
Semi-Supervised Learning with Decision Trees: Graph Laplacian Tree Alternating Optimization

Arman Zharmagambetov and Miguel Á. Carreira-Perpiñán
Dept. of Computer Science and Engineering, University of California, Merced

Introduction Semi-supervised learning (SSL) seeks to learn a machine learning model when only a small amount of the available data is labeled. The most widespread approach uses a graph prior, which encourages similar instances to have similar predictions. This has been very successful with models ranging from kernel machines [1] to neural networks [6], but has remained inapplicable to decision trees, for which the optimization problem is much harder.

Why trees? First, trees are considered to be interpretable models. Second, they are widely used in ensemble learning (e.g. random forests [2] or boosting [4]) as well as standalone classifiers [9]. However, similar to many non-linear methods, DTs are well-known to overfit for small sized (labeled) data which is the case in SSL. In our proposed approach, we first state the objective which consists of supervised loss (for labeled data only) and the graph Laplacian regularization (also known as manifold regularization [1]). The resulting optimization problem is long considered to be hard to solve since trees define non-differentiable and non-convex mapping. We solve this based on a reformulation of the problem which requires iteratively solving two simpler problems: a supervised tree learning problem, which can be solved by the Tree Alternating Optimization algorithm [3]; and a label smoothing problem, which can be solved through a sparse linear system. The algorithm is scalable and highly effective even with very few labeled instances, and makes it possible to learn accurate, interpretable models based on decision trees in such situations.

LapTAO: semi-supervised learning framework for decision trees We are given the dataset $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$, where $\mathcal{D}_l = \{\mathbf{x}_n, y_n\}_{n=1}^l$ is the labeled portion of data with size l and $\mathcal{D}_u = \{\mathbf{x}_n\}_{n=l+1}^N$ is unlabeled portion of data with size $N - l$. We minimize the following regularized objective:

$$E(\Theta) = \frac{1}{l} \sum_{n=1}^l (T(\mathbf{x}_n; \Theta) - y_n)^2 + \alpha \phi(\Theta) + \frac{\gamma}{N^2} \sum_{n,m=1}^N w_{nm} (T(\mathbf{x}_n; \Theta) - T(\mathbf{x}_m; \Theta))^2. \quad (1)$$

Here, w_{nm} are the entries in the similarity matrix, $T: \mathbb{R}^D \rightarrow \mathbb{R}$ is the tree predictive mapping with parameters $\Theta = \{\theta_i\}_{\text{nodes}}$, $\phi(\cdot)$ is the regularization term such as $\|\cdot\|_1$ and γ, α are hyperparameters. If T was differentiable, one could optimize (1) via gradient-based methods as it was done for neural nets [6]. However, this is non-trivial with a tree. Instead, we will reformulate the problem as *equivalent* constrained problem by introducing a new variable z for each training point:

$$\min_{z_1, \dots, z_N, \Theta} \frac{1}{l} \sum_{n=1}^l (z_n - y_n)^2 + \alpha \phi(\Theta) + \frac{\gamma}{N^2} \sum_{n,m=1}^N w_{nm} (z_n - z_m)^2 \quad (2)$$

$$\text{s.t. } z_n = T(\mathbf{x}_n; \Theta) \quad n = 1, \dots, N. \quad (3)$$

Denote $\mathbf{y} = [y_1, y_2, \dots, y_l, 0, 0, \dots]^T \in \mathbb{R}^N$, as the augmented ground truth vector. Similarly, introduce a diagonal matrix $\mathbf{J} = \text{diag}(1, \dots, 1, 0, \dots, 0) \in \mathbb{R}^{N \times N}$ with the first l diagonal entries equal to 1 and the rest 0. We also introduce graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$ where diagonal matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ with entries $d_{nn} = \sum_{m=1}^N w_{nm}$ and $\mathbf{W} = (w_{nm}) \in \mathbb{R}^{N \times N}$ is the affinity matrix. Finally, denote $\mathbf{z} = [z_1, \dots, z_N]^T$ and $\mathbf{t}(\mathbf{X}; \Theta) = [T(\mathbf{x}_1; \Theta), \dots, T(\mathbf{x}_N; \Theta)]^T$ where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$. Then we rewrite eq. (2)-(3) as follows:

$$\min_{\mathbf{z}, \Theta} (\mathbf{z} - \mathbf{y})^T \mathbf{J} (\mathbf{z} - \mathbf{y}) + \alpha \phi(\Theta) + \gamma \mathbf{z}^T \mathbf{L} \mathbf{z} \quad \text{s.t. } \mathbf{z} = \mathbf{t}(\mathbf{X}; \Theta). \quad (4)$$

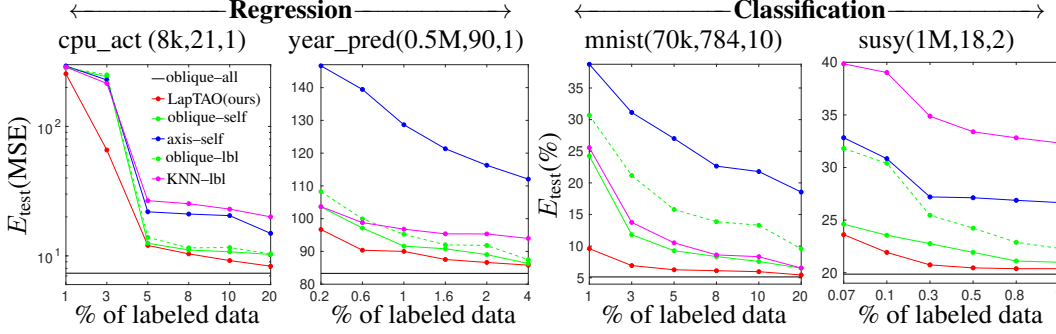


Figure 1: Results on regression and classification tasks. Numbers in brackets report the training size, number of features and output dimension (or number of classes).

Here, we absorb some constants (e.g. normalization $\frac{1}{l}$) into matrices. We solve this using the *augmented Lagrangian* [5] method which defines a new, unconstrained optimization problem:

$$\min_{\mathbf{z}, \Theta} (\mathbf{z} - \mathbf{y})^T \mathbf{J} (\mathbf{z} - \mathbf{y}) + \alpha \phi(\Theta) + \gamma \mathbf{z}^T \mathbf{L} \mathbf{z} - \lambda^T (\mathbf{z} - \mathbf{t}(\mathbf{X}; \Theta)) + \mu \|\mathbf{z} - \mathbf{t}(\mathbf{X}; \Theta)\|^2 \quad (5)$$

where $\lambda \in \mathbb{R}^N$ are the estimates of Lagrange multipliers. Optimizing this for fixed $\mu > 0$ produces the sequence of $(\mathbf{z}_\mu, \mathbf{t}_\mu(\cdot))$ and as $\mu \rightarrow \infty$, we force the minimizer to be in the feasible region for the constrained problem. Finally, we apply alternating optimization to minimize (5) over \mathbf{z} and $\mathbf{t}(\cdot)$:

- **Label-step** (optimizing over \mathbf{z} given fixed $\mathbf{t}(\mathbf{X}; \Theta)$). The objective in eq. (5) is a quadratic function and a minimizer is obtained by solving the linear system:

$$\begin{aligned} \min_{\mathbf{z}} (\mathbf{z} - \mathbf{y})^T \mathbf{J} (\mathbf{z} - \mathbf{y}) + \gamma \mathbf{z}^T \mathbf{L} \mathbf{z} - \lambda^T (\mathbf{z} - \mathbf{t}(\mathbf{X}; \Theta)) + \mu \|\mathbf{z} - \mathbf{t}(\mathbf{X}; \Theta)\|^2 \Rightarrow \\ \mathbf{A} \mathbf{z} = \mathbf{J} \mathbf{y} + \mu \mathbf{t}(\mathbf{X}; \Theta) + \frac{1}{2} \lambda \end{aligned} \quad (6)$$

where $\mathbf{A} = \mathbf{J} + \mu \mathbf{I} + \gamma \mathbf{L}$ is a positive definite matrix. Moreover, \mathbf{A} will be a sparse matrix if graph Laplacian \mathbf{L} is sparse. This allows us to solve the large scale linear system in an efficient way. Intuitively, the “label-step” can be interpreted as “approximating” the labels (for \mathcal{D}_u) using the graph Laplacian and predictions obtained from the current tree (i.e., label smoothing).

- **Tree-step** (optimizing over Θ given fixed \mathbf{z}). The problem (5) reduces to a regression fit of a tree:

$$\begin{aligned} \min_{\Theta} \mu \|\mathbf{z} - \mathbf{t}(\mathbf{X}; \Theta)\|^2 + \alpha \phi(\Theta) - \lambda^T (\mathbf{z} - \mathbf{t}(\mathbf{X}; \Theta)) \Leftrightarrow \\ \min_{\Theta} \left\| \left(\mathbf{z} - \frac{1}{2\mu} \lambda \right) - \mathbf{t}(\mathbf{X}; \Theta) \right\|^2 + \frac{\alpha}{\mu} \phi(\Theta). \end{aligned} \quad (7)$$

Note that here we use $(\mathbf{z} - \frac{1}{2\mu} \lambda)$ as labels (not y_n which is not defined for \mathcal{D}_u anyway). We solve this problem using TAO algorithm and refer a reader to [3, 8] for details. Intuitively, this step can be understood as fitting a tree with the current “estimates” of the labels.

After each (label,tree)-step, the penalty term μ increases, λ is updated and we keep iterating until a stopping criterion. We denote our proposed iterative algorithm as *LapTAO*. Although we derive our algorithm specifically for a tree, *it is obvious to extended it to other models*: gradient boosted trees, neural networks, etc. The only part that trivially changes is the “tree-step”.

Experiments We pick a *sparse oblique tree with constant leaves* as our main model for LapTAO. We compare against the following baselines: 1) *oblique-all* fits an oblique tree with full supervision (theoretical maximum performance); 2) *oblique-lbl* is the oblique trees trained on labeled data only; 3) Self-training (*axis-self*, *oblique-self*) is an iterative procedure that uses the model predictions to enlarge the portion of labeled data [7]; here, “axis” means traditional axis-aligned trees. The results in fig. 1 shows that *LapTAO consistently improves over all other SSL baselines*, often by a considerable margin. For instance, in case of 3% in *cpu_act* and 1% in *mnist*, the difference in the error with the second best SSL approach is several orders of magnitude. It shows acceptable results even in extreme label scarcity scenarios, e.g. when we provide $< 0.5\%$ of labeled data on *year_pred* and *susy*. Moreover, LapTAO approaches the fully supervised baseline more quickly: for *mnist*, we can achieve the same $\sim 5\%$ test error as “oblique-all” using only 20% of labeled training points.

Acknowledgments and Disclosure of Funding

Work funded in part by NSF award IIS–2007147.

References

- [1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Machine Learning Research*, 7:2399–2434, Nov. 2006.
- [2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.
- [3] M. Á. Carreira-Perpiñán and P. Tavallali. Alternating optimization of decision trees, with application to learning sparse oblique trees. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NEURIPS)*, volume 31, pages 1211–1221. MIT Press, Cambridge, MA, 2018.
- [4] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.
- [5] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York, second edition, 2006.
- [6] J. Weston, F. Ratle, and R. Collobert. Deep learning via semi-supervised embedding. In A. McCallum and S. Roweis, editors, *Proc. of the 25th Int. Conf. Machine Learning (ICML'08)*, pages 1168–1175, Helsinki, Finland, July 5–9 2008.
- [7] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL '95)*, pages 189–196, 1995.
- [8] A. Zharmagambetov and M. Á. Carreira-Perpiñán. Smaller, more accurate regression forests using tree alternating optimization. In H. Daumé III and A. Singh, editors, *Proc. of the 37th Int. Conf. Machine Learning (ICML 2020)*, pages 11398–11408, Online, July 13–18 2020.
- [9] A. Zharmagambetov, S. S. Hada, M. Gabidolla, and M. Á. Carreira-Perpiñán. Non-greedy algorithms for decision tree optimization: An experimental comparison. In *Int. J. Conf. Neural Networks (IJCNN'21)*, Virtual event, July 18–22 2021.