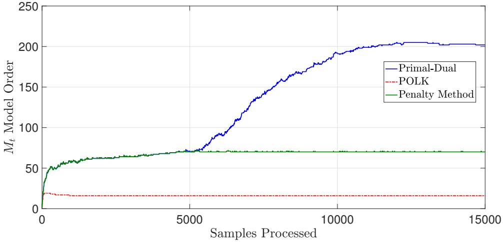
- Comparable in terms of test accuracy

Online Multi-Class Kernel SVM



- Primal-dual most effective for maintaining feasibility

Online Multi-Class Kernel SVM



- Constrained optimizer higher complexity than unconstrained
  $\Rightarrow$ control decision uncertainty $\Rightarrow$ higher order data moments
  $\Rightarrow$ define more complicated subspace than mean loss

Conclusion

- Focus on stochastic nonlinear interpolation with constraints
  - $\Rightarrow$ MPC with obstacle avoidance, risk-aware learning, etc.
  - $\Rightarrow$ parameterized with RKHS: "universal," preserves convexity
- Extended Representer Theorem to constrained settings
  - $\Rightarrow$ via use of empirical Lagrangian
- Proposed stochastic primal-dual method to solve it
  - $\Rightarrow$ operates in parallel with subspace projection scheme
- Generalized convergence results for vector-valued case
  - $\Rightarrow$ sub-optimality $\mathcal{O}(\sqrt{T})$; constraint violation as $\mathcal{O}(T^{3/4})$
- Online kernel multi-class SVM example
  - $\Rightarrow$ demonstrates effect of incorporating risk into decisions
  - $\Rightarrow$ confidence regions repelled by class overlap