ROSVM Package - Mathematical Background

Eric Bach

July 20, 2020

1 ToDo

- Add derivations for the exterior product features.

2 Introduction

This documents describes the mathematical background of the Ranking Support Vector Machine (RankSVM) [2] implemented in the ROSVM package.

3 Method

3.1 Notation

3.2 Ranking Support Vector Machine (RankSVM)

The RankSVM's primal optimization problem is given as:

$$\begin{aligned} \underset{\mathbf{w},\boldsymbol{\xi}}{min} \quad & f(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{(i,j)\in P} \xi_{ij} \\ \text{s.t.} \quad & y_{ij}\mathbf{w}^T(\phi_i - \phi_j) \geq 1 - \xi_{ij}, \quad \forall (i,j) \in \mathcal{P} \\ & \xi_{ij} \geq 0, \quad \forall (i,j) \in \mathcal{P}, \end{aligned} \tag{1}$$

where $C > 0$ is the regularization parameter. We define the pairwise labels as the retention time difference of the corresponding molecules, i.e. $y_{ij} := \text{sign}(t_i - t_j)$. From the primal problem in Eq. (1) we can derive the following dual optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & g(\boldsymbol{\alpha}) = \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{y}\mathbf{y}^T \circ \mathbf{BKB}^T\right)\boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_{ij} \leq C, \quad \forall (i,j) \in \mathcal{P}, \end{aligned} \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the vector of pairwise labels, and $\mathbf{B} \in \{-1,0,1\}^{m\times n}$ with row $p = (i,j)$ being $[\mathbf{B}]_{p\cdot} = (0,\dots,0,\underbrace{1}_{i},0,\dots,0,\underbrace{-1}_{j},0,\dots,0)$. For further details refer to the work by [3]. Using the properties of the Hadamard product $\circ$ we can reformulate the function $g(\boldsymbol{\alpha})$ of the problem in Eq. (2) [4]:

$$\begin{aligned} g(\boldsymbol{\alpha}) &= \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{y}\mathbf{y}^T \circ \mathbf{BKB}^T\right)\boldsymbol{\alpha} \\ &= \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{D_y}\mathbf{BKB}^T\mathbf{D_y}\right)\boldsymbol{\alpha} \\ &= \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{AKA}^T\boldsymbol{\alpha}. \end{aligned}$$

Table 1: Notation table

Notation	Description
$\mathcal{P}$	Set of preferences

Algorithm 1: Conditional gradient algorithm

Let $\boldsymbol{\alpha}^{(0)} \in \mathcal{A}$; `/* A feasible initial dual variable. */`
for $k = 0, \dots, (K-1)$ **do**
  $\mathbf{s} \leftarrow \arg\max_{\mathbf{s}' \in \mathcal{A}} \left\langle \nabla g(\boldsymbol{\alpha}^{(k)}), \mathbf{s}' \right\rangle$; `/* Solve sub-problem */`
  $\gamma \leftarrow \frac{2}{k+2}$; `/* Step-size; also line-search possible */`
  $\boldsymbol{\alpha}^{(k+1)} \leftarrow (1-\gamma)\boldsymbol{\alpha}^{(k)} + \gamma \mathbf{s}$; `/* Update */`
end
$\boldsymbol{\alpha}^* \leftarrow \boldsymbol{\alpha}^{(K)}$;

Here, $\mathbf{D_y} \in \mathbb{R}^{m \times m}$ is a diagonal matrix storing the pairwise labels, and $\mathbf{A} := \mathbf{D_y B} \in \{-1, 0, 1\}^{m \times n}$. The matrix $\mathbf{A}$ now contains the pairwise labels as well by multiplying each row $p = (i, j)$ of $\mathbf{B}$ with y_{ij}, i.e. $[\mathbf{A}]_{p\cdot} = y_{ij} \cdot (0, \dots, 0, \underbrace{1}_{i}, 0, \dots, 0, \underbrace{-1}_{j}, 0, \dots, 0)$.

> Check out '`_build_A_matrix`' for the actual implementation of the **A**-matrix construction from the data.

3.2.1 Optimizing the RankSVM Model Parameters

We find the optimal RankSVM model $\boldsymbol{\alpha}^*$ in the dual space given a training dataset $\mathcal{D} = \{(x_i, t_i)\}_{i=1}^n$ using the conditional gradient algorithm [1]. The algorithm is shown in 1. The feasible set is defined as $\mathcal{A} := \{\boldsymbol{\alpha} \in \mathbb{R}^m \mid 0 \leq \alpha_{ij} \leq C, \forall (i,j) \in \mathcal{P}\}$ which follows from the constraints of the dual optimization problem in Eq. (2). Note that $\mathcal{A}$ is compact convex set.

> The function '`_assert_is_feasible`' implements the feasibility check for a given $\boldsymbol{\alpha}^{(k)}$ iterate.

Solving the Sub-problem: In each iteration of Algorithm 1 we need to solve the following linear optimization problem:

$$\begin{aligned} \mathbf{s} &= \arg\max_{\mathbf{s}' \in \mathcal{A}} \left\langle \nabla g(\boldsymbol{\alpha}^{(k)}), \mathbf{s}' \right\rangle \\ &= \arg\max_{\mathbf{s}' \in \mathcal{A}} \left\langle \underbrace{\mathbf{1} - \mathbf{AKA}^T \boldsymbol{\alpha}^{(k)}}_{:=\mathbf{d}}, \mathbf{s}' \right\rangle. \end{aligned} \tag{3}$$

Eq. (3) can be solved by simply evaluating $\mathbf{d}$ and subsequently setting the components of $s \in \mathbb{R}^m$ as:

$$s_{ij} = \begin{cases} C & \text{if } d_{ij} > 0 \\ 0 & \text{else.} \end{cases}$$

The function '`_solve_sub_problem`' implements the sub problem solver.

References

[1] Martin Jaggi. "Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta et al. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 427–435. URL: `http://proceedings.mlr.press/v28/jaggi13.html`.

[2] Thorsten Joachims. "Optimizing Search Engines Using Clickthrough Data". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. Edmonton, Alberta, Canada: ACM, 2002, pp. 133–142. ISBN: 1-58113-567-X. DOI: `10.1145/775047.775067`. URL: `http://doi.acm.org/10.1145/775047.775067`.

[3] Tzu-Ming Kuo et al. "Large-scale kernel rankSVM". In: *Proceedings of the 2014 SIAM international conference on data mining*. SIAM. 2014, pp. 812–820.

[4] George P.H. Styan. "Hadamard products and multivariate statistical analysis". In: *Linear Algebra and its Applications* 6 (1973), pp. 217–240. ISSN: 0024-3795. DOI: `https://doi.org/10.1016/0024-3795(73)90023-2`. URL: `http://www.sciencedirect.com/science/article/pii/0024379573900232`.