Learning with SGD and Random Features

Luigi Carratino[1], Alessandro Rudi[2] and Lorenzo Rosasco[1,3]

[1] DIBRIS – Università degli Studi di Genova, Genoa, Italy
[2] INRIA – Sierra-project team & École Normale Supérieure, Paris, France
[3] LCSL – Istituto Italiano di Tecnologia, Genoa, Italy & Massachusetts Institute of Technology, Cambridge, USA

Neural Information Processing Systems Foundation

Motivations

Sketching and stochastic gradient methods are two of the most common techniques to derive efficient large scale learning algorithms.

We study the estimator defined by **stochastic gradient with mini-batches and random features,** showing how **different parameters**, such as number of features, iterations, step-size and mini-batch size **control the learning properties** of the solutions.

Learning Setting

Given $(x_i, y_i)_{i=1}^n$ i.i.d. samples from ρ, and a linear model with feature map $\phi : \mathcal{X} \to \mathcal{H}$ minimize the **expected risk**

$$\min_{w \in \mathcal{H}} \mathcal{E}(w) \qquad \mathcal{E}(w) = \int \left(y - w^\top \phi(x)\right)^2 d\rho(x, y)$$

space of models — unknown

Note: the problem can not be directly solved

In practice: empirical risk minimization $\widehat{w}_\lambda = \operatorname{argmin}_{w \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \left(y_i - w^T \phi(x_i)\right) + \lambda \|w\|^2$

Saving Time

Idea: use an iterative solver like **stochastic gradient descent**

SGD $w_{t+1} = w_t - \gamma \nabla \left(\left(y_t - w_t^T \phi(x_t)\right)^2 + \lambda \|w_t\|^2 \right)$ with $t = 1, \dots, T$, $\lambda \geq 0$

Saving Space

Idea: map the points into a lower dimensional space with $\phi_M : \mathcal{X} \to \mathbb{R}^M$

ERM with RF $\widehat{w}_{M,\lambda} = \operatorname{argmin}_{w \in \mathbb{R}^M} \frac{1}{n} \sum_{i=1}^n \left(y_i - w^T \phi_M(x_i)\right)^2 + \lambda \|w\|^2$

Random features[1] $\phi_M(x) = \left(\sigma\left(x^\top s_1\right), \dots, \sigma\left(x^\top s_M\right)\right)$

- $s_1, \dots, s_M$ iid random vectors
- $\sigma : \mathbb{R} \to \mathbb{R}$ non linear function

random feature $\psi(x, \beta)$

Note:

- The learned model $w^\top \phi_M(x) = \sum_{j=1}^M w^{(j)} \sigma(x^\top s_j)$ is a **NN with random weights**
- For many example of **RF**, when M is big $\phi_M(x)^\top \phi_M(x') \approx \phi(x)^\top \phi(x') = K(x, x')$ (positive definite kernel; More efficient than)

Examples:

- $\psi(x, \beta) = \cos(x^\top s + b)$ with $s \sim \mathcal{N}(0, \sigma^2)$, $b \sim \mathcal{U}(0, 2\pi)$ $\longrightarrow$ $\phi_M(x)^\top \phi_M(x') \approx e^{-\frac{\|x - x'\|^2}{2\sigma^2}}$ **Gaussian kernel**
- $\psi(x, \beta) = x^\top s$ with $s \sim \mathcal{N}(0, 1)$ $\longrightarrow$ $\phi_M(x)^\top \phi_M(x') \approx x^\top x'$ **Linear kernel**

SGD with Random Features

SGD-RF $\widehat{w}_{t+1} = \widehat{w}_t - \gamma_t \frac{1}{b} \sum_{i=b(t-1)+1}^{bt} \nabla \left(\left(y_{j_i} - \widehat{w}_t^\top \phi_M(x_{j_i})\right)^2 \right)$

with $t = 1, \dots, T$ and $j_1, \dots, j_{bT} \sim U(1, n)$

Note: one pass over the data is reached after $\lceil \frac{n}{b} \rceil$ iterations

Computational Complexity

• **Time:**	$O(MbT)$	(optimality) →	• **Time:** $O(n\sqrt{n})$
• **Space:**	$O(M)$		• **Space:** $O(\sqrt{n})$

Theoretical Analysis

Theorem: Under basic assumptions

$$\mathbb{E}\mathcal{E}(\widehat{w}_{t+1}) - \inf_{w \in \mathcal{H}} \mathcal{E}(w) \lesssim \frac{\gamma}{b} + \left(\frac{\gamma t}{M} + 1\right) \frac{\gamma t}{n} + \frac{1}{\gamma t} + \frac{1}{M}.$$

Which is the **best parameter choice** to have the fastest rate?

Corollary: Under basic assumptions, for one of the following conditions

1. $b = 1$, $\gamma \simeq \frac{1}{n}$, and $T = n\sqrt{n}$ iterations ($\sqrt{n}$ passes over the data);
2. $b = 1$, $\gamma \simeq \frac{1}{\sqrt{n}}$, and $T = n$ iterations (1 pass over the data);
3. $b = \sqrt{n}$, $\gamma \simeq 1$, and $T = \sqrt{n}$ iterations (1 pass over the data);
4. $b = n$, $\gamma \simeq 1$, and $T = \sqrt{n}$ iterations ($\sqrt{n}$ passes over the data);

and

$$M = \sqrt{n}$$

we have

$$\mathbb{E}\mathcal{E}(\widehat{w}_T) - \inf_{w \in \mathcal{H}} \mathcal{E}(w) \lesssim \frac{1}{\sqrt{n}}$$

Note:

- With no mini-batch, the step-size $\frac{1}{n} \leq \gamma \leq \frac{1}{\sqrt{n}}$ **determines the number of passes**
- Parameter choices 2. and 3. imply **lower time complexity**
- **Constant step-size** requires mini-batch size $\geq \sqrt{n}$
- Mini-batch size $> \sqrt{n}$ **do not allow** bigger step-size, and **requires more passes**

Remarks

Faster rates can be achieved under refined assumptions

Analysis holds for **decreasing step-size**

Comparison with **Ridge Regression:**

- **same rate** of "vanilla" **Kernel Ridge Regression (KRR)**
- same rate and **same number of RF** of **Random Features KRR [2]**

Comparison with **Stochastic Gradient Descent:**

- As $M \to \infty$, **SGD-RF** recovers the **same rates and results** of
  - **one-pass SGD [3]**
  - **multiple-pass SGD [4,5]**

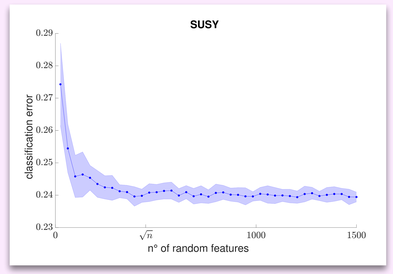
Note: [3] allows constant step-size by averaging, while SGD-RF by mini-batching

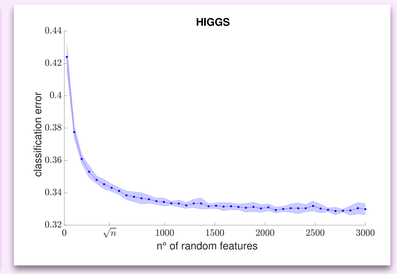
Experiments

Random Fourier Features that approximate the **Gaussian Kernel**

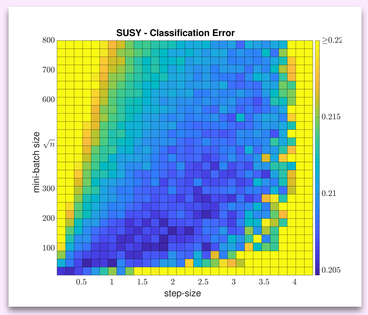
Practice validates theory:

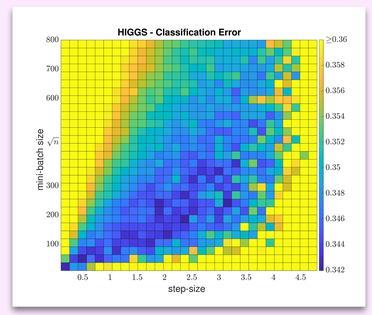
- A number of RF bigger that $\mathcal{O}(\sqrt{n})$ **does not improve accuracy**





- **Bigger** mini-batch size **requires bigger** step-size.
- One pass over the data is **not enough** for batch-size bigger than $\sqrt{n}$





References

[1] A. Rahimi and B. Recht. **Random features for large-scale kernel machines**. NeurIPS, 2008
[2] A. Rudi and L. Rosasco. **Generalization properties of learning with random features**. NeurIPS, 2017
[3] A. Dieuleveut, N. Flammarion and F. Bach. **Harder, better, faster, stronger convergence rates for least-squares regression**. JMLR, 2017
[4] J. Lin and L. Rosasco. **Optimal rates for multi-pass stochastic gradient methods**. JMLR, 2017
[5] L. Rosasco and S. Villa. **Learning with incremental iterative regularization**. NeurIPS, 2015