Fast Training of Support Vector Machines for Survival Analysis

Sebastian Pölsterl[1], Nassir Navab[1,2], and Amin Katouzian[1]

1: Computer Aided Medical Procedures, Technische Universität München, Munich, Germany

2: Johns Hopkins University, Baltimore MD, USA

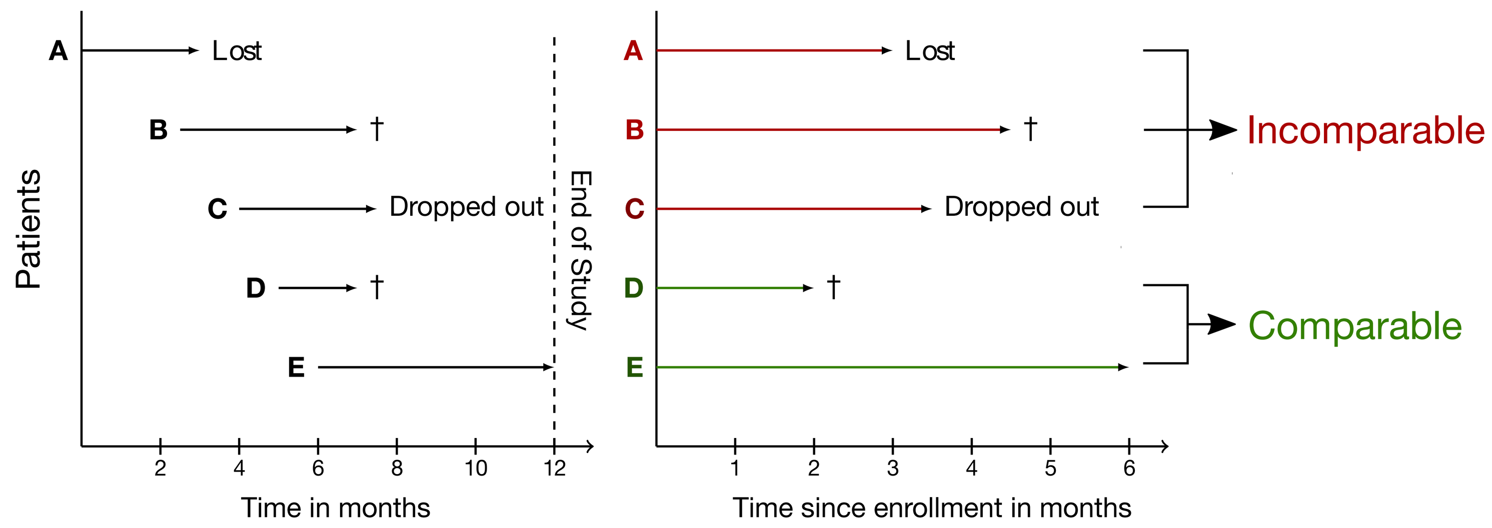
Overview

- **Problem**: Training a Survival SVM by converting the ranking problem into a classification problem and using a standard dual SVM solver requires $O(dn^4)$ time and $O(n^2)$ space.[1]
- Current RankSVM solvers[2] require $O(nd + n\log n + n\log r + d)$.
- **Solution**:
  - Optimization in the primal using truncated Newton optimization.
  - Use order statistics trees to lower complexities.
- **Conclusion**: Requires $O(n)$ space and has time complexity $[O(n\log n) + O(nd + d + n\log n)] \cdot \bar{N}_{\mathrm{CG}} \cdot N_{\mathrm{Newton}}$.

- Same optimization scheme can be applied to:
  - Non-linear Survival SVM: Transform data with Kernel PCA before training in primal.[3]
  - Hybrid ranking-regression to predict exact time of event.

Survival Analysis

- **Objective**: to establish a connection between a set of features and the time of an event (*survival time*).
- Patients that remain event-free during the study-period are **right censored**, because it is unknown whether an event has or has not occurred after the study ended. Only **partial information** about their survival is available.



Survival Support Vector Machine

- Survival analysis can be formulated as a ranking problem, with the objective to predict the order of patients according to their survival time.
- Survival SVM[1] is closely related to RankSVM[4], but in addition accounts for right censoring: subject with *smaller survival time* must always be *uncensored*.
- Number of relevance levels (here: *survival times*) are of the order of number of patients.
- Survival data consists of a feature vector ($\boldsymbol{x}_i \in \mathbb{R}^d$), the time of an event/censoring ($y_i > 0$), and an event indicator ($\delta_i \in \{0, 1\}$).

Truncated Newton Optimization

- **Objective function**:

$$\mathcal{P} = \{(i,j) \mid y_i > y_j \wedge \delta_j = 1\}_{i,j=1,\ldots,n}$$

$$f(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i,j\in\mathcal{P}} \max(0, 1 - (\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j))^2$$

$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\left(p_w + \boldsymbol{w}^T\boldsymbol{X}^T\left(\boldsymbol{A}_{\boldsymbol{w}}^T\boldsymbol{A}_{\boldsymbol{w}}\boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{A}_{\boldsymbol{w}}^T\right)\right)$$

$$(\boldsymbol{A}_{\boldsymbol{w}})_{k,q} = \begin{cases} 1 & \text{if } y_q < y_s \wedge \delta_q = 1 \wedge \boldsymbol{w}^T\boldsymbol{x}_s < \boldsymbol{w}^T\boldsymbol{x}_q + 1, \quad (1) \\ -1 & \text{if } y_q > y_s \wedge \delta_s = 1 \wedge \boldsymbol{w}^T\boldsymbol{x}_s > \boldsymbol{w}^T\boldsymbol{x}_q - 1, \quad (2) \\ 0 & \text{else,} \end{cases}$$

where $s, q \in \{1, \ldots, n\}, k \in \{1, \ldots, p_w\}$

- **Example**: $\mathcal{P} = \{(1,2); (1,3); (2,3); (4,2); (4,3)\}$

i	1	2	3	4
y_i	9	6	5	8
δ_i	0	1	1	0
$\boldsymbol{w}^T\boldsymbol{x}_i$	-0.7	-0.1	0.15	1.6

$$\boldsymbol{A}_{\boldsymbol{w}} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \end{pmatrix}$$

- Do not store Hessian, but only compute Hessian-vector product $\boldsymbol{H}\boldsymbol{v} = \boldsymbol{v} + \gamma\boldsymbol{X}^T\boldsymbol{A}_{\boldsymbol{w}}^T\boldsymbol{A}_{\boldsymbol{w}}\boldsymbol{X}\boldsymbol{v}$.

Using Order Statistics Trees

- Matrix $\boldsymbol{A}_{\boldsymbol{w}}^T\boldsymbol{A}_{\boldsymbol{w}}$ has size $O(n^2)$ and has to be recomputed in each iteration, which would be impractical. Instead we compactly express one entry as

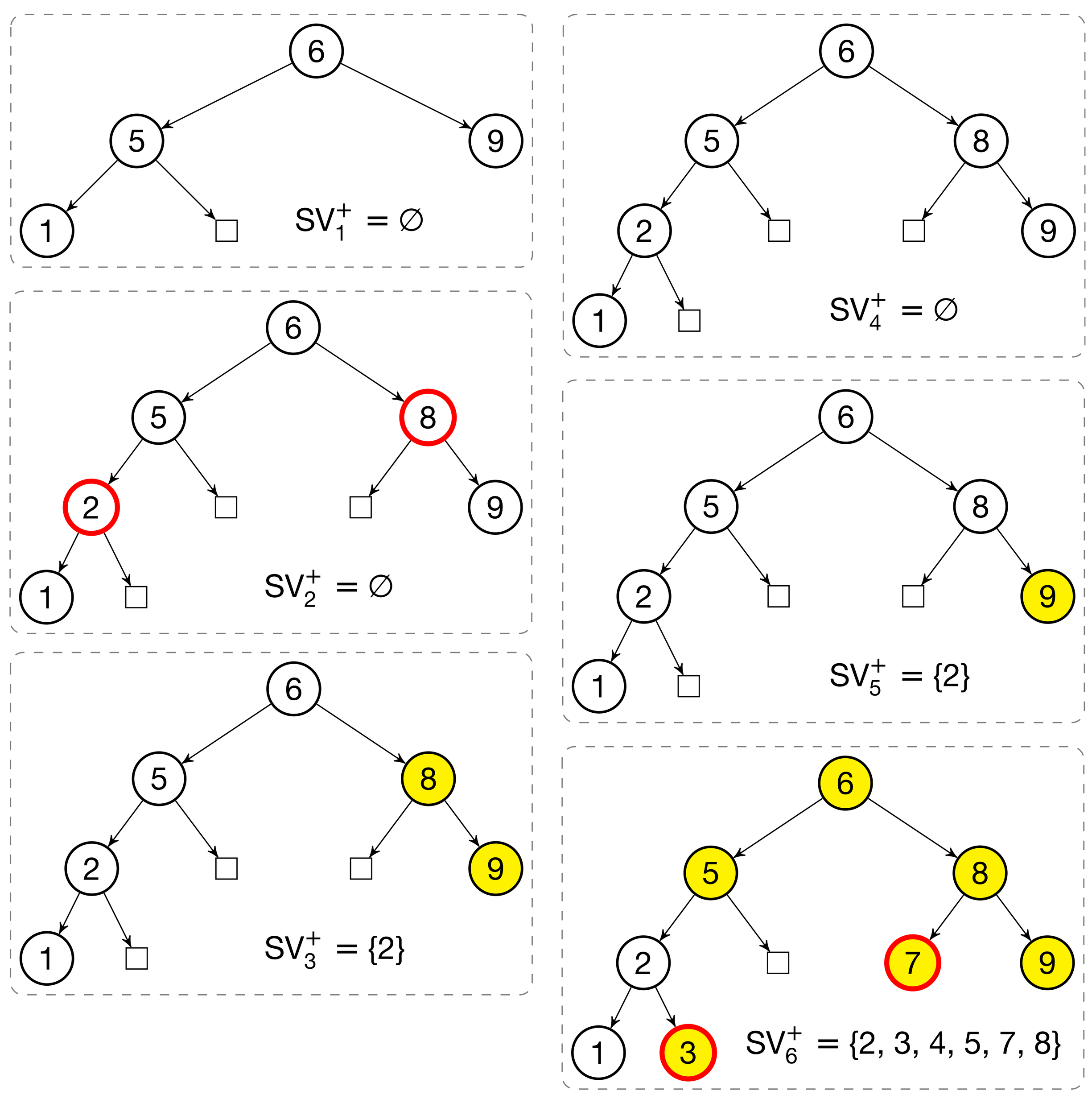
$$(\boldsymbol{A}_{\boldsymbol{w}}^T\boldsymbol{A}_{\boldsymbol{w}})_{i,j} = \begin{cases} l_i^+ + l_i^- & \text{if } i = j, \\ -1 & \text{if } i \neq j, \text{ and } j \in \mathrm{SV}_i^+ \text{ or } i \in \mathrm{SV}_j^-, \\ 0 & \text{else.} \end{cases}$$

$$\mathrm{SV}_i^+ = \{s \mid y_s > y_i \wedge \delta_i = 1 \wedge \boldsymbol{w}^T\boldsymbol{x}_s < \boldsymbol{w}^T\boldsymbol{x}_i + 1\} \qquad l_i^+ = |\mathrm{SV}_i^+|$$

$$\mathrm{SV}_i^- = \{s \mid y_s < y_i \wedge \delta_s = 1 \wedge \boldsymbol{w}^T\boldsymbol{x}_s > \boldsymbol{w}^T\boldsymbol{x}_i - 1\} \qquad l_i^- = |\mathrm{SV}_i^-|$$

- Sets can be updated incrementally by adding y_i and $\boldsymbol{w}^T\boldsymbol{x}_i$ to an order statistics tree (balanced binary search tree).
- **Example**:

i	1	2	3	4	5	6	7	8	9
$\boldsymbol{w}^T\boldsymbol{x}_i$	-0.7	-0.1	0.15	0.2	0.3	0.8	1.6	1.7	2.3
y_i	1	9	6	5	8	2	7	3	4
δ_i	0	0	1	0	1	1	1	0	0



1.Van Belle et al.: Support Vector Machines for Survival Analysis. In: Proc. 3rd Int. Conf. Comput. Intell. Med. Healthc. 1–8. 2007
2.Lee et al.: Large-Scale Linear RankSVM. Neural Comput. 26(4), 781–817. 2014
3.Chapelle et al.: Efficient algorithms for ranking with SVMs. Information Retrieval 13(3), 201–5. 2009
4.Herbrich et al.: Large Margin Rank Boundaries for Ordinal Regression. In: Advances in Large Margin Classifiers. 115–32. 2000