ROSVM Package - Mathematical Background

Eric Bach

August 19, 2020

Contents

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

Table 1 summarizes the notation used in this document.

3.2 Ranking Support Vector Machine (RankSVM)

The RankSVM's primal optimization problem is given as:

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

Table 1: Notation table

Notation	Description
$\mathcal{P}$	Set of preferences with size $m = \lvert\mathcal{P}\rvert$
$m \in \mathcal{M}$	Molecule from the space of molecules $\mathcal{M}$
$\mathbf{x} \in \mathbb{R}^{d_x}$	Molecule feature representation, e.g. fingerprint vector, with dimension d_x
$\mathbf{z} \in \mathbb{R}^{d_z}$	Chromatographic system feature representation with dimension d_z
$\phi(\mathbf{x}) \in \mathbb{R}^{d_\mathcal{X}}$	*Kernel* feature representation of a molecule based on $\mathbf{x}$
$\phi_i = \phi(\mathbf{x}_i)$	Shorthand for the kernel feature vector of example i
$\mathbf{\Phi} \in \mathbb{R}^{n \times d_\mathcal{X}}$	Kernel feature vector matrix, with n examples each of dimension $d_\mathcal{X}$

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:

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

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 $[\mathbf{B}]_{p\cdot} = (0, \ldots, 0, \underbrace{1}_{i}, 0, \ldots, 0, \underbrace{-1}_{j}, 0, \ldots, 0)$. 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]:

$$\begin{aligned} g(\boldsymbol{\alpha}) &= \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{y}\mathbf{y}^T \circ \mathbf{B}\mathbf{K}\mathbf{B}^T\right)\boldsymbol{\alpha} \\ &= \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{D}_\mathbf{y}\mathbf{B}\mathbf{K}\mathbf{B}^T\mathbf{D}_\mathbf{y}\right)\boldsymbol{\alpha} \\ &= \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{A}\mathbf{K}\mathbf{A}^T\boldsymbol{\alpha}. \end{aligned}$$

Here, $\mathbf{D}_\mathbf{y} \in \mathbb{R}^{m \times m}$ is a diagonal matrix storing the pairwise labels, and $\mathbf{A} := \mathbf{D}_\mathbf{y}\mathbf{B} \in \{-1, 0, 1\}^{m \times n}$. 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. $[\mathbf{A}]_{p\cdot} = y_{ij} \cdot (0, \ldots, 0, \underbrace{1}_{i}, 0, \ldots, 0, \underbrace{-1}_{j}, 0, \ldots, 0)$.

> 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 $\mathcal{A} := \{\boldsymbol{\alpha} \in \mathbb{R}^m \,|\, 0 \leq \alpha_{ij} \leq C, \forall (i,j) \in \mathcal{P}\}$ which follows from the constraints of the dual optimization problem in Eq. (2). Note that $\mathcal{A}$ is compact convex set.

Algorithm 1: Conditional gradient algorithm

Let $\boldsymbol{\alpha}^{(0)} \in \mathcal{A}$; `/* A feasible initial dual variable. */`

for $k = 0, \ldots, (K-1)$ **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 */`

  $\gamma^{(k)} \leftarrow \frac{2}{k+2}$; `/* Step-size; also line-search possible */`

  $\boldsymbol{\alpha}^{(k+1)} \leftarrow (1-\gamma)\boldsymbol{\alpha}^{(k)} + \gamma \mathbf{s}$; `/* Update */`

end

$\boldsymbol{\alpha}^{*} \leftarrow \boldsymbol{\alpha}^{(K)}$;

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:

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

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.

Line-search: The optimal step-size $\gamma^{(k)}$ can be determined by solving an univariate problem:

$$\gamma_{LS}^{(k)} = \max_{\gamma \in [0,1]} \quad g\left(\boldsymbol{\alpha}^{(k)} - \gamma\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)\right). \tag{4}$$

For that, we set the derivative of (4) to zero:

$$\begin{aligned} &\frac{\partial g\left(\boldsymbol{\alpha}^{(k)} - \gamma\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)\right)}{\partial \gamma} \\ &= \left(\boldsymbol{\alpha}^{(k)} - \gamma\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) - \mathbf{1}^T\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) \\ &= \left(\boldsymbol{\alpha}^{(k)}\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) - \gamma\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) - \mathbf{1}^T\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) \\ &= 0 \end{aligned}$$

and solve for γ:

$$\begin{aligned}
\gamma \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) &= \left(\boldsymbol{\alpha}^{(k)}\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) - \mathbf{1}^T \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) \\
&\Leftrightarrow \\
\gamma &= \frac{\left(\boldsymbol{\alpha}^{(k)}\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right) - \mathbf{1}^T \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)}{\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)} \\
&\Leftrightarrow \\
\gamma_{LS}^{(k)} &= \frac{\left\langle \nabla g\left(\boldsymbol{\alpha}^{(k)}\right), \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)\right\rangle}{\left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)^T \mathbf{A}\mathbf{K}\mathbf{A}^T \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)}. \qquad (5)
\end{aligned}$$

To ensure that $\gamma_{LS}^{(k)}$ we clip the value to the range of $[0, 1]$. To evaluate the nominator in Eq. (5) we can reuse the value of $\mathbf{d} = \nabla g\left(\boldsymbol{\alpha}^{(k)}\right)$ calculated when solving the sub-problem (see Eq. (3))

Determine Convergence: Jaggi [1] propose to use the duality gap:

$$h\left(\boldsymbol{\alpha}^{(k)}\right) := \left\langle \nabla g\left(\boldsymbol{\alpha}^{(k)}\right), \left(\mathbf{s} - \boldsymbol{\alpha}^{(k)}\right)\right\rangle$$

as convergence criteria by defining a threshold ϵ and iterating until $h(\boldsymbol{\alpha}^{(k)}) < \epsilon$. The ratio behind this idea is, that the duality gap is an upper bound on the difference of the current function value for $\boldsymbol{\alpha}^{(k)}$ and the one of the global maximizer $\boldsymbol{\alpha}^*$ of Eq. (1), i.e. $h\left(\boldsymbol{\alpha}^{(k)}\right) \geq g\left(\boldsymbol{\alpha}^*\right) - g\left(\boldsymbol{\alpha}^{(k)}\right)$[1].

In practice it the duality gap $h\left(\boldsymbol{\alpha}^{(k)}\right)$ was observed to have very large values and never approaching a reasonable threshold. However, the model performance was nevertheless very good. That might be because the $\boldsymbol{\alpha}^{(k)}$ has many entries and those can have values up to C. A quadratic function of the dual vector, like g in Eq. (2), can take on very large values. The "un-boundedness" or missing normalization might be an issue here.

The current implementation gets around this issue, by checking the following quantity for the convergence:

$$\frac{h(\boldsymbol{\alpha}^{(k)})}{h(\boldsymbol{\alpha}^{(0)})} < \epsilon,$$

where $\epsilon > 0$, e.g. $\epsilon = 0.005$, is the convergence threshold. Possible scaling factors of the function h are canceled out. This convergence criteria can be interpreted as the relative decrease of the duality gap given the initial model $\boldsymbol{\alpha}^{(0)}$.

3.2.2 Prediction Step

Given two molecules m_u and m_v their retention order label $\hat{y}_{uv}$ using a trained RankSVM model $\mathbf{w} = \mathbf{\Phi}^T \mathbf{A}^T \boldsymbol{\alpha}$:

$$\hat{y}_{uv} = \text{sign}(\langle \mathbf{w}, \phi_u - \phi_v \rangle) = \text{sign}\left(\left\langle \mathbf{\Phi}^T \mathbf{A}^T \boldsymbol{\alpha}, \phi_u - \phi_v \right\rangle\right).$$

[1] Note: Here we formulate the dual optimization as a maximization. In [1] the authors formulate it as a minimization which leads to slightly changed duality gap definition.

Sometimes, it might be of use, to get the decision score for a molecule. This score can be calculated by exploiting the linearity of the kernel feature vector *difference*

$$\langle \mathbf{w}, \phi_u - \phi_v \rangle = \langle \mathbf{w}, \phi_u \rangle - \langle \mathbf{w}, \phi_v \rangle.$$

We can now evaluate the decision score:

$$\begin{aligned} s_u &= \langle \mathbf{w}, \phi_u \rangle \\ &= \left\langle \mathbf{\Phi}^T \mathbf{A}^T \boldsymbol{\alpha}, \phi_u \right\rangle \\ &= \boldsymbol{\alpha}^T \mathbf{A} \mathbf{k}(\mathbf{x}_u) \\ &= \sum_{(i,j)\in\mathcal{P}} \alpha_{ij} y_{ij} \left(\kappa(\mathbf{x}_i, \mathbf{x}_u) - \kappa(\mathbf{x}_j, \mathbf{x}_u) \right), \end{aligned}$$

with $\mathbf{k}(\mathbf{x}_u) = (\kappa(\mathbf{x}_1, \mathbf{x}_u), \ldots, \kappa(\mathbf{x}_n, \mathbf{x}_u)) \in \mathbb{R}^n$ being a vector containing the kernel similarities between the molecule m_u (using its representation $\mathbf{x}_u$) and all molecules in the training set.

3.3 Include Chromatographic System Descriptors

In this section we are inspecting different ways to include feature descriptions of the utilized chromatographic system into the prediction. We will thereby focus on the inclusion using a joint feature-vector for molecules and chromatographic systems.

In the following we assume, that for all pairs in the training set, i.e. $(i, j) \in \mathcal{P}$, the molecules m_i and m_j have been measured with the same chromatographic system. That means, no cross-system pairwise relations are explicitly measured. It furthermore motivates the use of $\mathbf{z}_{ij} = \mathbf{z}_i = \mathbf{z}_i$ as the notation for the chromatographic system feature descriptor corresponding to the pair (i, j).

3.3.1 Concatenating x and z

Concatenating the feature descriptors of the molecules and the chromatographic system is one option to include the descriptors in to prediction. For that, the kernel feature vector ϕ_i (see Eq. (1)) defined as:

$$\phi_i = \phi\left(\begin{bmatrix} \mathbf{x}_i \\ \mathbf{z}_i \end{bmatrix}\right)$$

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.