Tree based ensemble models regularization by convex optimization

Bertrand Cornélusse
Pierre Geurts
Louis Wehenkel
http://www.montefiore.ulg.ac.be/∼cornelusse

Systems and Modeling,
Department of EE & CS

December 12, 2009
Optimization for Machine Learning
Whistler – Canada

Motivation

Standard **supervised learning regression**:

- infer $f(\cdot) : \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}$ from a sample $\{(x_i, y_i)\}_{i=1}^{n}$.

Additional information is sometimes available, for example:

- **Censored data** – for samples $i \in \{n+1, ..., n+c\}$ the output is right censored, we want to regularize f such that $f(x_i) \geq y_i$.
- **Unlabeled observations** – a (typically large) number of unlabeled input points $\{x_i\}_{i=n+1}^{n+u}$ are known, we want to exploit regularity assumptions about the input-output relation to bias the learning of f.

We propose a way to incorporate this information in tree based ensemble methods.

Motivation

Standard **supervised learning regression**:

- infer $f(\cdot) : \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}$ from a sample $\{(x_i, y_i)\}_{i=1}^{n}$.

Additional information is sometimes available, for example:

- **Censored data** – for samples $i \in \{n+1, ..., n+c\}$ the output is right censored, we want to regularize f such that $f(x_i) \geq y_i$.
- **Unlabeled observations** – a (typically large) number of unlabeled input points $\{x_i\}_{i=n+1}^{n+u}$ are known, we want to exploit regularity assumptions about the input-output relation to bias the learning of f.

We propose a way to incorporate this information in tree based ensemble methods.

Motivation

Standard **supervised learning regression**:

- infer $f(\cdot) : \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}$ from a sample $\{(x_i, y_i)\}_{i=1}^{n}$.

Additional information is sometimes available, for example:

- **Censored data** – for samples $i \in \{n+1, ..., n+c\}$ the output is right censored, we want to regularize f such that $f(x_i) \geq y_i$.
- **Unlabeled observations** – a (typically large) number of unlabeled input points $\{x_i\}_{i=n+1}^{n+u}$ are known, we want to exploit regularity assumptions about the input-output relation to bias the learning of f.

We propose a way to incorporate this information in tree based ensemble methods.

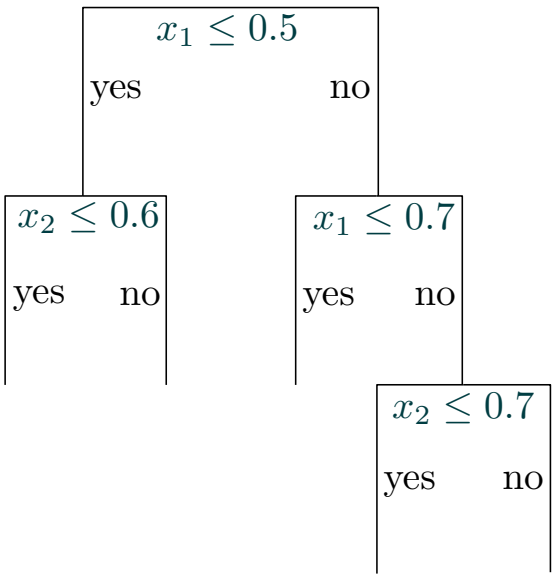
Outline

1. Kernel view of tree based ensemble methods
   - Single regression tree
   - Ensemble of regression trees
2. Regularization of a tree ensemble model
3. Applications
   - Censored data
   - Semi-supervised learning
4. Conclusion & further work

A single regression tree (quadratic loss)

Top-down **tree growing** by greedy recursive partitioning:

- reduce empirical loss as quickly as possible.



Leaves labels:

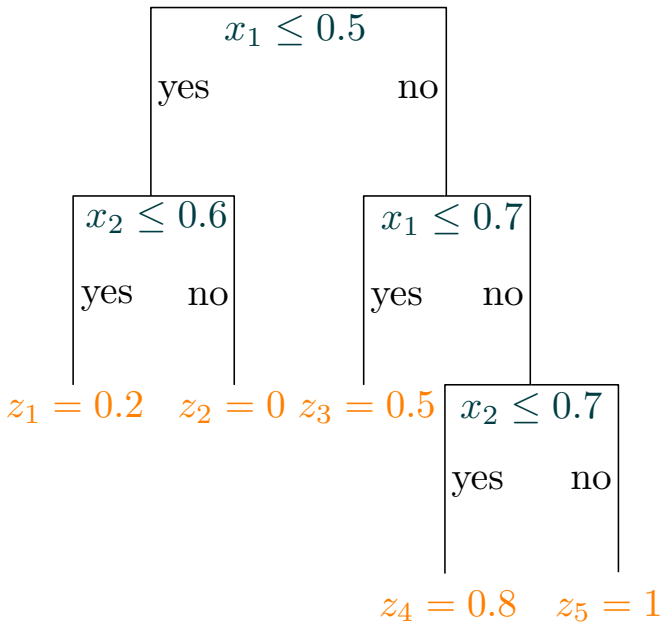
- l is the number of leaves in the tree,
- $l_j(x) = \begin{cases} 1 & \text{if } x \text{ reaches leaf } j, \\ 0 & \text{otherwise,} \end{cases}$
- $n_j = \sum_{i=1}^{n} l_j(x_i)$ is the number of objects reaching leaf j,
- the labels minimizing the loss are

$$z_j = \frac{\sum_{i=1}^{n} y_i l_j(x_i)}{n_j}.$$

A single regression tree (quadratic loss)

Top-down **tree growing** by greedy recursive partitioning:

- reduce empirical loss as quickly as possible.

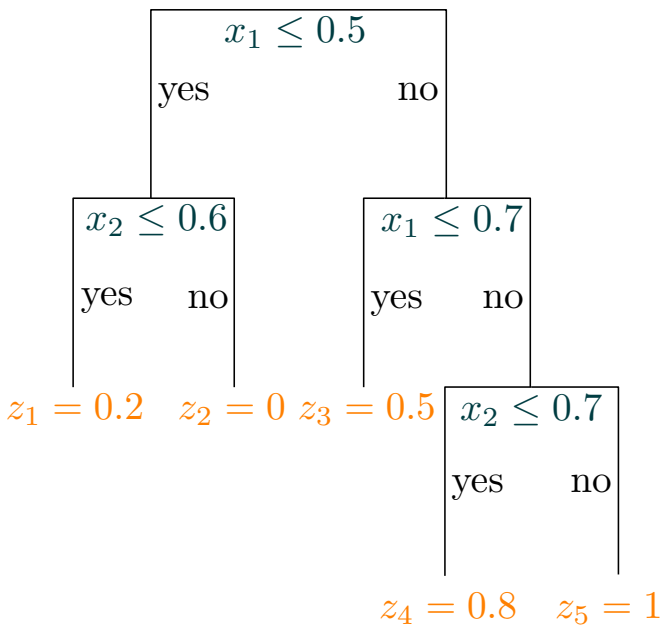


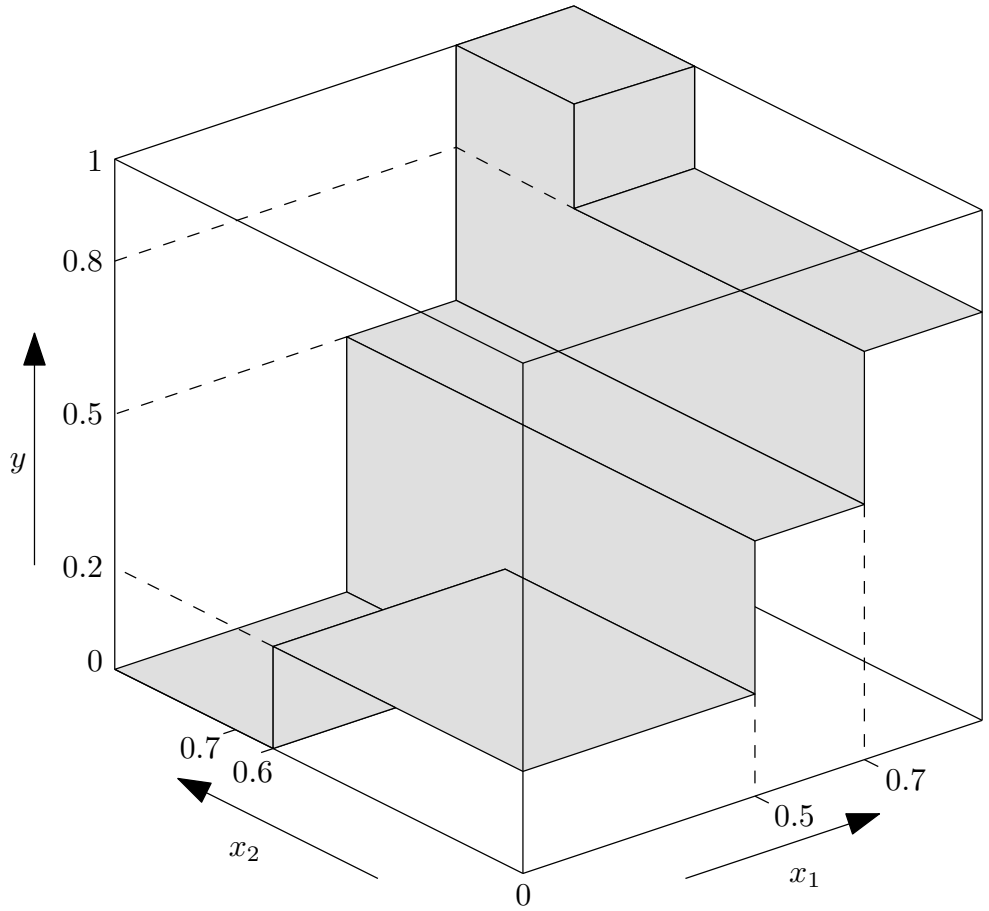
Leaves labels:

- l is the number of leaves in the tree,
- $l_j(x) = \begin{cases} 1 & \text{if } x \text{ reaches leaf } j, \\ 0 & \text{otherwise,} \end{cases}$
- $n_j = \sum_{i=1}^{n} l_j(x_i)$ is the number of objects reaching leaf j,
- the labels minimizing the loss are

$$z_j = \frac{\sum_{i=1}^{n} y_i l_j(x_i)}{n_j}.$$

A tree assigns constant values to regions of $\mathcal{X}$





A tree structure defines a kernel

- The prediction of the tree is

$$f(x) = \sum_{j=1}^{l} z_j l_j(x).$$

- An alternative view is that a tree structure defines a mapping $\phi(\cdot) : \mathcal{X} \to \mathbb{R}^l$,

$$\phi(x) = \left(\frac{l_1(x)}{\sqrt{n_1}}, \ldots, \frac{l_l(x)}{\sqrt{n_l}} \right)^T,$$

- and if $k(x_i, x) \doteq \phi^T(x_i)\phi(x)$ the prediction is

$$f(x) = \sum_{i=1}^{n} y_i k(x_i, x)$$

A tree structure defines a kernel

- The prediction of the tree is

$$f(x) = \sum_{j=1}^{l} z_j l_j(x).$$

- An alternative view is that a tree structure defines a mapping $\phi(\cdot) : \mathcal{X} \to \mathbb{R}^l$,

$$\phi(x) = \left(\frac{l_1(x)}{\sqrt{n_1}}, \ldots, \frac{l_l(x)}{\sqrt{n_l}} \right)^T,$$

- and if $k(x_i, x) \doteq \phi^T(x_i)\phi(x)$ the prediction is

$$f(x) = \sum_{i=1}^{n} y_i k(x_i, x)$$

A tree structure defines a kernel

- The prediction of the tree is

$$f(x) = \sum_{j=1}^{l} z_j l_j(x).$$

- An alternative view is that a tree structure defines a mapping $\phi(\cdot) : \mathcal{X} \to \mathbb{R}^l$,

$$\phi(x) = \left(\frac{l_1(x)}{\sqrt{n_1}}, \ldots, \frac{l_l(x)}{\sqrt{n_l}} \right)^T,$$

- and if $k(x_i, x) \doteq \phi^T(x_i)\phi(x)$ the prediction is

$$f(x) = \sum_{i=1}^{n} y_i k(x_i, x)$$

Ensemble of regression trees

Grow M perturbed trees and combine their outputs:

- improves accuracy by decreasing variance,
- destroys interpretability.

The ensemble defines the mapping $\Phi(\cdot) : \mathcal{X} \to \mathbb{R}^p$,

$$\Phi(x) = \left(w_1 \phi_1^T(x), \ldots, w_M \phi_M^T(x)\right)^T \in \mathbb{R}^p,$$

with $w_i \geq 0 \; \forall i \in \{1, \ldots, M\} : \sum_{i=1}^M w_i = 1$.
Then with $K(x_i, x) \doteq \Phi(x_i)^T \Phi(x)$,

$$\begin{aligned} f(x) &= \sum_{i=1}^n y_i K(x_i, x) \\ &= z^T \Phi(x), \end{aligned}$$

with $z = (\sqrt{n_1^1} z_1^1, \ldots, \sqrt{n_{l_M}^M} z_{l_M}^M)^T \in \mathbb{R}^p$.

Ensemble of regression trees

Grow M perturbed trees and combine their outputs:

- improves accuracy by decreasing variance,
- destroys interpretability.

The ensemble defines the mapping $\Phi(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^p$,

$$\Phi(x) = \left(w_1 \phi_1^T(x), \ldots, w_M \phi_M^T(x)\right)^T \in \mathbb{R}^p,$$

with $w_i \geq 0 \; \forall i \in \{1, \ldots, M\} : \sum_{i=1}^M w_i = 1$.

Then with $K(x_i, x) \doteq \Phi(x_i)^T \Phi(x)$,

$$\begin{aligned} f(x) &= \sum_{i=1}^n y_i K(x_i, x) \\ &= z^T \Phi(x), \end{aligned}$$

with $z = (\sqrt{n_1^1} z_1^1, \ldots, \sqrt{n_{l_M}^M} z_{l_M}^M)^T \in \mathbb{R}^p$.

Ensemble of regression trees

Grow M perturbed trees and combine their outputs:

- improves accuracy by decreasing variance,
- destroys interpretability.

The ensemble defines the mapping $\Phi(\cdot) : \mathcal{X} \to \mathbb{R}^p$,

$$\Phi(x) = \left(w_1 \phi_1^T(x), \ldots, w_M \phi_M^T(x)\right)^T \in \mathbb{R}^p,$$

with $w_i \geq 0 \; \forall i \in \{1, \ldots, M\} : \sum_{i=1}^M w_i = 1$.
Then with $K(x_i, x) \doteq \Phi(x_i)^T \Phi(x)$,

$$\begin{aligned} f(x) &= \sum_{i=1}^n y_i K(x_i, x) \\ &= z^T \Phi(x), \end{aligned}$$

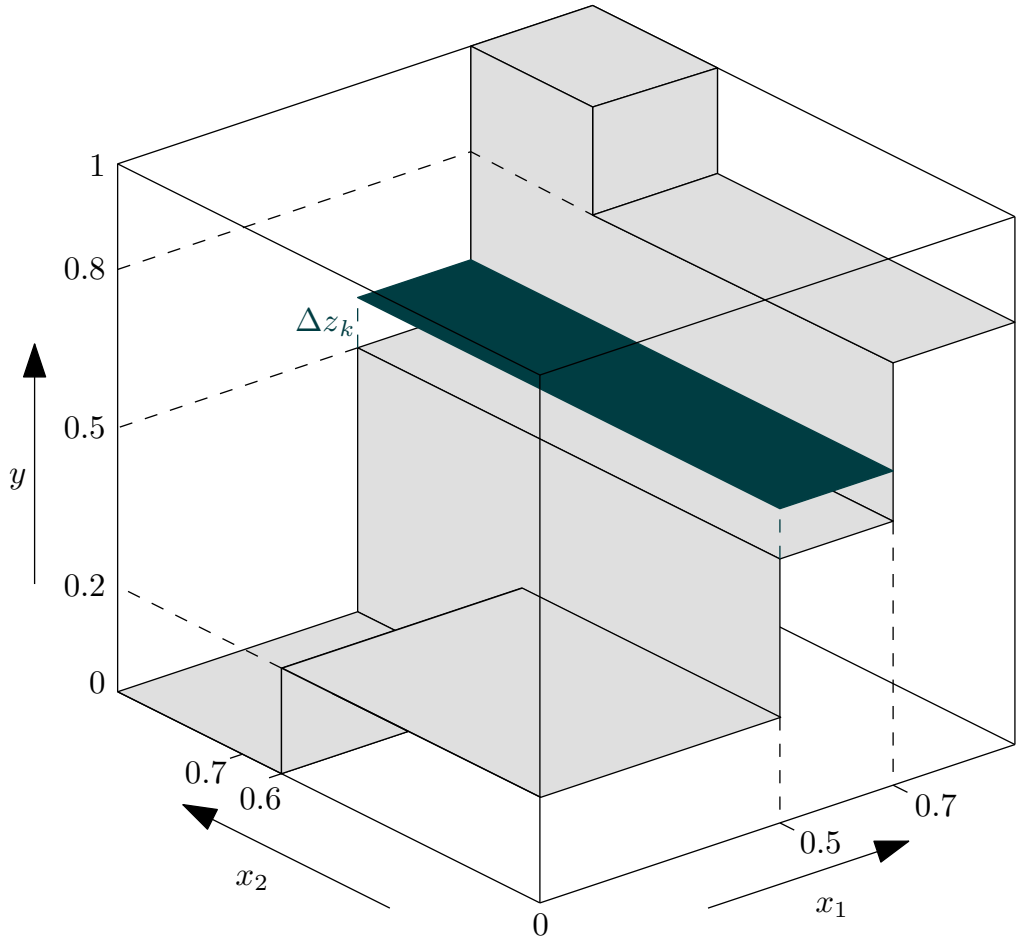
with $z = (\sqrt{n_1^1} z_1^1, \ldots, \sqrt{n_{l_M}^M} z_{l_M}^M)^T \in \mathbb{R}^p$.

Outline

1. Kernel view of tree based ensemble methods
   - Single regression tree
   - Ensemble of regression trees
2. Regularization of a tree ensemble model
3. Applications
   - Censored data
   - Semi-supervised learning
4. Conclusion & further work

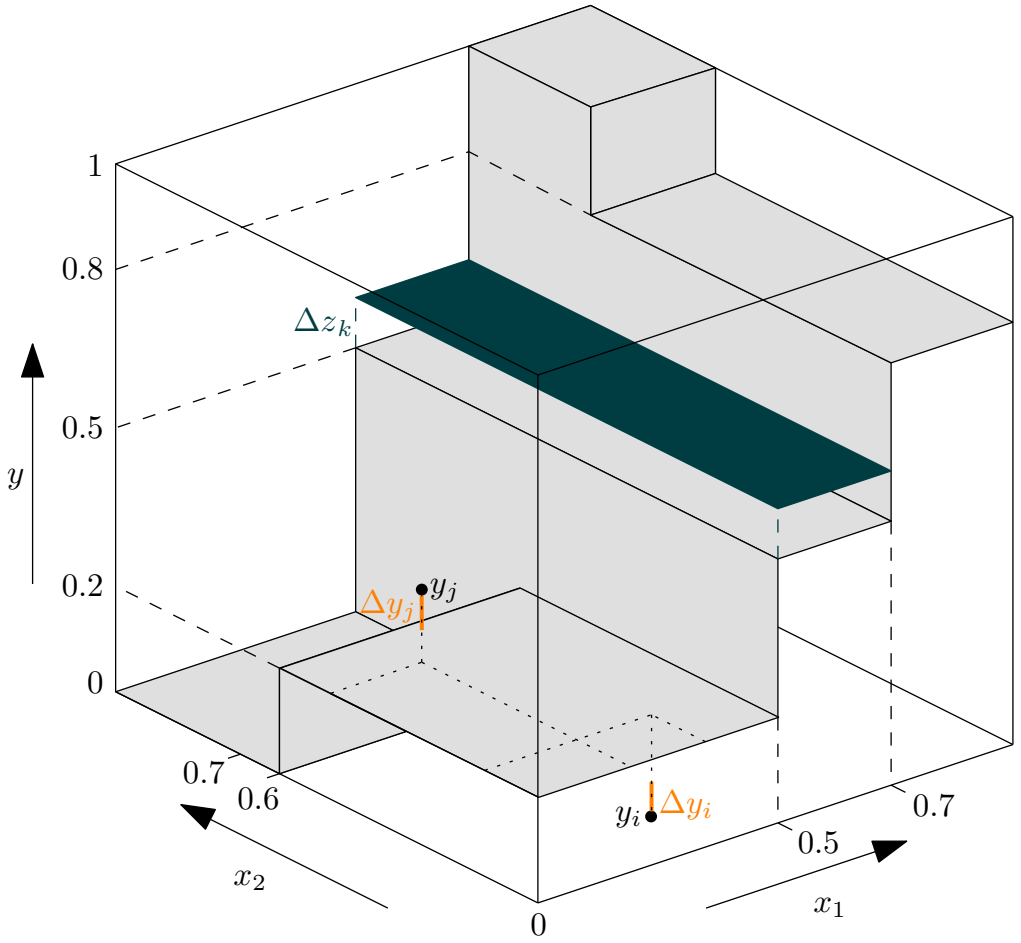
Modification/regularization of a tree predictor

1. By modifying leaf labels $\Rightarrow \Delta z \in \mathbb{R}^p$.
2. By modifying sample labels $\Rightarrow \Delta y \in \mathbb{R}^n$.



Modification/regularization of a tree predictor

1. By modifying leaf labels $\Rightarrow \Delta z \in \mathbb{R}^p$.
2. By modifying sample labels $\Rightarrow \Delta y \in \mathbb{R}^n$.



General formulation for tree ensemble regularization

$$
\begin{aligned}
\min \quad & \Omega(\Delta y, \Delta z, \nu) \\
\text{s.t.} \quad & -\nu \leq Ky + K\Delta y + L\Delta z - y \leq \nu \\
& (\Delta y, \Delta z, \nu) \in \mathcal{C}
\end{aligned}
$$

- $y \in \mathbb{R}^n$ is the vector of sample outputs,
- From the tree ensemble we get:
  - $K \in \mathbb{R}^{n \times n}$ the gram matrix over the training sample,
  - $L = \begin{pmatrix} \Phi^T(x_1) & \dots & \Phi^T(x_n) \end{pmatrix}^T \in \mathbb{R}^{n \times p}$, the matrix partionning the input space (up to some normalization),
  - K and L are not modified.

Regularization and information incorporation

$$\begin{aligned} \min \quad & \Omega(\Delta y, \Delta z, \nu) \\ \text{s.t.} \quad & -\nu \leq Ky + K\Delta y + L\Delta z - y \leq \nu \\ & (\Delta y, \Delta z, \nu) \in \mathcal{C} \end{aligned}$$

- $\nu \in \mathbb{R}^n$ measures training sample error,
- $\Omega(\cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}$ is a function expressing the compromises between regularization terms,
- $\mathcal{C} \subseteq \mathbb{R}^{n+p+n}$ is a set used to express constraints.

Problem dimensions and complexity

$$\begin{aligned} \min \quad & \Omega(\Delta y, \Delta z, \nu) \\ \text{s.t.} \quad & -\nu \leq Ky + K\Delta y + L\Delta z - y \leq \nu \\ & (\Delta y, \Delta z, \nu) \in \mathcal{C} \end{aligned}$$

- $p + 2n$ variables,
- $2n$ linear constraints + the constraints defining $\mathcal{C}$,
- and as long as Ω and $\mathcal{C}$ are convex the problem remains convex,
- If M trees are grown, $p = \sum_{i=1}^{M} l_i$ and potentially $p \gg n$.

Outline

1. Kernel view of tree based ensemble methods
   - Single regression tree
   - Ensemble of regression trees
2. Regularization of a tree ensemble model
3. Applications
   - Censored data
   - Semi-supervised learning
4. Conclusion & further work

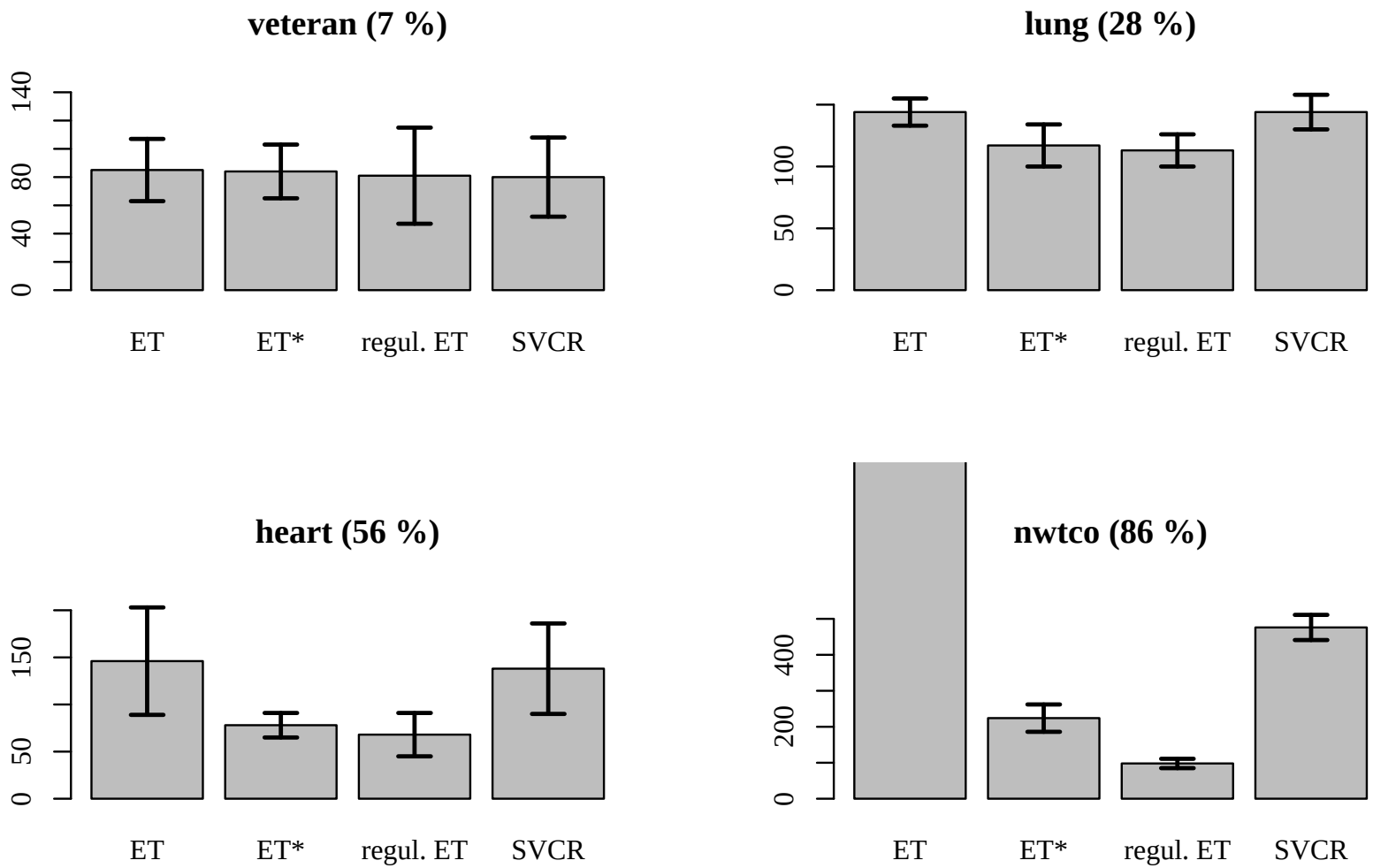
A reformulation for censored data

$$
\begin{aligned}
\min \quad & C_1\|\Delta y\| + C_2\|\Delta z\| + C_3\|\nu\| + C_4\|\nu^c\| \\
\text{s.t.} \quad & -\nu \leq Ky + K\Delta y + L\Delta z - y \leq \nu \\
& -\nu^c \leq K^c y + K^c \Delta y + L^c \Delta z - y^c
\end{aligned}
$$

- For a set of inputs $x_i, i \in \{n+1, ..., n+c\}$ we know that the output is greater or equal to y_i, $y^c = (y_{n+1}, \ldots, y_{n+c})^T$,
- K^c and L^c are computed on the censored data.

Results for 4 censored data sets (survival analysis)

Average MAE over 5 folds ± one unit standard deviation.



A reformulation for semi-supervised learning

$$
\begin{aligned}
\min \quad & \|\nu\|_2^2 + C(y + \Delta y)^T K' \mathcal{L} K' (y + \Delta y) \\
\text{s.t.} \quad & -\nu \leq \left(K'_{\ell\ell} | K'_{\ell u}\right)(y + \Delta y) - y_\ell \leq \nu, \\
& \Delta y_\ell = 0
\end{aligned}
$$

- $X = \begin{pmatrix} X_\ell \\ X_u \end{pmatrix}$, $y = \begin{pmatrix} y_\ell \\ y_u \end{pmatrix}$
- We label observations X_u while minimizing
  - the training sample error $\|\nu\|_2^2$,
  - a regularization term for predicting close values for objects whose inputs are close according to a graph, provided by its Laplacian $\mathcal{L}$.

A reformulation for semi-supervised learning

$$
\begin{aligned}
\min \quad & \|\nu\|_2^2 + C(y+\Delta y)^T K' \mathcal{L} K' (y+\Delta y) \\
\text{s.t.} \quad & -\nu \leq \left(K'_{\ell\ell} | K'_{\ell u}\right)(y+\Delta y) - y_\ell \leq \nu, \\
& \Delta y_\ell = 0
\end{aligned}
$$

- $X = \begin{pmatrix} X_\ell \\ X_u \end{pmatrix}$, $y = \begin{pmatrix} y_\ell \\ 0 \end{pmatrix}$
- We label observations X_u while minimizing
  - the training sample error $\|\nu\|_2^2$,
  - a regularization term for predicting close values for objects whose inputs are close according to a graph, provided by its Laplacian $\mathcal{L}$.

A reformulation for semi-supervised learning

$$\begin{aligned} \min \quad & \|\nu\|_2^2 + C(y+\Delta y)^T K' \mathcal{L} K' (y+\Delta y) \\ \text{s.t.} \quad & -\nu \leq \left(K'_{\ell\ell} | K'_{\ell u}\right)(y+\Delta y) - y_\ell \leq \nu, \\ & \Delta y_\ell = 0 \end{aligned}$$

- $X = \begin{pmatrix} X_\ell \\ X_u \end{pmatrix}$, $y = \begin{pmatrix} y_\ell \\ 0 \end{pmatrix}$, $\Delta y = \begin{pmatrix} \Delta y_\ell \\ \Delta y_u \end{pmatrix}$.
- We label observations X_u while minimizing
  - the training sample error $\|\nu\|_2^2$,
  - a regularization term for predicting close values for objects whose inputs are close according to a graph, provided by its Laplacian $\mathcal{L}$.

A reformulation for semi-supervised learning

$$\begin{aligned} \min \quad & \|\nu\|_2^2 + C(y+\Delta y)^T K' \mathcal{L} K' (y+\Delta y) \\ \text{s.t.} \quad & -\nu \leq \left(K'_{\ell\ell} | K'_{\ell u}\right)(y+\Delta y) - y_\ell \leq \nu, \\ & \Delta y_\ell = 0 \end{aligned}$$

- $X = \begin{pmatrix} X_\ell \\ X_u \end{pmatrix}$, $y = \begin{pmatrix} y_\ell \\ 0 \end{pmatrix}$, $\Delta y = \begin{pmatrix} 0 \\ \Delta y_u \end{pmatrix}$.
- We label observations X_u while minimizing
  - the training sample error $\|\nu\|_2^2$,
  - a regularization term for predicting close values for objects whose inputs are close according to a graph, provided by its Laplacian $\mathcal{L}$.

A reformulation for semi-supervised learning

$$
\begin{aligned}
\min \quad & \|\nu\|_2^2 + C(y + \Delta y)^T K' \mathcal{L} K' (y + \Delta y) \\
\text{s.t.} \quad & -\nu \leq \left(K'_{\ell\ell} | K'_{\ell u}\right)(y + \Delta y) - y_\ell \leq \nu, \\
& \Delta y_\ell = 0
\end{aligned}
$$

- $X = \begin{pmatrix} X_\ell \\ X_u \end{pmatrix}$, $y = \begin{pmatrix} y_\ell \\ 0 \end{pmatrix}$, $\Delta y = \begin{pmatrix} 0 \\ \Delta y_u \end{pmatrix}$.
- We label observations X_u while minimizing
  - the training sample error $\|\nu\|_2^2$,
  - a regularization term for predicting close values for objects whose inputs are close according to a graph, provided by its Laplacian $\mathcal{L}$.

Outline

1. Kernel view of tree based ensemble methods
   - Single regression tree
   - Ensemble of regression trees
2. Regularization of a tree ensemble model
3. Applications
   - Censored data
   - Semi-supervised learning
4. Conclusion & further work

Conclusion & further work

- A generic way to extend tree-based ensemble methods to censored problems and semi-supervised problems and to use other kind of prior knowledge,
- a convex regularization problem formulation,
- correct training sample outputs (Δy) or add leaf biases (Δz),
- first experiments show satisfying results.

- Validate on practical problems,
- incorporate regularization in the tree induction process.

Conclusion & further work

- A generic way to extend tree-based ensemble methods to censored problems and semi-supervised problems and to use other kind of prior knowledge,
- a convex regularization problem formulation,
- correct training sample outputs (Δy) or add leaf biases (Δz),
- first experiments show satisfying results.

- Validate on practical problems,
- incorporate regularization in the tree induction process.

Conclusion & further work

- A generic way to extend tree-based ensemble methods to censored problems and semi-supervised problems and to use other kind of prior knowledge,
- a convex regularization problem formulation,
- correct training sample outputs (Δy) or add leaf biases (Δz),
- first experiments show satisfying results.

- Validate on practical problems,
- incorporate regularization in the tree induction process.

Conclusion & further work

- A generic way to extend tree-based ensemble methods to censored problems and semi-supervised problems and to use other kind of prior knowledge,
- a convex regularization problem formulation,
- correct training sample outputs (Δy) or add leaf biases (Δz),
- first experiments show satisfying results.

- Validate on practical problems,
- incorporate regularization in the tree induction process.

Conclusion & further work

- A generic way to extend tree-based ensemble methods to censored problems and semi-supervised problems and to use other kind of prior knowledge,
- a convex regularization problem formulation,
- correct training sample outputs (Δy) or add leaf biases (Δz),
- first experiments show satisfying results.

- Validate on practical problems,
- incorporate regularization in the tree induction process.