Breaking the Curse of Dimensionality with Convex Neural Networks

Francis Bach

INRIA - Ecole Normale Supérieure, Paris, France

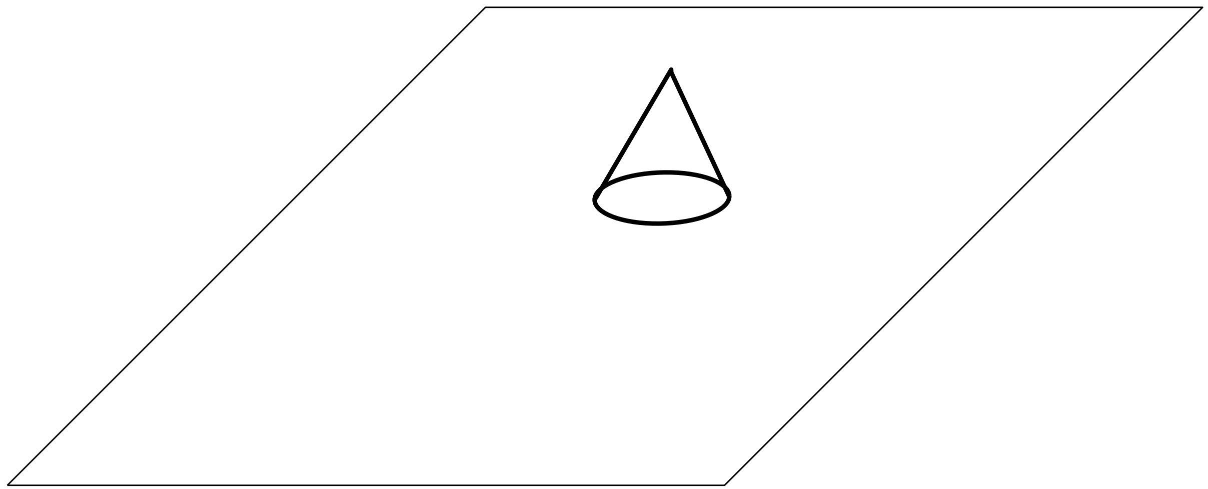
CIFAR meeting, Montréal - December 2014

Curse of dimensionality (supervised learning)

- **Goal: Learning a function $f : \mathbb{R}^d \to \mathbb{R}$ with minimal risk**

$$R(f) = \mathbb{E}\big[\ell(y, f(x))\big]$$

  - **Minimizer f^* only assumed to be Lipshitz-continuous**
  - **Need $n = \Omega(\varepsilon^{-d})$ observations to achieve $R(f) - R(f^*) \leqslant \varepsilon$**



Curse of dimensionality (supervised learning)

- **Goal: Learning a function $f : \mathbb{R}^d \to \mathbb{R}$ with minimal risk**

$$R(f) = \mathbb{E}\big[\ell(y, f(x))\big]$$

  - **Minimizer f^* only assumed to be Lipshitz-continuous**
  - **Need $n = \Omega(\varepsilon^{-d})$ observations to achieve $R(f) - R(f^*) \leqslant \varepsilon$**

- **Reducing sample complexity by exploiting structure**

Linear function	$w^\top x + b$	$d\varepsilon^{-2}$
Generalized additive model	$\sum_{j=1}^d f_j(x_j)$	$k^4 d^2 \varepsilon^{-4}$
One-hidden layer neural network	$\sum_{i=1}^k \eta_i \sigma(w_i^\top x + b)$	$k^2 d \varepsilon^{-2}$
Projection pursuit	$\sum_{i=1}^k f_i(w_i^\top x)$	$k^4 d^2 \varepsilon^{-4}$
Subspace dependence	$g(W^\top x)$	$(\frac{\varepsilon}{k\sqrt{d}})^{-\mathrm{rank(W)}+3}$

Goals

$$f(x) = \sum_{i=1}^{k} \eta_i \max\{w_i^\top x + b_i, 0\} = \sum_{i=1}^{k} \eta_i (w_i^\top x + b_i)_+$$

- **Generalization properties?**
  - **Adaptivity to structure**
  - **Non-linear variable selection**

- **Learning or sampling weights $(w_i, b_i) \in \mathbb{R}^{d+1}$?**
  - **Convexification by letting $k \to +\infty$**
  - **Selection (ℓ_1) vs. random sampling (ℓ_2)**

- **Hard or easy to optimize?**
  - **Polynomial time algorithms ...**
  - **... with same guarantees on unseen data**

Convex neural networks (Bengio, Le Roux, Vincent, Delalleau, and Marcotte, 2006)

Main idea

- **Replace the sum $\sum_{i=1}^{k} \eta_i (w_i^\top x + b_i)_+$ by an integral**

$$f(x) = \int_{\mathbb{R}^{d+1}} (w^\top x + b)_+ \, \eta(w,b) d\tau(w,b)$$

  - **Equivalence when $\eta d\tau$ is a weighted sum of Diracs: $\sum_{i=1}^{k} \eta_i \delta_{w_i,b_i}$**

- **Promote sparsity with an ℓ_1-norm: $\int_{\mathbb{R}^{d+1}} |\eta(w,b)| d\tau(w,b)$**

Convex neural networks
Formalization

- **Several points of views (Barron, 1993; Kurkova and Sanguineti, 2001; Bengio et al., 2006; Rosset et al., 2007)**

- **Define space $\mathcal{F}_1$ of functions f that can be decomposed as**

$$f(x) = \int_{\mathbb{R}^{d+1}} (w^\top x + b)_+ \, \eta(w,b) d\tau(w,b) \qquad (\bullet)$$

Convex neural networks

Formalization

- Several points of views (Barron, 1993; Kurkova and Sanguineti, 2001; Bengio et al., 2006; Rosset et al., 2007)

- Define space $\mathcal{F}_1$ of functions f that can be decomposed as

$$f(x) = \int_{\mathbb{R}^{d+1}} (w^\top x + b)_+ \, \eta(w,b) d\tau(w,b) \qquad (\bullet)$$

- Define the variation norm $\gamma_1(f)$ on $\mathcal{F}_1$ as

$$\gamma_1(f) = \inf \int_{\mathbb{R}^{d+1}} |\eta(w,b)| d\tau(w,b) \quad \text{such that } (\bullet) \text{ holds}$$

Variation norm and finite decomposition

- **Property 1 (Leshno et al., 1993): $\mathcal{F}_1$ is dense in L^2**

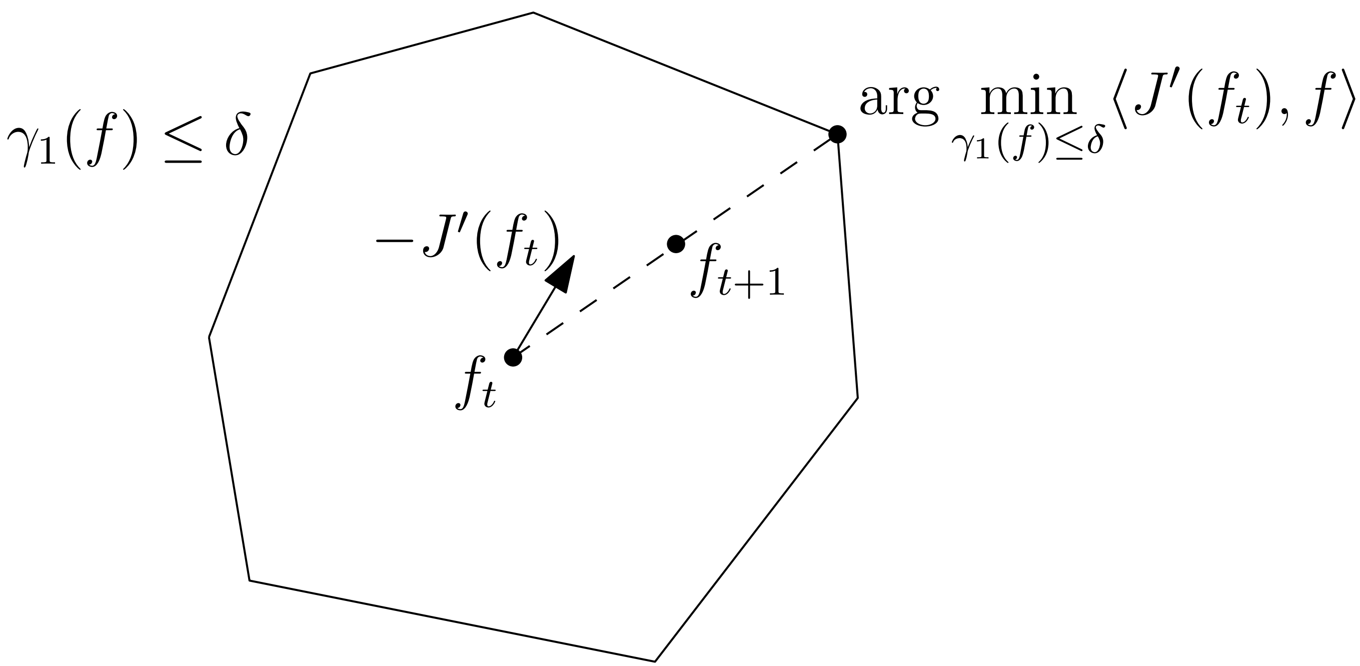
Variation norm and finite decomposition

- **Property 1 (Leshno et al., 1993):** $\mathcal{F}_1$ **is dense in** L^2
- **Property 2 (Barron, 1993): for any** $f \in \mathcal{F}_1$**, there exists a finite decomposition** $\hat{f}(x) = \sum_{i=1}^{k} \eta_i (w_i^\top x + b_i)_+$ **such that**
  - $\|f - \hat{f}\| \leqslant \varepsilon$ **in** L^2**-norm**
  - $k = O(\gamma_1(f)^2 \varepsilon^{-2})$
- **NB: constructive proof by conditional gradient algorithm**

Conditional gradient algorithm

- **Minimizing** $J(f)$ **such that** $\gamma_1(f) \leqslant \delta$
  - J **smooth and convex**
  - **Frank-Wolfe, conditional gradient, gradient boosting, etc. (Frank and Wolfe, 1956; Dem'yanov and Rubinov, 1967; Dudik et al., 2012; Harchaoui et al., 2013; Jaggi, 2013)**

- **Iteration:** $f_{t+1} = (1-\rho_t) f_t + \rho_t \operatorname*{argmin}_{\gamma_1(f)\leqslant\delta} \langle J'(f_t), f\rangle$



Conditional gradient algorithm

- Minimizing $J(f)$ such that $\gamma_1(f) \leqslant \delta$
  - J smooth and convex
  - Frank-Wolfe, conditional gradient, gradient boosting, etc. (Frank and Wolfe, 1956; Dem'yanov and Rubinov, 1967; Dudik et al., 2012; Harchaoui et al., 2013; Jaggi, 2013)

- Iteration: $f_{t+1} = (1-\rho_t) f_t + \rho_t \operatorname*{argmin}_{\gamma_1(f)\leqslant\delta} \langle J'(f_t), f \rangle$
  - Line search or $\rho_t = 2/(t+1)$
  - Convergence rate: $J(f) - \inf_{\gamma_1(g)\leqslant\delta} J(g) = O(\delta^2/t)$

- f_t = convex combination of t extreme points

Conditional gradient algorithm
Extreme points

- **Iteration:** $f_{t+1} = (1-\rho_t) f_t + \rho_t \operatorname*{argmin}_{\gamma_1(f) \leqslant \delta} \langle J'(f_t), f \rangle$
- f_t = **convex combination of** t **extreme points**
  - ℓ_1**-ball: extreme points are 1-sparse vectors**
  - **The set** $\{\gamma_1(f) \leqslant \delta\}$ **is the convex hull of all functions**

$$x \mapsto \pm \delta (w^\top x + b)_+, \ \textbf{for } (w, b) \in \mathbb{R}^{d+1}$$

Conditional gradient algorithm

Extreme points

- **Iteration:** $f_{t+1} = (1-\rho_t) f_t + \rho_t \operatorname*{argmin}_{\gamma_1(f) \leqslant \delta} \langle J'(f_t), f \rangle$

- f_t = **convex combination of t extreme points**
  - ℓ_1**-ball: extreme points are 1-sparse vectors**
  - **The set $\{\gamma_1(f) \leqslant \delta\}$ is the convex hull of all functions**

$$x \mapsto \pm\delta(w^\top x + b)_+, \ \textbf{for } (w, b) \in \mathbb{R}^{d+1}$$

- **Extreme points are single neurons/units**

$$\operatorname*{argmin}_{\gamma_1(f) \leqslant \delta} \langle J'(f_t), f \rangle = \pm\delta(w_t^\top \cdot + b_t)_+$$

  - **for** $(w_t, b_t) = -\operatorname{argmax}_{(w,b) \in \mathbb{R}^{d+1}} \left| \langle J'(f_t), (w^\top \cdot \ + b)_+ \rangle \right|$

Conditional gradient algorithm

Supervised learning from finite data set

- **Goal:** $\displaystyle \min_{\gamma_1(f)\leqslant\delta} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$

- **Adding a new unit/neuron/basis function:**

$$\operatorname*{argmax}_{(w,b)\in\mathbb{R}^{d+1}} \left| \frac{1}{n}\sum_{i=1}^{n} g_i \cdot (w^\top x_i + b)_+ \right| \quad \textbf{with } g_i = \ell'(y_i, f_t(x_i))$$

  – **Computational difficulty?**

Adding extra neuron/unit for ReLUs

- **Reformulation with** $v = (w, b) \in \mathbb{R}^{d+1}$ **and** $z = (x, 1) \in \mathbb{R}^{d+1}$**:**

$$\max_{\|v\|_2 \leqslant 1} \left| \sum_{i=1}^{n} g_i (v^\top z_i)_+ \right| = \max_{\|v\|_2 \leqslant 1} \left| \sum_{i \in I_+} (v^\top t_i)_+ - \sum_{i \in I_-} (v^\top t_i)_+ \right|$$

with $I_+ = \{i, g_i \geqslant 0\}$ **and** $I_- = \{i, g_i < 0\}$**, and** $t_i = |g_i| z_i \in \mathbb{R}^{d+1}$**,**

Adding extra neuron/unit for ReLUs
Hausdorff distance between zonotopes

- **Reformulation with** $v = (w, b) \in \mathbb{R}^{d+1}$ **and** $z = (x, 1) \in \mathbb{R}^{d+1}$**:**

$$\max_{\|v\|_2 \leqslant 1} \left| \sum_{i=1}^{n} g_i (v^\top z_i)_+ \right| = \max_{\|v\|_2 \leqslant 1} \left| \sum_{i \in I_+} (v^\top t_i)_+ - \sum_{i \in I_-} (v^\top t_i)_+ \right|$$

with $I_+ = \{i, g_i \geqslant 0\}$ **and** $I_- = \{i, g_i < 0\}$**, and** $t_i = |g_i| z_i \in \mathbb{R}^{d+1}$**,**

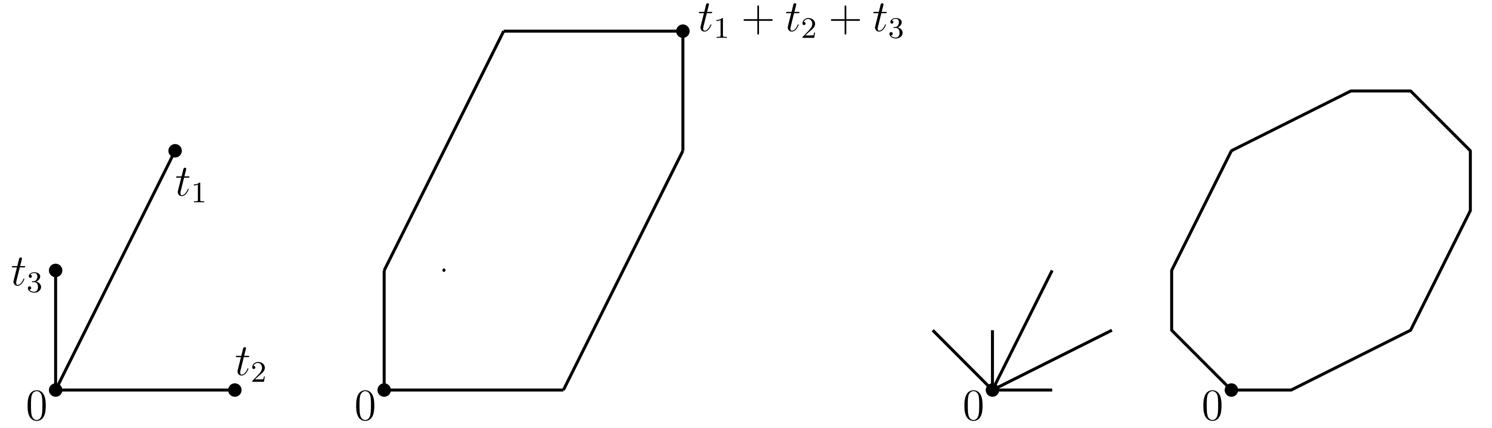
- **By convex duality, equivalent to**

$$\max \left\{ \min_{u_+ \in K_+} \max_{u_- \in K_-} \|u_+ - u_-\|_2, \min_{u_- \in K_-} \max_{u_+ \in K_+} \|u_+ - u_-\|_2 \right\}$$

with $K_+ = \left\{ \sum_{i \in I_+} b_i t_i, \; b_i \in [0, 1] \right\}$ **and** $K_- = \left\{ \sum_{i \in I_-} b_i t_i, \; b_i \in [0, 1] \right\}$

Hausdorff distance between zonotopes

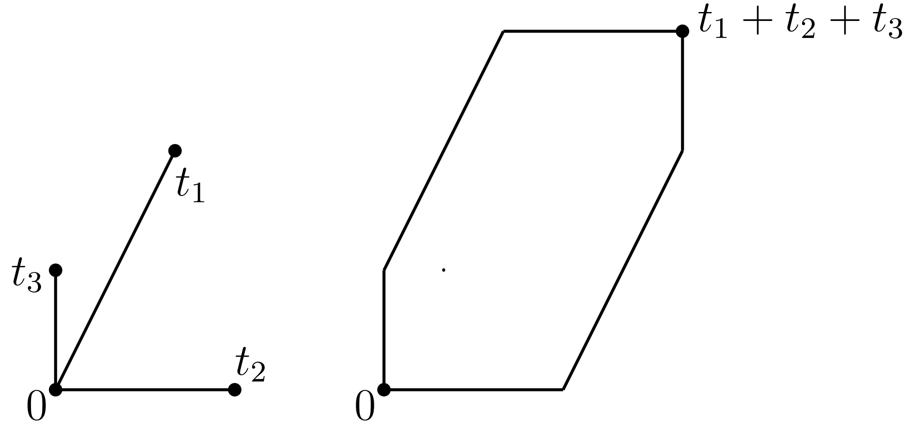
- **Zonotopes** $K = \left\{ \sum_i b_i t_i, \;\; b_i \in [0,1] \right\}$ **and zonoids (Bolker, 1969)**



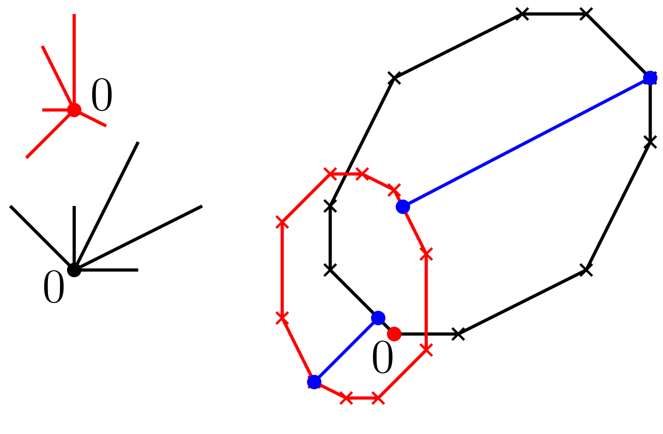
  - **Affine projections of hypercubes**
  - **Zonoids are limits of zonotopes**
  - **In $d = 2$ (only), all centrally symmetric convex sets are zonoids**

Hausdorff distance between zonotopes

- **Zonotopes** $K = \left\{ \sum_i b_i t_i,\ \ b_i \in [0,1] \right\}$ **and zonoids (Bolker, 1969)**

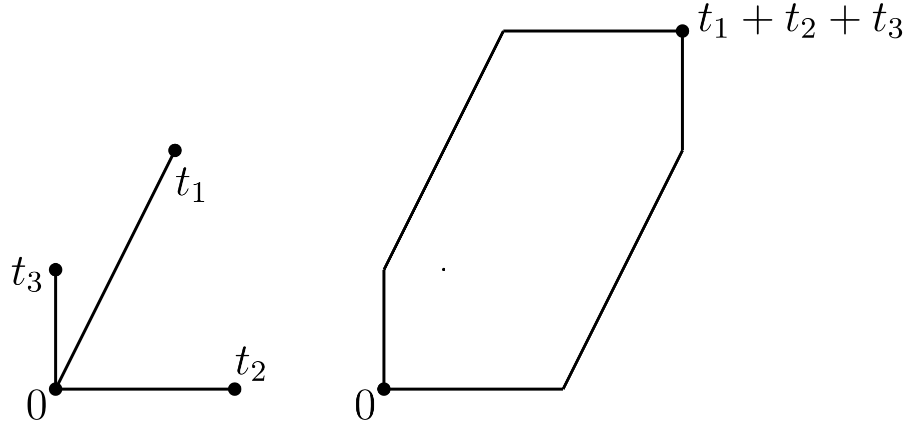


- **Hausdorff distance computation, still hard...**

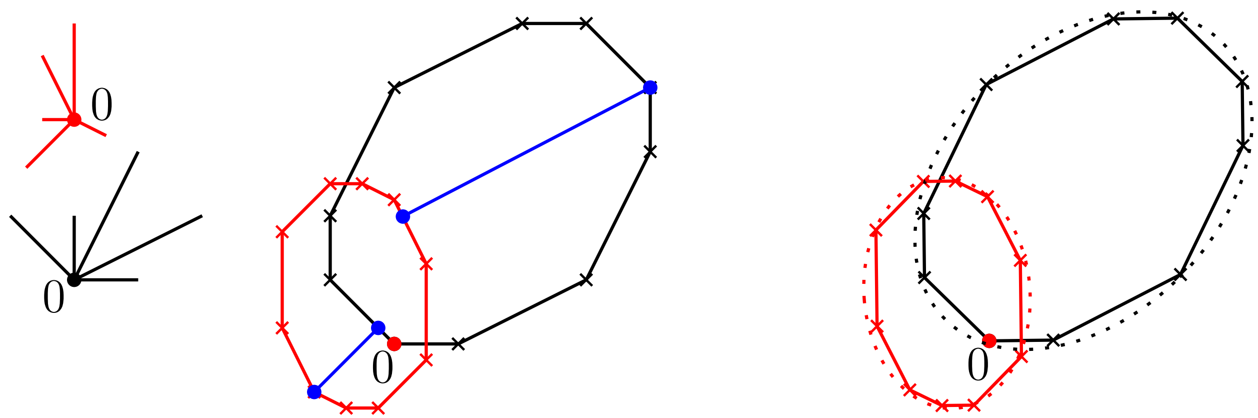


Hausdorff distance between zonotopes

- **Zonotopes** $K = \left\{ \sum_i b_i t_i,\ \ b_i \in [0,1] \right\}$ **and zonoids (Bolker, 1969)**



- **Hausdorff distance computation,** **approximation by ellipsoids?**



Convex relaxations and polynomial-time algorithms

- **Many possibilities (SDP, ellipsoids, etc.), no success (yet)...**
- **(conjectured) Impossible result: for any $g \in \mathbb{R}^n$, find $\hat{v}$ such that $\|\hat{v}\|_2 = 1$ and**

$$\left| \sum_{i=1}^{n} g_i (\hat{v}^\top z_i)_+ \right| \geqslant \frac{1}{\kappa} \max_{\|v\|_2=1} \left| \sum_{i=1}^{n} g_i (v^\top z_i)_+ \right|$$

Convex relaxations and polynomial-time algorithms

- Many possibilities (SDP, ellipsoids, etc.), no success (yet)...

- (conjectured) Impossible result: for any $g \in \mathbb{R}^n$, find $\hat{v}$ such that $\|\hat{v}\|_2 = 1$ and

$$\left| \sum_{i=1}^{n} g_i (\hat{v}^\top z_i)_+ \right| \geqslant \frac{1}{\kappa} \max_{\|v\|_2=1} \left| \sum_{i=1}^{n} g_i (v^\top z_i)_+ \right|$$

- Sufficient result for matching generalization bounds
  - Only in expectation for g standard Gaussian vector
  - Reduction to simple non-convex problem
  - NB: similar to linear binary classification (which is NP-hard)

Why not sampling weights?

- **Sampling m weights (w_i, b_i) and use features $(w_i^\top x + b_i)_+$**
  - **Linear combination and ℓ_2-regularizer**
  - **Equivalent to a kernel $k(x,y) = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}(w_i^\top x + b_i)_+(w_i^\top y + b_i)_+$**

Why not sampling weights?

- **Sampling m weights (w_i, b_i) and use features $(w_i^\top x + b_i)_+$**
  - **Linear combination and ℓ_2-regularizer**
  - **Equivalent to a kernel $k(x,y) = \dfrac{1}{m}\displaystyle\sum_{i=1}^m (w_i^\top x + b_i)_+ (w_i^\top y + b_i)_+$**
- **Letting $m \to \infty$**
  - **$k(x,y)$ tends to $\displaystyle\int_{\mathbb{R}^{d+1}} (w^\top x + b)_+ (w^\top y + b)_+ d\mu(w,b)$**
  - **Random kernel expansion (Neal, 1995; Rahimi and Recht, 2007)**
  - **Can be computed in closed form (Le Roux and Bengio, 2007; Cho and Saul, 2009)**
- **Defines a Hilbert space $\mathcal{F}_2$ with norm γ_2 such that:**

$$\gamma_2(f)^2 = \inf \int_{\mathbb{R}^{d+1}} |\eta(w,b)|^2 d\tau(w,b) \text{ s.t. } f(x) = \int_{\mathbb{R}^{d+1}} (w^\top x + b)_+ \eta(w,b) d\tau(w,b)$$

Generalization properties

- Minimization of empirical risk $\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i))$
  - subject to $\gamma_1(f) \leqslant \delta$: learning weights (w_j, b_j)
  - subject to $\gamma_2(f) \leqslant \delta$: sampling weights (w_j, b_j)
  - NB: $\gamma_1 \leqslant \gamma_2$, i.e., $\mathcal{F}_2 \subset \mathcal{F}_1$

Generalization properties

- Minimization of empirical risk $\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i))$
  - subject to $\gamma_1(f) \leqslant \delta$: learning weights (w_j, b_j)
  - subject to $\gamma_2(f) \leqslant \delta$: sampling weights (w_j, b_j)
  - NB: $\gamma_1 \leqslant \gamma_2$, i.e., $\mathcal{F}_2 \subset \mathcal{F}_1$
- Sampling weights (i.e., using ℓ_2 / kernel methods)
  - No adaptivity (e.g., a single neuron does not belong to $\mathcal{F}_2$)
- Learning sparse weights (i.e., using ℓ_1)
  - Automatic adaptivity to structure
  - E.g., $f(x) = g(W^\top x)$ for W of low-rank

Approximation properties with variation norm

- **Finite variation norm**
  - f $(d/2+3/2)$**-times differentiable** $\Rightarrow \gamma_1(f) \leqslant \gamma_2(f) < \infty$
  - **Smoothness index has to grow with dimension!**

Approximation properties with variation norm

- **Finite variation norm**
  - f $(d/2+3/2)$**-times differentiable** $\Rightarrow \gamma_1(f) \leqslant \gamma_2(f) < \infty$
  - **Smoothness index has to grow with dimension!**

- **Approximation of Lipschitz-continuous functions**
  - f **1-Lipschitz-continuous** $\Rightarrow$ **there exists** g **such that** $\gamma_1(g) \leqslant \delta$ **and with approximation error** $\delta^{-2/(d+1)} \log \delta$
  - **Proof based on spherical harmonics**

Approximation properties with variation norm

- **Finite variation norm**
  - f $(d/2+3/2)$-times differentiable $\Rightarrow \gamma_1(f) \leqslant \gamma_2(f) < \infty$
  - Smoothness index has to grow with dimension!

- **Approximation of Lipschitz-continuous functions**
  - f 1-Lipschitz-continuous $\Rightarrow$ there exists g such that $\gamma_1(g) \leqslant \delta$ and with approximation error $\delta^{-2/(d+1)} \log \delta$
  - Proof based on spherical harmonics

- **Adaptivity**
  - If f depends on a s-dimensional projection, replace d by s
  - Only works for γ_1

Generalization bounds

- **Assuming f^* of a certain form**
  - **Penalizing weight vectors w by ℓ_2-norms**

function space	$\|\cdot\|_2$
$w^\top x + b$	$\frac{d^{1/2}}{n^{1/2}}$
No assumption	$\frac{C(d)}{n^{1/(d+3)}} \log n$
$\sum_{j=1}^{k} f_j(w_j^\top x)$, $w_j \in \mathbb{R}^d$	$\frac{kd^{1/2}}{n^{1/4}} \log n$
$\sum_{j=1}^{k} f_j(W_j^\top x)$, $W_j \in \mathbb{R}^{d \times s}$	$\frac{kd^{1/2}C(s)}{n^{1/(s+3)}} \log n$

Generalization bounds

- **Assuming f^* of a certain form**
  - **Penalizing weight vectors w by ℓ_2-norms**
  - **Assuming q-sparse solution and penalizing w by ℓ_1-norm**

function space	$\|\cdot\|_2$	$\|\cdot\|_1$
$w^\top x + b$	$\frac{d^{1/2}}{n^{1/2}}$	$\sqrt{q}\frac{(\log d)^{1/2}}{n^{1/2}}$
No assumption	$\frac{C(d)}{n^{1/(d+3)}} \log n$	$\frac{q^{1/2}C(d)}{n^{1/(d+3)}} \log n$
$\sum_{j=1}^{k} f_j(w_j^\top x)$, $w_j \in \mathbb{R}^d$	$\frac{kd^{1/2}}{n^{1/4}} \log n$	$\frac{kq^{1/2}(\log d)^{1/2}}{n^{1/4}} \log n$
$\sum_{j=1}^{k} f_j(W_j^\top x)$, $W_j \in \mathbb{R}^{d\times s}$	$\frac{kd^{1/2}C(s)}{n^{1/(s+3)}} \log n$	$\frac{kq^{1/2}C(s)(\log d)^{2/(s+3)}}{n^{1/(s+3)}} \log n$

Conclusion

- Convex neural networks / infinitely many basis functions
  - Adaptivity to structure
  - Corresponding ernel methods are not adaptive
  - Provable high-dimensional non-linear variable selection

- Convex but no polynomial-time algorithm
  - Reduction to approximate Haussdorff distance between zonotopes
  - Open problem

Conclusion

- Convex neural networks / infinitely many basis functions
  - Adaptivity to structure
  - Corresponding ernel methods are not adaptive
  - Provable high-dimensional non-linear variable selection

- Convex but no polynomial-time algorithm
  - Reduction to approximate Haussdorff distance between zonotopes
  - Open problem

- Extensions
  - Multiple outputs
  - Multiple layers
  - Other models (e.g., Gaussian mixtures)

References

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

Y. Bengio, N. Le Roux, P. Vincent, O. Delalleau, and P. Marcotte. Convex neural networks. *Adv. NIPS*, 2006.

E. D. Bolker. A class of convex bodies. *Transactions of the American Mathematical Society*, 145: 323–345, 1969.

Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, 2009.

V. F. Dem'yanov and A. M. Rubinov. The minimization of a smooth convex functional on a convex set. *SIAM Journal on Control*, 5(2):280–294, 1967.

Miro Dudik, Zaid Harchaoui, Jérôme Malick, et al. Lifted coordinate descent for learning with trace-norm regularization. In *AISTATS-Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics-2012*, volume 22, pages 327–336, 2012.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3 (1-2):95–110, 1956.

Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. Technical Report 1302.2325, arXiv, 2013.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

V. Kurkova and M. Sanguineti. Bounds on rates of variable-basis and neural-network approximation. *IEEE Transactions on Information Theory*, 47(6):2659–2665, Sep 2001.

Nicolas Le Roux and Yoshua Bengio. Continuous neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 404–411, 2007.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6): 861–867, 1993.

Radford M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, 2007.

S. Rosset, G. Swirszcz, N. Srebro, and J. Zhu. ℓ_1-regularization in infinite dimensional feature spaces. In *Proc. COLT*, 2007.