Large scale greedy feature-selection for multi-target learning

Antti Airola, Tapio Pahikkala et al.

ECML 2015 BigTargets Workshop

Overview

Joint work with many authors

- University of Turku: Antti Airola, Pekka Naula, Tapio Pahikkala, Tapio Salakoski (Multi-target greedy RLS)

Overview

- Large scale feature selection for multi-target learning
- Task: select minimal set of common features allowing accurate predictions over target tasks
- Greedy RLS: greedy regularized least-squares
- Linear time (#inputs, #features, #outputs, #selected)
- Highlights from experiments
  - Broad-DREAM Gene Essentiality Prediction Challenge
  - Outperforms multi-task Lasso for small feature budgets
- Also scales to full Genome Wide Association Studies; thousands of samples, hundreds of thousands of features (recent PhD thesis: Sebastian Okser)

Motivation

Why feature selection?

1. Accuracy: regularizing effect, avoiding overfitting leads to better generalization
2. Interpretability: obtain a small set of features understandable by human expert
3. Budget constraints: obtaining features costs time and money

Model sparsity

$$\mathbf{W}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix}, \qquad \mathbf{W}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2 & 3 & -1 & 2 \\ 0 & 0 & 0 & 0 \\ 3 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- features x targets coefficient matrices
- $\mathbf{W}_1$ 8 features needed for prediction
- $\mathbf{W}_2$ 2 features needed for prediction

Learning task

Least-squares formulation

$$\arg\min_{\mathbf{W}\in\mathbb{R}^{d\times t}} \|\mathbf{X}\mathbf{W}-\mathbf{Y}\|_F^2$$
$$\text{subject to } C(\mathbf{W})$$

Notation

$\mathbf{X}$	data matrix
$\mathbf{Y}$	output matrix
$\mathbf{W}$	model coefficients
$\|\cdot\|_F$	Frobenius norm
$C(\cdot)$	Constraint (regularizer)

Multi-task Lasso (baseline)

Multi-task Lasso (Zhang, 2006)

$$\arg\min_{\mathbf{W}\in\mathbb{R}^{d\times t}} \|\mathbf{X}\mathbf{W}-\mathbf{Y}\|_F^2$$
$$\text{subject to } \sum_{i=1}^{d} \max_j |\mathbf{W}_{i,j}| \leq r$$

- $L_{1,\infty}$ norm enforces sparsity in the number of features
- $r > 0$ regularization parameter

Greedy RLS

Greedy RLS (proposed)

$$\arg\min_{\mathbf{W}\in\mathbb{R}^{d\times t}} \|\mathbf{X}\mathbf{W}-\mathbf{Y}\|_F^2$$
$$\text{subject to } \|\mathbf{W}\|_F^2 < r \qquad \text{and}$$
$$|\{i \mid \exists j, \mathbf{W}_{i,j} \neq 0\}| \leq k$$

- $r > 0$ regularization parameter
- $k > 0$ constraint on the number of features
- heuristics needed to search over the power set of features

Greedy RLS

- Greedy regularized least-squares (Greedy RLS)
- Starting from empty feature set, at each point add the feature reducing leave-one-out cross-validation error most
- Stop once k features have been selected

Greedy RLS

Algorithm 1 Multi-target greedy RLS

$\mathcal{S} \leftarrow \emptyset$ ▷ selected features common for all tasks
while $|\mathcal{S}| < k$ **do** ▷ select k features
  $e \leftarrow \infty$
  $b \leftarrow 0$
  for $i \in \{1, \dots, d\} \setminus \mathcal{S}$ **do** ▷ test all features
    $e_{avg} \leftarrow 0$
    for $j \in \{1, \dots, t\}$ **do**
      $e_{i,j} \leftarrow \mathcal{L}(\mathbf{X}_{:,\mathcal{S}\cup\{i\}}, \mathbf{Y}_{:,j})$ ▷ LOO for task j
      $e_{avg} \leftarrow e_{avg} + e_{i,j}/t$
    if $e_{avg} < e$ **then**
      $e \leftarrow e_{avg}$
      $b \leftarrow i$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{b\}$ ▷ feature with lowest LOO-error
$\mathbf{W} \leftarrow \mathcal{A}(\mathbf{X}_{:,\mathcal{S}}, \mathbf{Y})$ ▷ train final models
return $\mathbf{W}$, $\mathcal{S}$

- Greedy RLS could be implemented as a general wrapper code calling a black-box solver
- #selected×#features×#targets×#CV-rounds calls for naive implementation!
- Matrix algebraic optimization for feature addition, leave-one-out... (for all targets simultaneously)
- Linear time algorithm (#inputs, #features, #outputs, #selected)
- P. Naula, A. Airola, T. Salakoski and T. Pahikkala. Multi-label learning under feature extraction budgets. *Pattern Recognition Letters*, 2014.

Greedy RLS

Algorithm 2 Multi-target greedy RLS

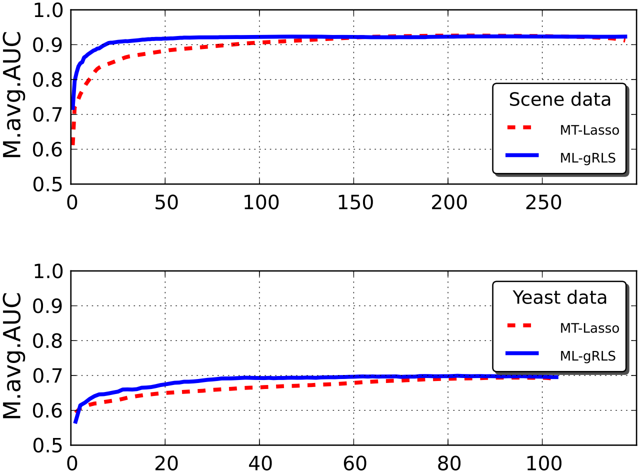
$\mathbf{A} \leftarrow \lambda^{-1}\mathbf{Y}$
$\mathbf{g} \leftarrow \lambda^{-1}\mathbf{1}$
$\mathbf{C} \leftarrow \lambda^{-1}\mathbf{X}$
$\mathcal{S} \leftarrow \emptyset$
while $|\mathcal{S}| < k$ **do**
  $e \leftarrow \infty$
  $b \leftarrow 0$
  for $i \in \{1, \ldots, d\} \setminus \mathcal{S}$ **do**
    $\mathbf{u} \leftarrow \mathbf{C}_{:,i}(1 + (\mathbf{X}_{:,i})^\mathsf{T}\mathbf{C}_{:,i})^{-1}$
    $e_i \leftarrow 0$
    $\widetilde{\mathbf{A}} \leftarrow \mathbf{A} - \mathbf{u}((\mathbf{X}_{:,i})^\mathsf{T}\mathbf{A})$
    for $h \in \{1, \ldots, t\}$ **do**
      for $j \in \{1, \ldots, n\}$ **do**
        $\tilde{\mathbf{g}}_j \leftarrow \mathbf{g}_j - \mathbf{u}_j\mathbf{C}_{j,i}$
        $e_i \leftarrow e_i + (\tilde{\mathbf{g}}_j)^{-2}(\widetilde{\mathbf{A}}_{j,h})^2$
    if $e_i < e$ **then**
      $e \leftarrow e_i$
      $b \leftarrow i$
  $\mathbf{u} \leftarrow \mathbf{C}_{:,b}(1 + (\mathbf{X}_{:,b})^\mathsf{T}\mathbf{C}_{:,b})^{-1}$
  $\mathbf{A} \leftarrow \mathbf{A} - \mathbf{u}((\mathbf{X}_{:,b})^\mathsf{T}\mathbf{A})$
  for $j \in \{1, \ldots, n\}$ **do**
    $\mathbf{g}_j \leftarrow \mathbf{g}_j - \mathbf{u}_j\mathbf{C}_{j,b}$
  $\mathbf{C} \leftarrow \mathbf{C} - \mathbf{u}((\mathbf{X}_{:,b})^\mathsf{T}\mathbf{C})$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{b\}$
$\mathbf{W} \leftarrow (\mathbf{X}_{:,\mathcal{S}})^\mathsf{T}\mathbf{A}$

Benchmarking greedy RLS and multi-task Lasso

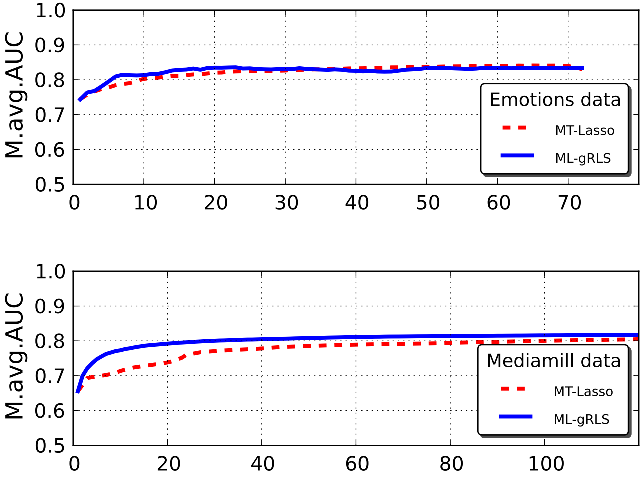
Table: Mulan datasets (Tsoumakas et al. 2011).

Data sets	domain	labels	features	instances
Scene	image	6	294	2407
Yeast	biology	14	103	2417
Emotions	music	6	72	593
Mediamill*	text	9	120	41583
Delicious	text	983	500	16105
Tmc2007	text	22	49060	28596

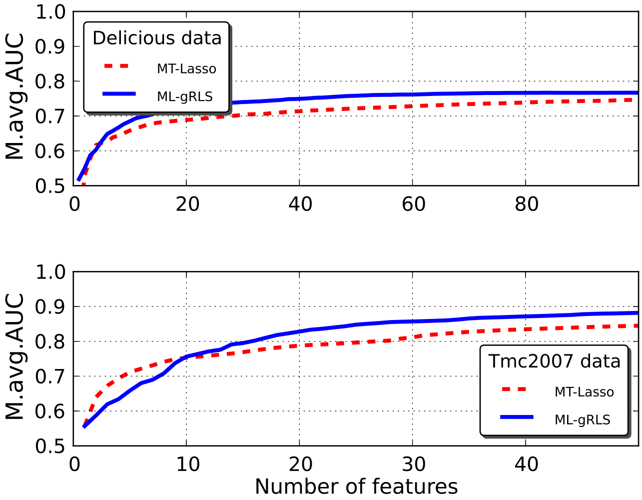
Greedy RLS vs. Lasso



Greedy RLS vs. Lasso



Greedy RLS vs. Lasso



Conclusion

- Greedy RLS: linear time algorithm for (multi-target) feature selection
- Selects joint features for the target tasks
- Competitive, when number of features to be selected small
- Applications on Genome-Wide Association Studies
- RLScore open source implementation at `https://github.com/aatapa/RLScore`