An Efficient Training Algorithm for Kernel Survival Support Vector Machines

Sebastian Pölsterl / Nassir Navab / Amin Katouzian

Overview

- Survival analysis tries to establish a connection between a set of features and the time until an event of interest.
- Problem: For a dataset with \( n \) samples and \( p \) features, previous training algorithms for the kernel survival SVM (SSVM) require \( O(n^3) \) space and \( O(pn^2) \) time.\(^2\)
- Recently, an efficient training algorithm for linear SSVM with much lower time complexity and linear space complexity has been proposed.\(^3\)
- We extend this optimisation scheme to the non-linear case and show that it allows analysing large-scale data with no loss in prediction performance.

Survival Analysis

- In survival analysis, parts of the training and test data can only be partially observed.
- 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.

Kernel Survival Support Vector Machine

- The SSVM\(^1,2\) is an extension of the Rank SVM\(^4\) to right censored survival data: patients with a lower survival time should be ranked before patients with longer survival time.
- Survival data consists of feature vectors \( x_i \in \mathbb{R}^p \), the time of an event/censoring \( t_i \), and an event indicator \( y_i \).

Objective function: \[ P = \{(i, j) \mid y_i > y_j \wedge (t_i > t_j)\} \]

Lagrangian dual problem with \( K_{i, j} = \phi(x_i)^T \phi(x_j) \):

\[ \max_{\alpha} \frac{1}{2} \sum_{(i, j) \in P} \alpha_i \alpha_j K_{i, j} - \frac{1}{2} \alpha^T \mathbf{A} \alpha \]

subject to \( \alpha \geq 0 \) and \( \sum_{i} \alpha_i = 1 \).

Proposed Optimisation Scheme

- Find a function \( f \colon \mathcal{X} \rightarrow \mathbb{R} \) from a reproducing Kernel Hilbert space \( \mathcal{H}_k \) with \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \) (usually \( \mathcal{X} \sim \mathbb{R}^d \)):

\[ \min_{f, \alpha} \frac{1}{2} \|f\|^2_{\mathcal{H}_k} + \sum_{(i, j) \in P} \max\{0, 1 - (f(x_i) - f(x_j))\}^2 \]

- Directly apply representor theorem\(^5\) \( f(x) = \sum_{i=1}^{n} \beta_i k(x_i, x) \) and minimise squared hinge loss with respect to \( \beta \in \mathbb{R}^n \):

\[ R(\beta) = \frac{1}{2} \sum_{i=1}^{n} \max\{0, 1 - \sum_{j=1}^{n} \beta_j (k(x_i, x_j) - k(x_i, x_j))\}^2 + \frac{\gamma}{2} \sum_{(i, j) \in P} \max\{0, 1 - \sum_{l=0}^{n} \beta_l k(x_i, x_j) - k(x_i, x_j)\} \]

- \( D(\beta)_{i,j} = \begin{cases} 1 & \text{if } f(x_i) > f(x_j) \Rightarrow \beta_i > \beta_j, \\ 0 & \text{else}. \end{cases} \)

- The form of the optimisation problem is very similar to the one of linear SSVM, which allows using truncated Newton optimisation and order statistic trees to avoid storing \( A \).\(^3,5\)

Complexity

- Space: \( O(n) \) or \( O(n^2) \) if \( K \) is to be stored in memory.
- Time (assuming evaluating kernel function costs \( O(p) \)): \( O(n^3) \) to compute \( K_{i, i} \) for all \( i \) and \( n \) to sort samples according to \( K_{i, i} \); \( O(n^2 + n \log n) \) to compute Hessian-vector product.

\[ \text{Overall (if kernel matrix is stored in memory):} \]

\[ O(n^2p) + O(n \log n) \cdot \mathcal{N} \cdot \mathcal{C} \]

Experiments

Synthetic:

Real-world:

References

3. Pölder et al.: Fast training of support vector machines for survival analysis. ECML PKDD 2015
5. Kuo et al.: Large-scale kernel RankSVM. SIAM International Conference on Data Mining. 2014
7. Van Belle et al.: Learning transformation models for ranking and survival analysis. JMLR. 10, pp. 819–42. 2011

Acknowledgments

This work has been supported by the CRUK Centre at the Institute of Cancer Research and Royal Marsden (Grant No. C309/A18077), the Heather Brothers Charitable Settlement, The John L. Beckwith Charitable Trust, and the Leibniz Supercomputing Centre.