Learning problem

Solve

$$\min_w \mathcal{E}(w), \quad \mathcal{E}(w) = \int d\rho(x, y)L(w^\top x, y)$$

given \((x_1, y_1), \ldots, (x_n, y_n)\)

Beyond linear models: non-linear features and kernels
Regularization by penalization

Replace

\[ \min_w \mathcal{E}(w) \quad \text{by} \quad \min_w \hat{\mathcal{E}}(w) + \lambda \|w\|^2 \]

\[ \hat{\mathcal{E}}(w) = \frac{1}{n} \sum_{i=1}^n L(w^\top x_i, y_i) \]

\[ \lambda > 0 \quad \text{regularization parameter} \]
Early stopping regularization

Another example of regularization:

Early stopping of an iterative procedure applied to noisy data.
Gradient descent for square loss

\[ w_{t+1} = w_t - \gamma \hat{X}^\top (\hat{X} w_t - \hat{y}) \]

\[ \sum_{i=1}^{n} (y_i - w^\top x_i)^2 = \| \hat{X} w - \hat{y} \|^2 \]

- no penalty
- stepsize chosen a priori \( \gamma = \frac{2}{\| \hat{X}^\top \hat{X} \|} \)
Early stopping at work

Fitting on the training set
Iteration #1

Fitting on the training set
Iteration #2

Fitting on the training set
Iteration #7

Fitting on the training set
Iteration #5000

L. Rosasco, TUM 2016
Semi-convergence

\[
\min_w \mathcal{E}(w) \quad \text{vs} \quad \min_w \hat{\mathcal{E}}(w)
\]
Connection to Tikhonov

\[ w_{t+1} = w_t - \gamma \hat{X}^\top (\hat{X} w_t - \hat{y}) \]
\[ = (I - \gamma \hat{X}^\top \hat{X})w_t + \gamma \hat{X}^\top \hat{y} \]

by induction

\[ w_t = \gamma \sum_{j=0}^{t-1} (I - \gamma \hat{X}^\top \hat{X})^j \hat{X}^\top \hat{y} \]

\[ \text{Truncated power series} \]
Neumann series

\[ \sum_{j=0}^{t-1} \gamma (I - \gamma \hat{X}^\top \hat{X})^j \]

- \( |a| < 1 \)

\[ (1 - a)^{-1} = \sum_{j=0}^{\infty} a^j \quad \Rightarrow \quad a^{-1} = \sum_{j=0}^{\infty} (1 - a)^j \]

- \( A \in \mathbb{R}^{d \times d}, \|A\| < 1, \) invertible

\[ A^{-1} = \sum_{j=0}^{\infty} (I - A)^j \]
Stable matrix inversion

Truncated Neumann Series

$$\left(\hat{X}^\top \hat{X}\right)^{-1} = \gamma \sum_{j=0}^{\infty} (I - \gamma \hat{X}^\top \hat{X})^j \approx \gamma \sum_{j=0}^{t-1} (I - \gamma \hat{X}^\top \hat{X})^j$$

compare to

$$\left(\hat{X}^\top \hat{X}\right)^{-1} \approx \left(\hat{X}^\top \hat{X} + \lambda n I\right)^{-1}$$
Early-stopping: extensions

Early stopping regularization, so far analogous to

\[
\min_w \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|^2
\]

Extensions

Early stopping regularization analogous to

\[
\min_w \frac{1}{n} \sum_{i=1}^{n} V(w^T x_i, y_i) + \lambda \|w\|^2
\]

or

\[
\min_w \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda R(w)
\]

...or both.
Early-stopping why?

- Regularization path
- Warm-restart
- Computational regularization
Beyond Tikhonov: TSVD

\[ \hat{X}^\top \hat{X} = V \Sigma V^\top, \quad w_M = (\hat{X}^\top \hat{X})^{-1}_M \hat{X}^\top \hat{y} \]

- \((\hat{X}^\top \hat{X})^{-1}_M = V \Sigma^{-1}_M V^\top\)
- \(\Sigma^{-1}_M = \text{diag}(\sigma^{-1}_1, \ldots, \sigma^{-1}_M, 0 \ldots, 0)\)

Also known as principal component regression (PCR)
Principal component analysis (PCA)

Dimensionality reduction

\[ \hat{X}^\top \hat{X} = V \Sigma V^\top \]

Eigenfunctions are directions, of
- maximum variance
- best reconstruction

L. Rosasco, TUM 2016
TSVD and PCA

\[ TSVD \quad \Leftrightarrow \quad PCA + ERM \]

Regularization by projection
Non-linear function

\[
f(x) = \sum_{i=1}^{p} w_i \phi_i(x) = \Phi(x)^\top w
\]

with

\[
w = (\hat{\Phi}^\top \hat{\Phi})_M^{-1} \hat{\Phi}^\top \hat{y}
\]

Let \( \hat{\Phi} = (\Phi(x_1), \ldots, \Phi(x_n))^\top \in \mathbb{R}^{n \times p} \).

\[
\hat{\Phi}^\top \hat{\Phi} = V \Sigma V^\top, \quad (\hat{\Phi}^\top \hat{\Phi})_M^{-1} = V \Sigma_M^{-1} V^\top
\]

\[
\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p), \quad \Sigma_M^{-1} = \text{diag}(\sigma_1^{-1}, \ldots, \sigma_M^{-1}, 0, \ldots)
\]
TSVD/PCR with kernels

\[ f(x) = \sum_{i=1}^{n} K(x, x_i) c_i, \quad c = (\hat{K})^{-1}_M \hat{y} \]

\[ \hat{K}_{ij} = K(x_i, x_j), \quad \hat{K} = U \Sigma U^\top, \quad \Sigma = (\sigma_1, \ldots, \sigma_n), \]

\[ \hat{K}_M^{-1} = U \Sigma_M^{-1} U^\top, \quad \Sigma_M^{-1} = (\sigma_1^{-1}, \ldots, \sigma_M^{-1}, 0, \ldots), \]
Complexity of nonparametric learning

- time: $O(n^3)$ or $O(n^2T)$ or $O(n^2M)$
- space: $O(n^2)$
Bottleneck of non-linear learning with kernel methods

Memory

$\hat{K}$ is $O(n^2)$
An intuition

PCR/spectral filtering: first compute then discard.

Since we know we need only part of the information in the data:

Can we compute less?
Approaches to large scale

- (Random) features - find $\tilde{\Phi} : X \to \mathbb{R}^M$, with $M \ll n$ s.t.

\[ K(x, x') \approx \tilde{\Phi}(x)^\top \tilde{\Phi}(x') \]

- Subsampling (Nyström) - replace

\[
f(x) = \sum_{i=1}^{n} K(x, x_i)c_i \quad \text{by} \quad f(x) = \sum_{i=1}^{M} K(x, \tilde{x}_i)c_i
\]

$\tilde{x}_i$ subsampled from training set, $M$
It holds (using Fourier transform),

\[
K(x, x') = e^{-\|x-x'\|^2 \gamma} = \int \frac{d\omega e^{-\omega^2 c}}{dp(\omega)} e^{i\omega^T x} e^{-i\omega^T x'}.
\]

Consider,

\[
\tilde{K}(x, x') = \frac{1}{m} \sum_{j=1}^{M} e^{i\omega_j^T x} e^{-i\omega_j^T x'}
\]

with \(\omega_1, \ldots, \omega_M\) i.i.d. samples w.r.t. to \(p\).
Then,
\[ e^{-\|x-x'\|^2 \gamma} \approx \tilde{\Phi}(x) \top \tilde{\Phi}(x') \]

with,
\[ \tilde{\Phi}(x) = (e^{i\omega_1^T x}, \ldots, e^{i\omega_M^T x}). \]

Alternatively consider
\[ \tilde{\Phi}(x) = (\cos(\omega_1^T x + b_1), \ldots, \cos(\omega_M^T x + b_M)) \]
Other examples of random features

- **translation invariant** kernels $K(x, x') = H(x - x')$,

  $$\tilde{\Phi}(x)^j = e^{i\omega_j^\top x}, \quad \omega_j \sim \pi = \mathcal{F}(H)$$

- **infinite neural nets** kernels

  $$\tilde{\Phi}(x)^j = |\omega_j^\top x + b_j|_+, \quad (\omega_j, b_j) \sim \pi = U[S^{d+1}]$$

- **infinite dot product** kernels
  
- **homogeneous additive** kernels
  
- **group invariant** kernels
  
  - ...

**Note:** Connections with **hashing** and **sketching** techniques.
Let

\[ f(x) = w^\top \tilde{\Phi}(x) \]

with coefficients solving

\[
\min_{w \in \mathbb{R}^M} \frac{1}{n} \left\| \tilde{\Phi}_n w - \hat{y} \right\|_n^2 + \lambda \|w\|_2^2,
\]

\( \tilde{\Phi}_n \) \( n \) by \( M \) matrix with rows \( \tilde{\Phi}(x_i) \).
Complexity of learning with random features

\[
\min_{w \in \mathbb{R}^M} \frac{1}{n} \left\| \tilde{\Phi}_n w - \hat{y} \right\|_n^2 + \lambda \|w\|^2
\]

\[
\downarrow
\]

\[
(\tilde{\Phi}_n^\top \tilde{\Phi}_n + \lambda nI) w = \tilde{\Phi}_n^\top \hat{y}
\]

 Computations

- Time: \(O(n^3)\) \(\underbrace{O(nM^2)}\)
- Space: \(O(n^2)\) \(O(nM)\)
Considering,

\[ f(x) = \sum_{j=1}^{d} \cos(\omega_j^\top x + b_j)w_j \]

or more generally,

\[ f(x) = \sum_{j=1}^{d} q(x, \omega_j)w_j. \]

with

- \( w_j \) optimized
- \( \omega_j \) randomized \textbf{independently of data}

What about data dependent sampling?
N"ystrom methods

\[ f(x) = \sum_{i=1}^{n} K(x, x_i) c_i \quad \rightarrow \quad f(x) = \sum_{i=1}^{M} K(x, \tilde{x}_i) c_i \]

\(\tilde{x}_i\) centers subsampled from training set \(M\)

**Note:** keep all data! (just use fewer to parameterize functions)
Nöyström ridge regression

$$\min_{c \in \mathbb{R}^M} \left\| \hat{K}_{n,M} c - \hat{y} \right\|_n^2 + \lambda c^\top \hat{K}_{M,M} c$$

- \((\hat{K}_{n,M})_{i,j} = K(\tilde{x}_i, x_j)\)
- \((\hat{K}_{M,M})_{i,j} = K(\tilde{x}_i, \tilde{x}_j)\)
Complexity of N"ystrom ridge regression

\[
\min_{c \in \mathbb{R}^M} \frac{1}{n} \left\| \hat{K}_{M,n} c - \hat{y} \right\|_n^2 + \lambda c^\top \hat{K}_{M,M} c
\]

\[
\begin{align*}
\downarrow \\
(K_{n,M}^\top K_{n,M} + \lambda n K_{M,M}) c &= K_{n,M}^\top \hat{y}
\end{align*}
\]

Computations

- Time: $O(n^3)$ $O(nM^2)$
- Space: $O(n^2)$ $O(nM)$
Subsampling and regularization

- Random features

\[ f(x) = \sum_{i=1}^{M} q(x, \omega_i) w_i \]

- Nyström

\[ f(x) = \sum_{i=1}^{M} K(x, \tilde{x}_i) c_i \]
Subsampling as stochastic regularization

The subsampling level $M$ can be seen as a regularization parameter!

$M$ controls: statistics, space and time complexity!
An incrementation approach

Algorithm

1. Pick a center + compute solution
2. Pick another center + rank one update
3. Pick another center . . .
Computational regularization idea:

*use computations to regularize*

Iterative and subsampling regularization can be seen as instances.
Approaches to large scale non-linear learning

Consider,

\[ f(x) = \sum_{j=1}^{d} Q(x, \omega_j) w_j. \]

with \( Q \) feature or kernel and

- \( w_j \) optimized,
- \( \omega_j \) randomized.
Shallow nets

Consider,

\[ f(x) = \sum_{j=1}^{d} Q(x, \omega_j) w_j. \]

This is a one layer neural net!

Neural nets

- \( w_j \) optimized
- \( \omega_j \) randomized optimized

L. Rosasco, TUM 2016
From shallow to deep nets

Shallow nets

\[ f(x) = w^\top \Phi_W(x), \quad \Phi_W(x) = Q(W^\top x) \]

\( Q \) activation function.

Deep nets

\[ f(x) = w^\top \bar{\Phi}(x), \quad \bar{\Phi} = \Phi_{W_L} \circ \cdots \circ \Phi_{W_1} \quad \Phi_{W_j} = Q(W_j^\top x) \]
Deep nets

\[ f(x) = w^\top \bar{\Phi}(x), \quad \bar{\Phi} = \Phi_{W_L} \circ \cdots \circ \Phi_{W_1} \quad \Phi_{W_j} = Q(W_j^\top x) \]

Neural nets

- \( w_j, W_j \) optimized,
- learning data representation (?)
This class

- early stopping
- projection regularization
- subsampling & regularization
Next class

▶ a practical experience


Ali Rahimi and Benjamin Recht.  
Random features for large-scale kernel machines.  

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco.  
Less is more: Nyström computational regularization.  

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco.  
Generalization properties of learning with random features.  

Alex J Smola and Bernhard Schölkopf.  
Sparse greedy matrix approximation for machine learning.  
2000.