On the Complexity of Learning with Kernels

Nicolò Cesa-Bianchi ¹  Yishay Mansour ²  Ohad Shamir ³

¹Università degli Studi di Milano
²Tel-Aviv University and Microsoft Research
³Weizmann Institute

COLT 2015
Kernel Learning

\[ x \mapsto \psi(x), \quad \langle \psi(x), \psi(x') \rangle = k(x, x') \]

**Kernel Empirical Risk Minimization**

Given \( \{(x_i, y_i)\}_{i=1}^m \), solve

\[
\min_{w \in \mathcal{W}} \frac{1}{m} \sum_{i=1}^{m} \ell(\langle w, \psi(x_i) \rangle, y_i) + \frac{\lambda}{2} \|w\|^2.
\]
Kernel Learning

Letting $K_{i,j} = k(x_i, x_j)$ be the $m \times m$ kernel matrix, equivalent to

$$\min_{\alpha : w(\alpha) \in W} \frac{1}{m} \sum_{i=1}^{m} \ell \left( \alpha^\top K e_i, y_i \right) + \frac{\lambda}{2} \alpha^\top K \alpha$$

- Convex problem, solvable in polynomial time
Letting $K_{i,j} = k(x_i, x_j)$ be the $m \times m$ kernel matrix, equivalent to

$$\min_{\alpha : w(\alpha) \in \mathcal{W}} \frac{1}{m} \sum_{i=1}^{m} \ell \left( \alpha^\top K e_i, y_i \right) + \frac{\lambda}{2} \alpha^\top K \alpha$$

- Convex problem, solvable in polynomial time
- Doesn’t scale well: Need to compute and handle $m \times m$ matrix
- Many methods proposed to make kernels more efficient

Schölkopf and Smola (2001); Fine and Scheinberg (2002); Shawe Taylor and Cristianini (2004); Drineas and Mahoney (2005); Bach and Jordan (2005); Yao, Rosasco, Caponnetto (2007); Kumar, Mohri, Talwalkar (2009); Rahimi and Recht (2007,2008); Cavallanti, Cesa-Bianchi, Gentile (2007); Dekel, Shalev-Shwartz, Singer (2008); Raskutti, Wainwright, Yu (2014); Mahoney and Drineas (2009); Cortes, Mohri, Talwalkar (2010); Yang, Mahdavi, Jin, Zhou (2012); Cotter, Shalev-Shwartz, Srebro (2012); Zhang, Duchi, Wainwright (2013); Bach (2013); Dai, Xie, He, Liang, Raj, Balcan, Song (2014); Lin, Weng, Zhang (2014); Alaoui and Mahoney (2014); Hsieh, Si, Dhillon(2014)....
Making Kernels More Efficient

Methods generally use one or both of the following:

1. Limiting # of kernel evaluations
   - E.g. sampling rows/columns (Nyström); blocks (divide-and-conquer, early stopping, budgeted perceptrons); random entries...

2. Approximating $K$ by a low-rank matrix
   - E.g. random features

Lots of work on algorithms and upper bounds – but how well can we hope to perform??
Model:

- \( y_1, \ldots, y_m \) are given, but \( x_1, \ldots, x_m \) and kernel matrix \( K \) are unknown
- **Any** \( B \) entries of \( K \) can be (adaptively) observed

How well can we solve the kernel ERM optimization problem?
Model:

- $y_1, \ldots, y_m$ are given, but $x_1, \ldots, x_m$ and kernel matrix $K$ are unknown
- Any $B$ entries of $K$ can be (adaptively) observed

How well can we solve the kernel ERM optimization problem?

Answer:

- Depends on kernel matrix, but also on loss function and regularization
- Not the same question as matrix approximation!
**Budget Constraints**

Example (Kernel ERM, linear loss, soft regularization)

\[
\min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell \left( \alpha^\top K e_i, y_i \right) + \frac{\lambda}{2} \alpha^\top K \alpha
\]

where \( \ell(u, y) = uy \)
Example (Kernel ERM, linear loss, soft regularization)

\[
\min_\alpha \frac{1}{m} \sum_{i=1}^{m} \ell \left( \alpha^\top K e_i, y_i \right) + \frac{\lambda}{2} \alpha^\top K \alpha
\]

where \( \ell(u, y) = uy \)

How many entries of \( K \) needed to solve up to \( \epsilon \) error?
Example (Kernel ERM, linear loss, soft regularization)

\[
\min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell \left( \alpha^\top K e_i, y_i \right) + \frac{\lambda}{2} \alpha^\top K \alpha
\]

where \( \ell(u, y) = uy \)

How many entries of \( K \) needed to solve up to \( \epsilon \) error?

Zero!

\[
\alpha_{OPT} = -\frac{1}{\lambda m} y
\]
Hard Kernel Matrices: $\mathcal{K}_{d,m}$

Row/column permutations of $m \times m$ block-diagonal matrices with $\leq d$ blocks

\[
\begin{bmatrix}
1 & 1 & 1 & \cdots & 0 \\
0 & 1 & 1 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\]

rank $\leq d$

Induced by e.g. linear kernel, homogeneous polynomial kernel, Gaussian kernel... although technique is really more general
Absolute Loss, no strong convexity

\[ \min_{\alpha : \alpha^\top K \alpha \leq 2} \frac{1}{m} \sum_{i=1}^{m} \left| \alpha^\top K e_i - y_i \right| \quad , \quad y_i \in [-1, +1] \]
Absolute Loss, no strong convexity

\[
\min_{\alpha : \alpha^\top K \alpha \leq 2} \frac{1}{m} \sum_{i=1}^{m} \left| \alpha^\top K e_i - y_i \right|, \quad y_i \in [-1, +1]
\]

**Theorem**

*For any \(m\) and budget constraint \(B \ll m^2\), \(\exists m \times m\) kernel matrix such that error \(\geq \Omega \left( B^{-1/4} \right)\).*

Optimal: Achieved by solving kernel ERM on a sub-sample of \(\sqrt{B}\) training examples

(can also get lower bound for any fixed rank \(d\))
Proof Idea

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & 1 & 1 \\
0 & 0 & \ddots
\end{bmatrix}
\]
Proof Idea
Proof Idea
Proof Idea
Proof Idea

- Pick $y_i = \frac{1}{\sqrt{d}}$ for all $i$
- Exists zero-error $\alpha$ in domain
Proof Idea

- Pick $y_i = \frac{1}{\sqrt{d}}$ for all $i$
- Exists zero-error $\alpha$ in domain
- Finding $\alpha$ with error $\ll \frac{1}{\sqrt{d}}$ reduces to guessing correctly size of almost all blocks

$$
\frac{1}{m} \sum_{i=1}^{m} \left| \alpha K e_i - \frac{1}{\sqrt{d}} \right|
= \frac{1}{m} \sum_{i=1}^{m} \left| \sum_{j \in \text{Block}(i)} \alpha_j - \frac{1}{\sqrt{d}} \right|
$$

However, hard to get information on all blocks, since matrix is sparse and randomly permuted

Lemma
If $B < \frac{3}{50} d^2$, expected number of "missed" blocks $\geq d^2$
Proof Idea

- Pick $y_i = \frac{1}{\sqrt{d}}$ for all $i$
- Exists zero-error $\alpha$ in domain
- Finding $\alpha$ with error $\ll \frac{1}{\sqrt{d}}$ reduces to guessing correctly size of almost all blocks

\[
\frac{1}{m} \sum_{i=1}^{m} \left| \alpha K e_i - \frac{1}{\sqrt{d}} \right|
\]

= \[
\frac{1}{m} \sum_{i=1}^{m} \left| \sum_{j \in \text{Block}(i)} \alpha_j - \frac{1}{\sqrt{d}} \right|
\]

- However, hard to get information on all blocks, since matrix is sparse and randomly permuted

Lemma

If $B < \frac{3}{50} d^2$, expected number of “missed” blocks $\geq \frac{d}{2}$
\[ \min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell \left( \alpha^\top K e_i, y_i \right) + \frac{\lambda}{2} \alpha^\top K \alpha \quad , \quad y_i \in \mathcal{Y} \]
Soft Regularization, General Losses

\[
\min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell \left( \alpha^\top K e_i, y_i \right) + \frac{\lambda}{2} \alpha^\top K \alpha , \quad y_i \in \mathcal{Y}
\]

**Theorem**

For any rank parameter \(d\), if \(m > 2^7 d\) and \(B < \frac{3}{50} d^2\),
\[\exists \ m \times m \ rank \ d \ kernel \ matrix \ s.t. \ error \ is\]

\[
\Omega \left( \lambda d \min_{p \in \left[\frac{1}{2}, 2\right]} \max_{y \in \mathcal{Y}} \left(2u_1^* - u_2^*\right)^2 \right) \quad \text{where}
\]

\[u_1^* = \arg \min_u \ell(u, y) + p\lambda d u^2 , \quad u_2^* = \arg \min_u \ell(u, y) + \frac{p\lambda d}{2} u^2\]

Lower bound depends on non-linearity/smoothness of the loss
Some Corollaries

Optimization error lower bounds for:

- Linear loss: $0$
- Absolute loss: $\Omega \left( \frac{1}{\lambda} \sqrt{B} \right)$
- Squared loss: $\Omega \left( \min\{1, \frac{1}{\lambda} \left( \frac{1}{\lambda} \sqrt{B} \right)^{3/2} \} \right)$
  Need $B \geq \frac{1}{\lambda^2}$ for sub-constant error
  E.g. if $\lambda \leq \frac{1}{m}$, no efficient learning is possible in the worst-case
- Hinge loss: Need $B \geq \frac{1}{\lambda^2}$ for sub-constant error
Some Corollaries

Optimization error lower bounds for:

- Linear loss: 0
- Absolute loss: $\Omega\left(\frac{1}{\lambda\sqrt{B}}\right)$
  Optimal, achieved by "trivial" algorithm of subsampling $\sqrt{B}$ training examples
- Squared loss: $\Omega\left(\min\left\{1, \frac{1}{\lambda\sqrt{B}}\right\}\right)$
  Need $B \geq \frac{1}{\lambda^2}$ for sub-constant error
  E.g. if $\lambda \leq \frac{1}{m}$, no efficient learning is possible in the worst-case
- Hinge loss: Need $B \geq \frac{1}{\lambda^2}$ for sub-constant error

Cesa-Bianchi, Mansour, Shamir
On the Complexity of Learning with Kernels
Some Corollaries

Optimization error lower bounds for:

- Linear loss: 0
- Absolute loss: $\Omega\left(\frac{1}{\lambda\sqrt{B}}\right)$
  - Optimal, achieved by “trivial” algorithm of subsampling $\sqrt{B}$ training examples

Cesa-Bianchi, Mansour, Shamir
On the Complexity of Learning with Kernels
Some Corollaries

Optimization error lower bounds for:

- Linear loss: 0
- Absolute loss: $\Omega \left( \frac{1}{\lambda \sqrt{B}} \right)$
  - Optimal, achieved by “trivial” algorithm of subsampling $\sqrt{B}$ training examples
- Squared loss: $\Omega \left( \min \left\{ 1, \frac{1}{(\lambda \sqrt{B})^3} \right\} \right)$

$B \geq \frac{1}{\lambda^2}$ for sub-constant error

E.g. if $\lambda \leq \frac{1}{m}$, no efficient learning is possible in the worst-case

Hinge loss: $B \geq \frac{1}{\lambda^2}$ for sub-constant error
Optimization error lower bounds for:

- **Linear loss**: $0$

- **Absolute loss**: $\Omega \left( \frac{1}{\lambda \sqrt{B}} \right)$
  - Optimal, achieved by "trivial" algorithm of subsampling $\sqrt{B}$ training examples

- **Squared loss**: $\Omega \left( \min \left\{ 1, \frac{1}{(\lambda \sqrt{B})^3} \right\} \right)$
  - Need $B \geq \frac{1}{\lambda^2}$ for sub-constant error
  - E.g. if $\lambda \leq \frac{1}{m}$, no efficient learning is possible in the worst-case
Some Corollaries

Optimization error lower bounds for:

- Linear loss: 0
- Absolute loss: $\Omega \left( \frac{1}{\lambda \sqrt{B}} \right)$
  - Optimal, achieved by “trivial” algorithm of subsampling $\sqrt{B}$ training examples
- Squared loss: $\Omega \left( \min \left\{ 1, \frac{1}{(\lambda \sqrt{B})^3} \right\} \right)$
  - Need $B \geq 1/\lambda^2$ for sub-constant error
  - E.g. if $\lambda \leq 1/m$, no efficient learning is possible in the worst-case
- Hinge loss: Need $B \geq 1/\lambda^2$ for sub-constant error
Different model: Algorithm performs kernel ERM using any low-rank surrogate $K'$ for $K$
Different model: Algorithm performs kernel ERM using any low-rank surrogate $K'$ for $K$

Theorem (Kernel ridge regression)

For rank-$d$ surrogate, soft regularization with parameter $\lambda$, error is

$$\Omega \left( \min \left\{ 1, \frac{1}{(\lambda d)^3} \right\} \right)$$

Implication: For sub-constant error, required rank is $\Omega \left( \frac{1}{\lambda} \right)$
No low-rank approximation would work when $\lambda = 1/m$
Bad News

∃ losses and kernel matrices for which speeding up kernel learning is impossible (except by throwing away data).

For general losses with regularization, computational effort scales with $1/\lambda$.

Cannot be sped up in the worst case, when $\lambda = 1/m$.

Non-smooth regression losses appear to be generally difficult.

Good News

Lower bounds are weaker for low-rank kernel matrices.

One-sided losses (e.g., hinge loss).

Smooth losses and strong convexity.

Can we utilize such assumptions on the loss?

Or can the lower bounds be improved?
Bad News

- ∃ losses and kernel matrices for which speeding up kernel learning is impossible (except by throwing away data)
- For general losses with regularization, computational effort scales with $1/\lambda$
  - Cannot be sped up in the worst case, when $\lambda = 1/m$
- Non-smooth regression losses appear to be generally difficult
Bad News

- ∃ losses and kernel matrices for which speeding up kernel learning is impossible (except by throwing away data)
- For general losses with regularization, computational effort scales with $1/\lambda$
  - Cannot be sped up in the worst case, when $\lambda = 1/m$
- Non-smooth regression losses appear to be generally difficult

Good News

- Lower bounds are weaker for
  - Low-rank kernel matrices
  - One-sided losses (e.g. hinge loss)
  - Smooth losses and strong convexity

Can we utilize such assumptions on the loss? Or can the lower bounds be improved?
THANKS!