Machine Learning

Regularization and Feature Selection

Fabio Vandin November 13th, 2023

Learning Model

- A: learning algorithm for a machine learning task
- S: m i.i.d. pairs $z_i = (x_i, y_i), i = 1, \dots, m$, with $z_i \in Z = \mathcal{X} \times Y$, generated from distribution $\mathcal{D}$ $\Rightarrow$ training set available to A to produce $A(S)$;
- $\mathcal{H}$: the hypothesis (or model) set for A
- loss function: $\ell(h, (x, y))$, $\ell : \mathcal{H} \times Z \to \mathbb{R}^+$
- $L_S(h)$: empirical risk or training error of hypothesis $h \in \mathcal{H}$

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i)$$

- $L_{\mathcal{D}}(h)$: true risk or generalization error of hypothesis $h \in \mathcal{H}$:

$$L_{\mathcal{D}}(h) = \mathbb{E}_{z \in \mathcal{D}}[\ell(h, z)]$$

Learning Paradigms

We would like A to produce $A(S)$ such that $L_{\mathcal{D}}(A(S))$ is *small*, or at least close to the smallest generalization error $L_{\mathcal{D}}(h^*)$ achievable by the "best" hypothesis h^* in $\mathcal{H}$:

$$h^* = \arg\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$$

We have seen a *learning paradigm*: Empirical Risk Minimization

We will now see another learning paradigm...

Regularized Loss Minimization

Assume h is defined by a vector $\mathbf{w} = (w_1, \ldots, w_d)^T \in \mathbb{R}^d$ (e.g., linear models)

Regularization function $R : \mathbb{R}^d \rightarrow \mathbb{R}$

Regularized Loss Minimization (RLM): pick h obtained as

$$\arg\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + R(\mathbf{w}) \right)$$

Intuition: $R(\mathbf{w})$ is a "measure of complexity" of hypothesis h defined by $\mathbf{w}$
$\Rightarrow$ regularization balances between low empirical risk and "less complex" hypotheses

We will see some of the most common regularization function

ℓ_1 Regularization

Regularization function: $R(\mathbf{w}) = \lambda||\mathbf{w}||_1$

- $\lambda \in \mathbb{R}, \lambda > 0$
- ℓ_1 norm: $||\mathbf{w}||_1 = \sum_{i=1}^{d} |w_i|$

Therefore the *learning rule* is: pick

$$A(S) = \arg\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda||\mathbf{w}||_1\right)$$

Intuition:

- $||\mathbf{w}||_1$ measures the "complexity" of hypothesis defined by $\mathbf{w}$
- λ regulates the tradeoff between the empirical risk ($L_S(\mathbf{w})$) or overfitting and the complexity ($||\mathbf{w}||_1$) of the model we pick

LASSO

Linear regression with squared loss + ℓ_1 regularization $\Rightarrow$ *LASSO* (*least absolute shrinkage and selection operator*)

LASSO: pick

$$\mathbf{w} = \arg\min_{\mathbf{w}} \lambda ||\mathbf{w}||_1 + \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2$$

How?

Notes:

- no closed form solution!
- ℓ_1 norm is a convex function and squared loss is convex $\Rightarrow$ problem can be solved efficiently! (true for every convex loss function)

LASSO and Sparse Solutions: Example

(Equivalent) one dimensional regression problem with squared loss:

$$\arg\min_{w\in\mathbb{R}} \left(\frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda |w| \right)$$

Is equivalent to:

$$\arg\min_{w\in\mathbb{R}} \left(\frac{1}{2} \left(\frac{1}{m} \sum_{i=1}^{m} x_i^2 \right) w^2 - \left(\frac{1}{m} \sum_{i=1}^{m} x_i y_i \right) w + \lambda |w| \right)$$

Assume for simplicity that $\frac{1}{m}\sum_{i=1}^{m} x_i^2 = 1$, and let $\sum_{i=1}^{m} x_i y_i = \langle \mathbf{x}, \mathbf{y} \rangle$.

Then the optimal solution is

$$w = \mathrm{sign}(\langle \mathbf{x}, \mathbf{y} \rangle)[\langle \mathbf{x}, \mathbf{y} \rangle / m - \lambda]_+$$

where $[a]_+ =^{(def)} \max\{a, 0\}$.

Tikhonov regularization

Regularization function: $R(\mathbf{w}) = \lambda||\mathbf{w}||^2$

- $\lambda \in \mathbb{R}, \lambda > 0$
- ℓ_2 norm: $||\mathbf{w}||^2 = \sum_{i=1}^{d} w_i^2$

Therefore the *learning rule* is: pick

$$A(S) = \arg\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda||\mathbf{w}||^2 \right)$$

Intuition:

- $||\mathbf{w}||^2$ measures the “complexity” of hypothesis defined by $\mathbf{w}$
- λ regulates the tradeoff between the empirical risk ($L_S(\mathbf{w})$) or overfitting and the complexity ($||\mathbf{w}||^2$) of the model we pick

Ridge Regression

Linear regression with squared loss + Tikhonov regularization
$\Rightarrow$ *ridge regression*

Linear regression with squared loss:

- **given**: training set $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$
- **want**: $\mathbf{w}$ which minimizes empirical risk:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{m} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2$$

equivalently, find $\mathbf{w}$ which minimizes the *residual sum of squares* $RSS(\mathbf{w})$

$$\mathbf{w} = \arg\min_{\mathbf{w}} RSS(\mathbf{w}) = \arg\min_{\mathbf{w}} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2$$

Linear regression: pick

$$\mathbf{w} = \arg\min_{\mathbf{w}} RSS(\mathbf{w}) = \arg\min_{\mathbf{w}} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2$$

Ridge regression: pick

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left(\lambda ||\mathbf{w}||^2 + \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2 \right)$$

RSS: Matrix Form

Let

$$\mathbf{X} = \begin{bmatrix} \cdots & \mathbf{x}_1 & \cdots \\ \cdots & \mathbf{x}_2 & \cdots \\ \cdots & \vdots & \cdots \\ \cdots & \mathbf{x}_m & \cdots \end{bmatrix}$$

X: *design matrix*

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

⇒ we have that RSS is

$$\sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

Ridge Regression: Matrix Form

Linear regression: pick

$$\arg\min_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

Ridge regression: pick

$$\arg\min_{\mathbf{w}} \left(\lambda ||\mathbf{w}||^2 + (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) \right)$$

Want to find $\mathbf{w}$ which minimizes $f(\mathbf{w}) = \lambda||\mathbf{w}||^2 + (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})$.

How?

Compute gradient $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}}$ of objective function w.r.t $\mathbf{w}$ and compare it to 0.

$$\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} = 2\lambda\mathbf{w} - 2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$

Then we need to find $\mathbf{w}$ such that

$$2\lambda\mathbf{w} - 2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

$$2\lambda \mathbf{w} - 2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

is equivalent to

$$\left(\lambda \mathbf{I} + \mathbf{X}^T\mathbf{X}\right)\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

Note:

- $\mathbf{X}^T\mathbf{X}$ is positive semidefinite
- $\lambda \mathbf{I}$ is positive definite

$\Rightarrow \lambda \mathbf{I} + \mathbf{X}^T\mathbf{X}$ is positive definite
$\Rightarrow \lambda \mathbf{I} + \mathbf{X}^T\mathbf{X}$ is invertible

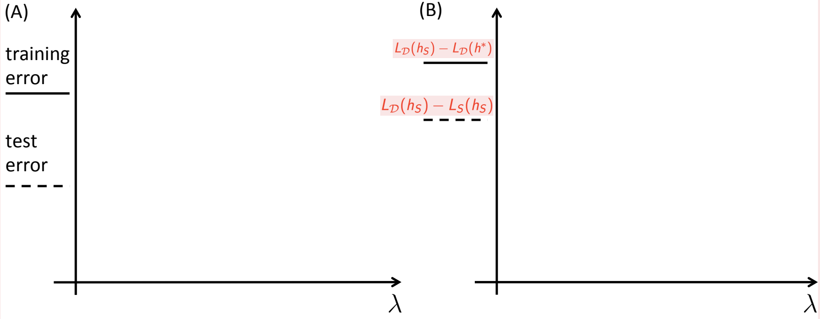
Ridge regression solution:
$\mathbf{w} = \left(\lambda \mathbf{I} + \mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$

Exercise 5

Consider the ridge regression problem $\arg\min_{\mathbf{w}} \lambda||\mathbf{w}||^2 + \sum_{i=1}^{m}(\langle \mathbf{w}, \mathbf{x_i} \rangle - y_i)^2$. Let: h_S be the hypothesis obtained by ridge regression on with training set S; h^* be the hypothesis of minimum generalization error among all linear models.

(A) Draw, in the plot below, a *typical* behaviour of (i) *the training error* and (ii) *the test/generalization error* of h_S as a function of λ.

(B) Draw, in the plot below, a *typical* behaviour of (i) $L_{\mathcal{D}}(h_S) - L_{\mathcal{D}}(h^*)$ and (ii) $L_{\mathcal{D}}(h_S) - L_S(h_S)$ as a function of λ.



Feature Selection

In general, in machine learning one has to decide what to use as features ($=$ input) for learning.

Even if somebody gives us a representation as a feature vector, maybe there is a “better” representation?

What is “better”?

Example

- features x_1, x_2, output y
- $x_1 \sim Uniform(-1, 1)$
- $y = x_1^2$
- $x_2 \sim y + Uniform(-0.01, 0.01)$

If we want to predict y, which feature is better: x_1 or x_2?

No-free lunch...

Feature Selection: Scenario

We have a large pool of features

Goal: select a small number of features that will be used by our (final) predictor

Assume $\mathcal{X} = \mathbb{R}^d$.

Goal: learn (final) predictor using $k << d$ predictors

Motivation?

- prevent overfitting: less predictors $\Rightarrow$ hypotheses of lower complexity!
- predictions can be done faster
- useful in many applications!

Feature Selection: Computational Problem

Assume that we use the Empirical Risk Minimization (ERM) procedure.

The problem of selecting k features that minimize the empirical risk can be written as:

$$\min_{\mathbf{w}} L_S(\mathbf{w}) \text{ subject to } ||\mathbf{w}||_0 \leq k$$

where $||\mathbf{w}||_0 = |\{i : w_i \neq 0\}|$

How can we solve it?

Subset Selection

How do we find the solution to the problem below?

$$\min_{\mathbf{w}} L_S(\mathbf{w}) \;\; \text{subject to} \;\; ||\mathbf{w}||_0 \leq k$$

Note: the solution will always include k features

Let:

- $\mathcal{I} = \{1, \ldots, d\}$;
- given $p = \{i_1, \ldots, i_k\} \subseteq \mathcal{I}$: $\mathcal{H}_p$ = hypotheses/models where only features $w_{i_1}, w_{i_2} \ldots, w_{i_k}$ are used

```
$P^{(k)} \leftarrow \{J \subseteq \mathcal{I} : |J| = k\}$;
foreach $p \in P^{(k)}$ do
    $h_p \leftarrow \arg\min_{h \in \mathcal{H}_p} L_S(h)$;
return $h^{(k)} \leftarrow \arg\min_{p \in P^{(k)}} L_S(h_p)$;
```

Complexity? Learn $\Theta\left(\binom{d}{k}\right) \in \Theta\left(d^k\right)$ models $\Rightarrow$ exponential algorithm!

Can we do better?

Proposition

The optimization problem of feature selection NP-hard.

What can we do?

Heuristic solution $\Rightarrow$ greedy algorithms

Greedy Algorithms for Feature Selection

Forward Selection: start from the empty solution, add one feature at the time, until solution has cardinality k

```
$sol \leftarrow \emptyset$;
while $|sol| < k$ do
    foreach $i \in \mathcal{I} \setminus sol$ do
        $p \leftarrow sol \cup \{i\}$;
        $h_p \leftarrow \arg\min_{h \in \mathcal{H}_p} L_S(h)$;
    $sol \leftarrow sol \cup \arg\min_{i \in \mathcal{I} \setminus sol} L_S(h_{sol \cup \{i\}})$;
return $sol$;
```

Complexity? Learns $\Theta(kd)$ models

Backward Selection: start from the solution which includes all features, remove one feature at the time, until solution has cardinality k

Pseudocode: analogous to forward selection [Exercize!]

Complexity? Learns $\Theta\left((d-k)d\right)$ models

Notes

We have used only training set to select the best hypothesis...

$\Rightarrow$ we may overfit!

Solution? Use validation! (or cross-validation)

Split data into training data and validation data, learn models on training, evaluate (= pick among different hypothesis models) on validation data. Algorithms are similar.

Note: now the best model (in terms of validation error) may include less than k features!

Subset Selection with Validation Data

S = training data (from data split)
V = validation data (from data split)

Using training and validation:

```
for $\ell \leftarrow 0$ to $k$ do
    $P^{(\ell)} \leftarrow \{J \subseteq \mathcal{I} : |J| = \ell\}$;
    foreach $p \in P^{(\ell)}$ do
        $h_p \leftarrow \arg\min_{h \in \mathcal{H}_p} L_S(h)$;
    $h^{(\ell)} \leftarrow \arg\min_{p \in P^{(\ell)}} L_V(h_p)$;
return $\arg\min_{h \in \{h^{(0)}, h^{(1)}, \ldots, h^{(k)}\}} L_V(h)$
```

Forward Selection with Validation Data

Using training and validation:

```
$sol \leftarrow \emptyset$;
while $|sol| < k$ do
    foreach $i \in \mathcal{I} \setminus sol$ do
        $p \leftarrow sol \cup \{i\}$;
        $h_p \leftarrow \arg\min_{h \in \mathcal{H}_p} L_S(h)$;
    $sol \leftarrow sol \cup \arg\min_{i \in \mathcal{I} \setminus sol} L_V(h_{sol \cup \{i\}})$;
return $sol$;
```

Backward Selection with validation: similar [Exercize]

Similar approach for all algorithms with cross-validation [Exercize]

Bibliography [UML]

Regularization and Ridge Regression: Chapter 12

- no Section 13.3;
- Section 13.4 only up to Corollary 13.8 (excluded)

Feature Selection and LASSO: Chapter 25

- only Section 25.1.2 (introduction and "Backward Elimination") and 25.1.3