Machine Learning

Regularization and Feature Selection

Fabio Vandin

November 25th, 2022

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

(features)

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

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:

[assumption: an hypothesis $h \in \mathcal{H}$ corresponds to a vector $\vec{w} \in \mathbb{R}^d$]

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

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

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?

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

Solution 1 : i) enumerate all subsets of features with $\overset{=}{\cancel{\leq}}\, k$ elements $\neq 0$

ii) learn the best model (minimizes the training error) for each of the subsets

iii) keep the subset of minimum error

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**

$\quad h_p \leftarrow \arg\min_{h \in \mathcal{H}_p} L_S(h)$;

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!

$\approx$ it is unlikely that there is a poly-time algorithm to solve problem (exactly)

Can we do better?

Proposition

The optimization problem of feature selection NP-hard.

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)

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