MIT 9.520/6.860, Fall 2018
Statistical Learning Theory and Applications

Class 07: Learning with (Random) Projections

Lorenzo Rosasco

Learning algorithm design so far

- ERM + Optimization

$$\widehat{w}_\lambda = \arg\min_{w \in \mathbb{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^n \ell(y_i, w^\top x_i) + \lambda \|w\|^2}_{\widehat{L}^\lambda(w)}, \qquad w_{t+1} = w_t - \gamma_t \nabla \widehat{L}^\lambda(w_t)$$

- Learning by optimization (GD/SGD)

$$\widehat{w}_{t+1} = \widehat{w}_t - \gamma_t \nabla \widehat{L}(\widehat{w}_t), \qquad \underbrace{\frac{1}{n}\sum_{i=1}^n \ell(y_i, w^\top x_i)}_{\widehat{L}(w)}.$$

Non linear extensions via features/kernels.

Statistics and computations

- Regularization by penalization separates statistics and computations

- Implicit regularization: training time controls statistics and computations

What about memory?

Large scale learning

In many modern applications, space is the real constraint.

$$\underbrace{\widehat{X}}_{n \times d}, \qquad \underbrace{\widehat{X}^\top \widehat{X}}_{d \times d}, \qquad \underbrace{\widehat{X}\widehat{X}^\top \text{ or } \widehat{K}}_{n \times n}$$

Think $n \sim d$ large!

Projections and dimensionality reduction

Let S be a $d \times M$ matrix and

$$\widehat{X}_M = \widehat{X} S$$

Equivalenty

$$x \in \mathbb{R}^d \quad \mapsto \quad x_M = (s_j^\top x)_{j=1}^M \in \mathbb{R}^m$$

with $s_1, \dots, s_M$ columns of S.

Learning with projected data

$$\min_{w \in \mathbb{R}^M} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, w^\top (x_M)_i) + \lambda \|w\|^2, \qquad \lambda \geq 0$$

We will focus on ERM based learning and least squares in particular.

PCA

The SVD of $\widehat{X}$ is

$$\widehat{X} = U\Sigma V^T$$

Consider V_M the matrix $d \times M$ of the first M columns of V.

A corresponding projection is given by

$$\widehat{X}_M = \widehat{X}S, \qquad S = V_M.$$

Representer theorem for PCA

Note that

$$\widehat{X} = U\Sigma V^T \qquad \Leftrightarrow \qquad \widehat{X}^\top = V\Sigma U^\top \qquad \Leftrightarrow \qquad V = \widehat{X}^\top U\Sigma^{-1}$$

and $V_M = \widehat{X}^\top U_M \Sigma_M^{-1}$.

Then

$$\widehat{X}_M = \widehat{X} V_M = \underbrace{\widehat{X}\widehat{X}^\top}_{\widehat{K}} U_M \Sigma_M^{-1} = U_M \Sigma_M$$

and for any x

$$x^\top v_j = \sum_{i=1}^n \underbrace{x^\top x_i}_{k(x,x_i)} \frac{u_j^i}{\sigma_j},$$

with $(u_j, \sigma_j^2)_j$ eigenvectors/eigenvalues of $\widehat{K}$.

Kernel PCA

If Φ is a feature map, then the SVD in feature space is

$$\widehat{\Phi} = U \Sigma V^T$$

and if V_M is the matrix $d \times M$ of the first M columns of V,

$$\widehat{\Phi}_M = \widehat{\Phi} V_M.$$

Equivalently using kernels

$$\widehat{\Phi}_M = \widehat{K} U_M \Sigma_M^{-1} = U_M \Sigma_M,$$

and for any x

$$\Phi(x)^\top v_j = \sum_{i=1}^n k(x, x_i) \frac{u_j^i}{\sigma_j}.$$

PCA+ERM for least squares

Consider (no penalization)

$$\min_{w \in \mathbb{R}^M} \frac{1}{n} \left\| \widehat{X}_M w - \widehat{Y} \right\|^2 .$$

The solution is[1]

$$\widehat{w}_M = (\widehat{X}_M^\top \widehat{X}_M)^{-1} \widehat{X}_M^\top \widehat{Y}.$$

[1] Assuming invertibility for simplicity. In general replace with pseudoiverse.

PCA+ERM for least squares

It is easy to see that that , for all x

$$f_M(x) = x_M^\top \widehat{w}_M = \sum_{j=1}^{M} \frac{1}{\sigma_j} u_j^\top \widehat{Y} v_j^\top x$$

where $x_M = V_M x$.

Essentially due to the fact that

$$\widehat{X}_M^\top \widehat{X}_M = V_M^\top \widehat{X}^\top \widehat{X} V_M$$

is the covariance matrix projected on its first M eigenvectors.

PCR, TSVD, Filtering

$$f_M(x) = \sum_{j=1}^{M} \frac{1}{\sigma_j} u_j^\top \widehat{Y} v_j^\top x$$

- PCA+ERM is called Principal component regression in statistics
- ...and truncated singular value decomposition in linear algebra.
- It corresponds to the spectral filter

$$F(\sigma_j) = \begin{cases} \frac{1}{\sigma_j}, & j \leq M \\ 0, & \text{oth.} \end{cases}$$

Compare to Tikhonov and Landweber,

$$F_{\text{Tik.}}(\sigma_j) = \sigma_j/(1 + \lambda\sigma_j) \qquad F_{\text{Land.}}(\sigma_j) = (1 - (1 - \gamma\sigma_j)^t)\sigma_j^{-1}.$$

Projection and complexity

Then,

- PCA + ERM = regularization.
- In principle, down stream learning is computationally cheaper...

...however SVD requires time

$$O(nD^2 \vee d^3)$$

or with kernel matrices

$$O(n^2 C_K \vee n^3)$$

Sketching

Let S be a $d \times M$ matrix s.t. $S_{ij} \sim \mathcal{N}(0,1)$ and

$$\widehat{X}_M = \widehat{X} S.$$

Computing $\widehat{X}_M$ is time $O(ndM)$ and memory $O(nd)$

Dimensionality reduction with sketching

Note that if $x_M = S^\top x$ and $x'_M = S^\top x'$, then

$$\frac{1}{M}\,\mathbb{E}[x_M^\top x'_M] = \frac{1}{M}\,\mathbb{E}[x^\top S S^\top x'] = x^\top\,\mathbb{E}[S S^\top] x' = \frac{1}{M} x^\top \sum_{j=1}^{M} \underbrace{\mathbb{E}[s_j s_j^\top]}_{Identity} x' = x^\top x'.$$

- Inner products, norms distances preserved in expectation..
- ... and with high probability for given M (Johnson-Linderstrauss Lemma).

Least squares with sketching

Consider

$$\min_{w \in \mathbb{R}^M} \frac{1}{n} \left\| \widehat{X}_M w - \widehat{Y} \right\|^2 + \lambda \|w\|^2, \quad \lambda > 0.$$

Regularization is needed. For sketching

$$\widehat{X}_M^\top \widehat{X}_M = S^\top \widehat{X}^\top \widehat{X} S,$$

is **not** the covariance matrix projected on its first M eigenvectors, but

$$\mathbb{E}[\widehat{X}_M \widehat{X}_M^\top] = \mathbb{E}[\widehat{X} S S^\top \widehat{X}^\top] = \widehat{X}\widehat{X}^\top.$$

There is extra variability.

Least squares with sketching (cont.)

Consider

$$\min_{w \in \mathbb{R}^M} \frac{1}{n} \left\| \widehat{X}_M w - \widehat{Y} \right\|^2 + \lambda \|w\|^2, \quad \lambda > 0.$$

The solution is

$$\widehat{w}_{\lambda,M} = (\widehat{X}_M^\top \widehat{X}_M + \lambda n I)^{-1} \widehat{X}_M^\top \widehat{Y}.$$

Computing $\widehat{w}_{\lambda,M}$ is time $O(nM^2 + ndM)$ and memory $O(nM)$.

Beyond linear sketching

Let S be a $d \times M$ random matrix and

$$\widehat{X}_M = \sigma(\widehat{X}S)$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is a given nonlinearity.

Then consider functions of the form,

$$f_M(x) = x_M^\top w = \sum_{j=1}^{M} w^j \sigma(s_j^\top x).$$

Learning with random weights networks

$$f_M(x) = x_M^\top w = \sum_{j=1}^{M} w^j \sigma(s_j^\top x)$$

Here, $w^1, \dots, w^M$ can be computed solving a convex problem

$$\min_{w \in \mathbb{R}^M} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_M(x_i))^2 + \lambda \|w\|^2, \quad \lambda > 0,$$

in time $O(nM^2 + ndM)$ and memory $O(nM)$.

Neural networks, random features and kernels

$$f_M(x) = \sum_{j=1}^{M} w^j \sigma(s_j^\top x)$$

- It is a *one hidden layer neural network with random weights*.
- It is defined by a random feature map $\Phi_M(x) = \sigma(S^\top x)$.
- There are a number of cases in which

$$\mathbb{E}[\Phi_M(x)^\top \Phi_M(x')] = k(x, x')$$

with k a suitable pos. def. kernel k.

Random Fourier features

Let $X = \mathbb{R}$, $s \sim \mathcal{N}(0,1)$ and

$$\Phi_M^j(x) = \frac{1}{\sqrt{M}} \underbrace{e^{is_j x}}_{\text{complex exp.}} .$$

For $k(x,x') = e^{-|x-x'|^2\gamma}$ it holds

$$\mathbb{E}[\Phi_M(x)^\top \Phi_M(x')] = k(x,x').$$

Proof: from basic properties of the Fourier transform

$$e^{-|x-x'|^2\gamma} = const. \int ds \underbrace{e^{isx}}_{\text{Inv. transf. -}} \underbrace{e^{-isx}}_{\text{Transl.}} \underbrace{e^{\frac{s^2}{\gamma}}}_{\text{- Tranf. of Gaussian}} .$$

Random Fourier features (cont.)

- The above reasoning immediately extends to $X = \mathbb{R}^d$.

- Using symmetry one can show the same result holds for

$$\Phi_M^j(x) = \frac{1}{\sqrt{M}} \cos(s_j^\top x + b_j)$$

with b_j uniformly distributed.

Other random features

The relation

$$\mathbb{E}[\Phi_M(x)^\top \Phi_M(x')] = k(x, x').$$

is satisfied by a number of nonlinearities and corresponding kernels:

- ReLU $\sigma(a) = |a|_+ \dots$
- Sigmoidal $\sigma(a), \dots$
- $\dots$

As for all feature map the relation with kernels is not one two one.

Infinite networks and large scale kernel methods

- One hidden layer network with infinite random weights= kernels.

- Random features are an approach to scaling kernel methods: from

$$\text{time} O(n^2 C_k \vee n^3) \qquad \text{memory} O(n^2)$$

to

$$\text{time} O(ndM \vee nM^2) \qquad \text{memory} O(nM)$$

Subsampling aka Nyström method

Through the representer theorem, the ERM solution has the form,

$$w = \sum_{i=1}^{n} x_i c_i = \widehat{X}^\top c.$$

For $M < n$, choose a set of *centers* $\{\widetilde{x}_1, \ldots, \widetilde{x}_M\} \subset \{x_1, \ldots, x_n\}$ and let

$$w_M = \sum_{i=1}^{M} x_i (c_M)_i = \widetilde{X}_M^\top c_M.$$

Least squares with Nyström centers

Consider

$$\min_{w_M \in \mathbb{R}^{d}} \frac{1}{n} \left\| \widehat{X} w_M - \widehat{Y} \right\|^2 + \lambda \|w_M\|^2, \quad \lambda > 0.$$

Equivalently

$$\min_{c \in \mathbb{R}^{M}} \frac{1}{n} \| \underbrace{\widehat{X} \widetilde{X}_M^\top}_{\widehat{K}_{nM}} c_M - \widehat{Y} \|^2 + \lambda c_M^\top \underbrace{\widetilde{X}_M \widetilde{X}_M^\top}_{\widehat{K}_M} c_M, \quad \lambda > 0.$$

Least squares with Nyström centers

$$\min_{c \in \mathbb{R}^{M}} \frac{1}{n} \| \underbrace{\widehat{X} \widetilde{X}_M^\top}_{\widehat{K}_{nM}} c_M - \widehat{Y} \|^2 + \lambda c_M^\top \underbrace{\widetilde{X}_M \widetilde{X}_M^\top}_{\widehat{K}_M} c_M, \quad \lambda > 0.$$

The solutions is

$$\widehat{c}_{\lambda,M} = (\widehat{K}_{nM}^\top \widehat{K}_M + n\lambda \widehat{K}_M)^{-1} \widehat{K}_{nM}^\top \widehat{Y}$$

requiring

$$\text{time} O(ndM \vee nM^2) \qquad \text{memory} O(nM)$$

Nyström centers and sketching

Note that Nyström corresponds to sketching

$$\widehat{X}_M = \widehat{X} S,$$

with

$$S = \widetilde{X}_M.$$

Regularization with sketching and Nyström centers

Considering regularization as we did for sketching leads to

$$\min_{c \in \mathbb{R}^M} \frac{1}{n} \|\widehat{X}\widetilde{X}_M^\top c_M - \widehat{Y}\|^2 + \lambda c_M^\top c_M, \quad \lambda > 0.$$

In the Nyström derivation we ended up with Equivalently

$$\min_{c \in \mathbb{R}^M} \frac{1}{n} \|\widehat{X}\widetilde{X}_M^\top c_M - \widehat{Y}\|^2 + \lambda c_M^\top \widetilde{X}_M \widetilde{X}_M^\top c_M, \quad \lambda > 0.$$

Different regularizers are considered.

Nyström approximation

A classical discrete approximation to integral equations.
For all x

$$\int k(x,x')c(x')dx' = y(x) \qquad \mapsto \qquad \sum_{j=1}^{M} k(x,\tilde{x}_j)c(\tilde{x}_j) = y(\tilde{x}_j)$$

Related to to quadrature methods.

From operators to matrices.
For all $i = 1,\ldots,n$

$$\sum_{j=1}^{n} k(x_i,x_j)c_j = y_j \qquad \mapsto \qquad \sum_{j=1}^{M} k(x_i,\tilde{x}_j)c_i = y_j$$

Nyström approximation and subsampling

For all $i = 1, \dots, n$

$$\sum_{j=1}^{n} k(x_i, x_j) c_j = y_j \qquad \mapsto \qquad \sum_{j=1}^{M} k(x_i, \tilde{x}_j) c_i = y_j$$

The above formulation highlights connection to columns subsampling

$$\widehat{K} c = \widehat{Y} \qquad \mapsto \qquad \widehat{K}_{nM} c_M = \widehat{Y}$$

In summary

- Projection (dim. reductions) regularizes.

- Reducing computations by sketching

- Nyström approximation and columns subsampling.