We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



11

Model Selection for Ranking SVM Using Regularization Path

Karina Zapien[1], Gilles Gasso[1], Thomas Gärtner[2] and Stéphane Canu[1]
[1] *LITIS EA 4108 - INSA de Rouen*
[2] *Fraunhofer IAIS,*
[1]*France*
[2]*Germany*

1. Introduction

This chapter deals with supervised learning problems under the ranking framework. Ranking algorithms are typically introduced as a tool for personalizing the order in which document recommendations or search results - in the web, for example - are presented. That is, the more important a result is to the user, the earlier it should be listed. To this end, two possible settings can be considered :

i. the algorithm tries to interactively rearrange the results of one search such that relevant results come the closer to the top the more (implicit) feedback the user provides,
ii. the algorithm tries to generalize over several queries and presents the results of one search in an order depending on the feedback obtained from previous searches.

The first setting deals with an active learning while the second setting deals with a passive supervised learning. This kind of problems have gain major attention given the nowadays amount of available informations. This is without doubt a challenging task in the medium and large scale context.

Several methods have been proposed to solve these problems. For the passive setting, the Rankboost algorithm (Freund et al. (2003)) is an adaptation from the Adaboost algorithm to the ranking problem. This is a boosting algorithm which works by iteratively building a linear combination of several "weak" algorithms to form a more accurate algorithm. The Pranking algorithm (Crammer & Singer (2001)) is an online version of the weighted algorithm. The SVRank and RankSVMalgorithms are the adaptation of the Support Vector machines for classification and regression, respectively, while the MPRank (Cortes et al. (2007)) is a magnitude-preserving algorithm, which searches not only to keep the relative position of each sample but also to preserve the distance given by the correct ordering. This last algorithm has as well the form of a regularization problem as the two previous with a different cost function.

Later, the Ranking SVM (RankSVM) algorithm was proposed by Herbrich et al. (2000) and Joachims (2002) as an optimization problem with constraints given by the induced graph of the ordered queries' results. This algorithm forms part of the family of kernel algorithms of the SVM type (Boser et al. (1992); Schölkopf & Smola (2002)).

Kernel methods like the SVM or the ranking SVM solve optimization problems of the form

Source: Machine Learning, Book edited by: Abdelhamid Mellouk and Abdennacer Chebira,
ISBN 978-3-902613-56-1, pp. 450, February 2009, I-Tech, Vienna, Austria

Open Access Database www.intechweb.org

$$\hat{f}_\lambda = \underset{f \in \mathcal{H}}{\operatorname{argmin}}\, \mathcal{V}(f) + \lambda \Omega(f) \tag{1}$$

where $\mathcal{V} : \mathcal{H} \to \mathbb{R}^+$ is a loss function, $\lambda \in \mathbb{R}^+$ is a regularization parameter, Ω: $\mathcal{H} \to \mathbb{R}^+$ is the regularizer (which allows to enforce some nice properties as smoothness or simplicity of f) and $\mathcal{H}$ represents the hypothesis space. Usually $\mathcal{H}$ is chosen as a reproducing kernel Hilbert space. Although a key bottleneck for applying such algorithms in the real-world is choosing λ, research often ignores this. As empirical results, however, strongly depend on the chosen λ, runtime intensive repeated cross-validations have to be performed. Hence, in this chapter we concentrate on speeding up and automating this choice by building on the *regularization path* for SVMs (Hastie et al. (2004)).

2. Piecewise linear solutions

This framework is a kind of a more generic regularized optimization problems, already studied for regularization problems (Rosset & Zhu (2007)) and for parametric quadratic programming (Markowitz (1959)) for portfolio optimization. We are interested by the efficient computation of the regularization path. Hence, let us define first this notion.

Definition 2.1 (Regularization path)

The regularization path of Problem (1) is the set of all solutions obtained when varying λ over $\mathbb{R}^+$ *i.e.* $Path = \{f_\lambda, with\ \lambda \in [0,+\infty]\}$.

As one can see, with this definition, the pursued policy can have a high computational price. In order to gain in efficiency, the family of piecewise linear solution path is of particular interest. To highlight this fact, we consider the following definition.

Definition 2.2 (piecewise linear solution path)

The solution path is said to be piecewise linear when there exists a strictly decreasing (or increasing) sequence λ^t, $t = 1, \ldots, N$ *such that* :

$$\hat{f}_\lambda = \hat{f}_{\lambda^t} + (\lambda - \lambda^t)\, h^t \qquad \forall \lambda \in [\lambda^{t+1}, \lambda^t] \ \ (resp.\ in\ [\lambda^t, \lambda^{t+1}]) \tag{2}$$

where h^t, $t = 1, \ldots, N$ *denotes a sequence of functions in* $\mathcal{H}$.

With such property, it is easy to efficiently generate the whole path of solution. Indeed, in such case, one only needs the sequence λ^t and the corresponding h^t. Any other functions in-between can be simply obtained by linear interpolation. Hence, owing to such property, the computational cost of obtaining the whole path of solution may be of the order of a single solution computation.

The question induced by this remark is to find which kind of objective functions makes the solution path piecewise linear. In Rosset and Zhu (2007), the necessary conditions were given for Problem (1) to admit a linear solution path. The main result is summarized by the theorem below.

Theorem 1

Assume the loss $\mathcal{V}(f)$ *and the regularizer* $\Omega(f)$ *are convex functions. If one objective function (either* $\mathcal{V}(f)$ *or* $\Omega(f)$*) is piecewise linear and the other one piecewise quadratic then the solution path of the Problem (1) is piecewise linear.*

Proof Assume $\mathcal{V}(f)$ and $\Omega(f)$) are twice differentiable in a neighborhood of f_{λ^t} solution of (1) corresponding to λ^t. Let also $\lambda = \lambda^t + \delta\lambda$ and its related solution f_λ. Consider finally $J(f) = \mathcal{V}(f_\lambda) + \lambda\,\Omega(f_\lambda)$. The optimality conditions associated to f_{λ^t} and f_λ are respectively

$$\nabla_f J(f_{\lambda^t}) = \nabla_f \mathcal{V}(f_{\lambda^t}) + \lambda^t \nabla_f \Omega(f_{\lambda^t}) = 0 \tag{3}$$

$$\nabla_f J(f_\lambda) = \nabla_f \mathcal{V}(f_\lambda) + \lambda \nabla_f \Omega(f_\lambda) = 0 \tag{4}$$

where $\nabla_f J(f)$ represents the functional derivative of J in $\mathcal{H}$. For small values of $\delta\lambda$ we can consider the following second order Taylor expansion of (4) around f_{λ^t}

$$\nabla_f \mathcal{V}(f_{\lambda^t}) + \delta\lambda\, f'_{\lambda^t} \nabla_f^2 \mathcal{V}(f_{\lambda^t}) + \lambda^t \left(\nabla_f \Omega(f_{\lambda^t}) + \delta\lambda\, f'_{\lambda^t} \nabla_f^2 \Omega(f_{\lambda^t})\right) + \delta\lambda \nabla_f \Omega(f_{\lambda^t}) = 0$$

with $f_\lambda = f_{\lambda^t} + \delta\lambda f'_{\lambda^t}$. Using it we have the following limit

$$\lim_{\delta\lambda \to 0} \frac{\nabla_f J(f_\lambda) - \nabla_f J(f_{\lambda_t})}{\delta\lambda} = f'_{\lambda^t} \nabla_f^2 \mathcal{V}(f_{\lambda^t}) + \lambda^t f'_{\lambda^t} \nabla_f^2 \Omega(f_{\lambda^t}) + \nabla_f \Omega(f_{\lambda^t}) = 0$$

that gives

$$f'_{\lambda^t} = -\left(\nabla_f^2 \mathcal{V}(f_{\lambda^t}) + \lambda^t \nabla_f^2 \Omega\,(f_{\lambda^t})\right)^{-1} \nabla_f \Omega(f_{\lambda^t})$$

The piecewise behavior is possible if f'_{λ^t} is constant. To fulfill this condition, it requires $\nabla_f^2 \Omega(f_{\lambda^t}) = 0$ (independence with respect to λ) and $\nabla_f^2 \mathcal{V}(f_{\lambda^t})$ to be constant. The latter condition is satisfied as the loss or the regularizer are assumed linear or quadratic. These requirements achieve the proof.

In fact, similar to SVM classification, it turns out that $\hat{f}_\lambda$ as a function of λ is piecewise linear and hence forms a *regularization path*. Indeed, in the RankSVM algorithm, the loss function $\mathcal{V}(f)$ is the hinge loss (which is a L_1 type-function) and the regularizer $\Omega(f)$ is chosen as a quadratic or L_1 function (see Figure 1). These choices therefore fulfill the requirements of the theorem.

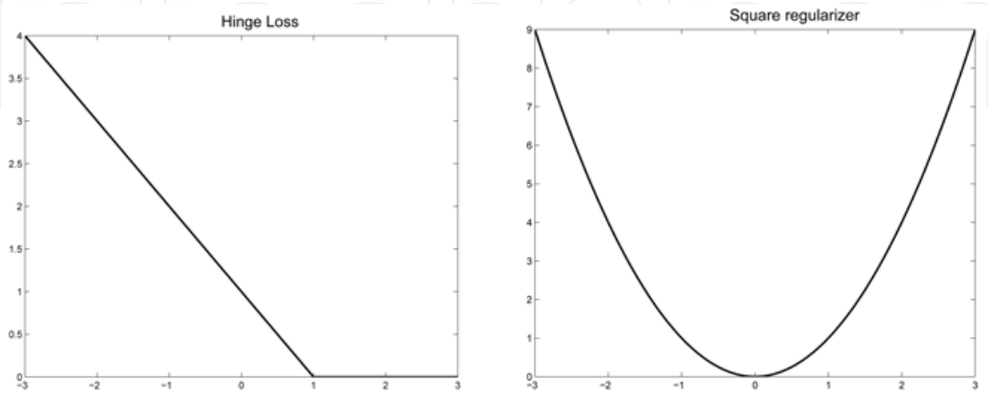


Fig. 1. Illustration of the typical choices of loss function and regularizer in SVM framework. Left) Hinge loss, Right) Square regularizer.

As in SVM classification, the breakpoints of this path correspond to certain events (described in more detail in Section 5). Points of the regularization path which are not breakpoints can not be distinguished in terms of margin-errors of the training data. To choose a particular regularization parameter, and hence a particular solution to the ranking problem, we evaluate $\hat{f}_\lambda$ on a validation set for each breakpoint of the regularization path.
Before delving into the details of solution path computation, the next two sections present the ranking SVM algorithm.

3. Ranking SVM

For clarity and simplification sakes, let consider the example of web pages search in ranking problems like (*i*) and (*ii*) from the introduction. To this purpose, we consider a set of query-document samples $x = (q, d) \in X$, together with a label y that induces a relative order or preference between the documents d accordingly to a query q. Each query induces a directed acyclic graph (X, E), with $E \subseteq X^2$ (See Figure 2).

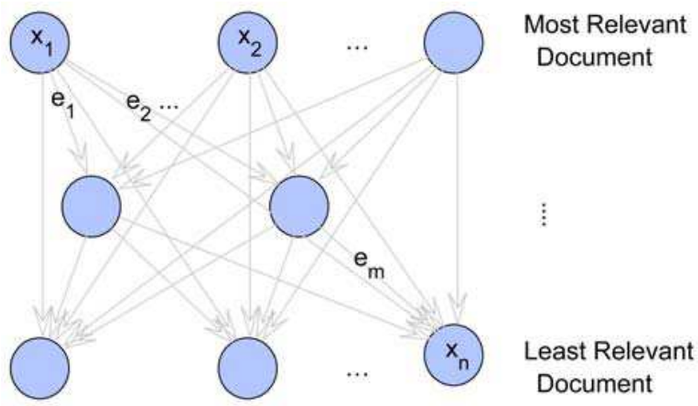


Fig. 2. Induced graph from ranking constraints for a particular query

For (*i*) the set of web pages forms the vertex set X of the digraph and we are also given some further information about the web pages (like a bag-of-words representation). For (*ii*) each vertex of the graph is a pair containing a query ($q \in \mathcal{Q}$) and a document ($d \in \mathcal{D}$). Hence, the vertex set is $X \subset \mathcal{Q} \times \mathcal{D}$ and edges of the form $((q, d), (q, d')) \in E$ with $d, d' \in \mathcal{D}$; $q \in \mathcal{Q}$ represent that d was more relevant than d' for an user asking query q. In addition one typically assumes some joint representations of queries and web pages.
The beauty of these problems is that classification and ordinal regression problems can be written as a ranking problem, therefore, the ranking SVM framework can be as well used for this kind of problems. The exact decision frontier can be calculated via a ROC curve, for example.
In both cases, ranking algorithms aim to find an ordering (permutation) of the vertex $\pi : X \to [\![n]\!]$ where $n = |X|$ and $[\![n]\!] = \{1, \ldots, n\}$ such that the more relevant a document is, the higher it is placed after the permutation, while as few as possible preferences are violated by the permutation.
Ranking SVM approaches such learning problems by solving the following primal optimization problem :

$$\begin{aligned} \hat{\mathrm{f}}_\lambda \;=\; & \underset{\mathrm{f}\in\mathcal{H}}{\operatorname{argmin}} \quad \xi^\top \mathbb{1} + \lambda \|\mathrm{f}\|_{\mathcal{H}}^2 \\ & \text{subject to :} \quad \mathrm{f}(v) - \mathrm{f}(u) \geq 1 - \xi_{vu} \qquad \forall (u,v) \in E \\ & \qquad\qquad \xi_{vu} \geq 0 \qquad\qquad\qquad\quad \forall (u,v) \in E\,. \end{aligned} \tag{5}$$

Here, $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), $\lambda \in \mathbb{R}^+$ is a regularization parameter, and the square norm $\|\cdot\|_{\mathcal{H}}^2$ in the Hilbert space serves as the regularizer. As in SVM for classification, the slack variables ξ_{vu}, $(u, v) \in E$ traduce the cost related to the violation of the constraints (u, v). The final permutation π is then obtained by sorting X according to f and resolving ties randomly.

Now, to easy the notation, let $k : X \times X \to \mathbb{R}$ be the reproducing kernel of $\mathcal{H}$ and denote the vertex by $\mathbf{x}_i$ such that $X = \{\mathbf{x}_i \mid i \in [\![n]\!]\}$. The set of violated constraints is $\{(\mathbf{x}_i, \mathbf{x}_j) \in E \mid \pi(\mathbf{x}_i) < \pi(\mathbf{x}_j)\}$. The decision function will have the form $\hat{\mathrm{f}}_\lambda(\cdot) = \sum_{i=1}^n \beta_i k(\mathbf{x}_i, \cdot)$ with $\beta_i \in \mathbb{R}$. With slight abuse of notation we write $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), ..., k(\mathbf{x}, \mathbf{x}_n))^{\mathrm{T}}$. Using this notation, a ranking problem (5) with m preferences $E = \{(\mathbf{x}_{k_i}, \mathbf{x}_{l_i}) \mid i \in [\![m]\!]\}$ can be written as :

$$\begin{aligned} \hat{\boldsymbol{\beta}}(\lambda) = \; & \underset{\boldsymbol{\beta}\in\mathbb{R}^n, \boldsymbol{\xi}\in\mathbb{R}^m}{\operatorname{argmin}} \quad \boldsymbol{\xi}^\top \mathbb{1} + \tfrac{\lambda}{2} \boldsymbol{\beta}^\top K \boldsymbol{\beta} \\ & \text{s. t.} \quad \boldsymbol{\beta}^\top (\mathbf{k}(\mathbf{x}_{k_i}) - \mathbf{k}(\mathbf{x}_{l_i})) \geq 1 - \xi_i \quad \forall i \in [\![m]\!] \\ & \qquad\; \xi_i \geq 0 \qquad\qquad\qquad\qquad\; \forall i \in [\![m]\!] \end{aligned} \tag{6}$$

with $K = [K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}^{n\times n}$ the Gram matrix and $\boldsymbol{\beta} = [\beta_1 \ldots \beta_n]^{\mathrm{T}}$.

The complexity of the problem comes from the fact that the number of such preference constraints m is of order the square of the training set size that is $m = O(n^2)$. The Lagrangian L of problem (6) is given by :

$$\mathcal{L} = \boldsymbol{\xi}^\top \mathbb{1} + \frac{\lambda}{2} \boldsymbol{\beta}^\top K \boldsymbol{\beta} - \sum_{i=1}^m \alpha_i \left(\boldsymbol{\beta}^\top \left(\mathbf{k}(\mathbf{x}_{k_i}) - \mathbf{k}(\mathbf{x}_{l_i}) \right) - 1 + \xi_i \right) - \sum_i^m \gamma_i \xi_i$$

with $\alpha_i \geq 0$, $\gamma_i \geq 0$. A matrix $P \in \mathbb{R}^{m\times n}$ can be defined with entries

$$P_{ij} = \begin{cases} +1 & \text{if } j = k_i \\ -1 & \text{if } j = l_i \\ 0 & \text{otherwise} \end{cases} \quad \Longrightarrow \quad PK = \begin{pmatrix} \mathbf{k}(\mathbf{x}_{k_1})^\top - \mathbf{k}(\mathbf{x}_{l_1})^\top \\ \mathbf{k}(\mathbf{x}_{k_2})^\top - \mathbf{k}(\mathbf{x}_{l_2})^\top \\ \vdots \\ \mathbf{k}(\mathbf{x}_{k_m})^\top - \mathbf{k}(\mathbf{x}_{l_m})^\top \end{pmatrix} \tag{7}$$

so that the Lagrangian can be expressed as :

$$\mathcal{L} = \boldsymbol{\xi}^\top \mathbb{1} + \frac{\lambda}{2} \boldsymbol{\beta}^\top K \boldsymbol{\beta} - \boldsymbol{\beta}^\top K P^\top \boldsymbol{\alpha} + \mathbb{1}^\top \boldsymbol{\alpha} - \boldsymbol{\xi}^\top \boldsymbol{\alpha} - \boldsymbol{\xi}^\top \boldsymbol{\gamma}$$

with $\boldsymbol{\alpha} \geq \mathbf{0}$, $\boldsymbol{\gamma} \geq \mathbf{0}$ (the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ contain respectively the Lagrange parameters α_i and γ_i). Using the Karush-Kuhn-Tucker (KKT) conditions, we obtain:

$$\frac{\partial \mathcal{L}}{\partial \xi} = 0 \Rightarrow 0 = \mathbb{1} - \boldsymbol{\alpha} - \boldsymbol{\gamma} \quad \text{together with} \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = 0 \Rightarrow 0 = \lambda K\boldsymbol{\beta} - KP^\top \boldsymbol{\alpha}.$$

These equations result in conditions, $0 \leq \alpha_i \leq 1$ and $\boldsymbol{\beta} = \frac{1}{\lambda} P^\top \boldsymbol{\alpha}$, so that

$$\hat{\mathrm{f}}_\lambda(\mathbf{x}) = \frac{1}{\lambda} \boldsymbol{\alpha}^\top P \mathbf{k}(\mathbf{x}).$$

Finally, the dual of Problem (6) is:

$$\begin{aligned} \hat{\boldsymbol{\alpha}}(\lambda) \quad = \quad & \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\operatorname{argmax}} \quad \boldsymbol{\alpha}^\top \mathbb{1} - \tfrac{1}{2\lambda} \boldsymbol{\alpha}^\top P K P^\top \boldsymbol{\alpha} \\ & \text{s.t.} \qquad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbb{1}. \end{aligned} \tag{8}$$

4. RankSVM singularity

As mentioned in the introduction, the Ranking SVM optimization problem induces a directed graph for each query. This structure constraints an edge for each relationship of relevance between samples that has to be satisfied. These constraints include as well all transitive relationships that could in fact be induced by other ones. This redundancy in the constraints setting cause the Hessian matrix in Problem (8) to be singular.

This issue can be overcome by designing for each query a sample as the maximum of all his rank for this query, so that edges from the chosen sample will be added to the other samples. For the immediate upper level, all samples in it will be joint to the maximum of the previous rank and so on. The obtained graph would look as in Figure (3).

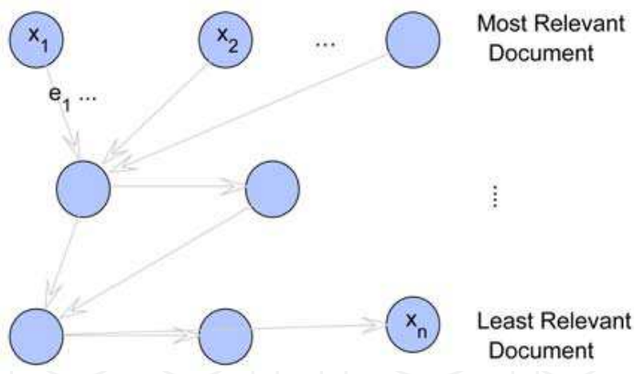


Fig. 3. New graph that will generate a non singular Hessian on the dual problem

The advantage of this new formulation is that the number of constraints is significantly smaller than in the original RankSVMalgorithm. The first one can be of order $O(n^2)$, while the second one is of order $O(n)$ This will lead to a smaller problem and faster training time with a consistent problem equivalence.

5. Regularization path for ranking SVM

Following the arguments developed in Rosset and Zhu (2007), it can be shown that the solution $\hat{\boldsymbol{\alpha}}(\lambda)$ of the above dual problem is a piecewise linear function of λ. Hence the

problem admits a piecewise linear regularization path. A regularization path has breakpoints $\lambda^1 > \lambda^2 > \ldots$ such that for an interval $[\lambda^{t+1}, \lambda^t]$ (i.e., with no breakpoint) the optimal solutions $\hat{\boldsymbol{\alpha}}(\lambda)$ and $\hat{\boldsymbol{\beta}}(\lambda)$ can easily be obtained for all $\lambda \in [\lambda^{t+1}, \lambda^t]$.

Following the work of Hastie et al. (2004) we now derive the regularization path of ranking SVM. For given λ, and to simplify the notations, let $\mathsf{f}(\mathbf{x})$ and $\boldsymbol{\alpha}$ be the decision function and the optimal solution for Problems (6) and (8), respectively (i.e. $\hat{\mathsf{f}}_\lambda(\mathbf{x}) \equiv \mathsf{f}(\mathbf{x})$ and $\hat{\boldsymbol{\alpha}}(\lambda) \equiv \boldsymbol{\alpha}$). Then, the following partition derived from the KKT optimality conditions can be made :

$$\begin{aligned}
\mathcal{I}_\alpha &= \{i \in [\![m]\!] \mid \mathsf{f}(\mathbf{x}_{k_i}) - \mathsf{f}(\mathbf{x}_{l_i}) = 1\} = \{i \in [\![m]\!] \mid 0 < \alpha_i < 1\}, \\
\mathcal{I}_0 &= \{i \in [\![m]\!] \mid \mathsf{f}(\mathbf{x}_{k_i}) - \mathsf{f}(\mathbf{x}_{l_i}) > 1\} = \{i \in [\![m]\!] \mid \alpha_i = 0\}, \quad \text{and} \\
\mathcal{I}_1 &= \{(i \in [\![m]\!] \mid \mathsf{f}(\mathbf{x}_{k_i}) - \mathsf{f}(\mathbf{x}_{l_i}) < 1\} = \{i \in [\![m]\!] \mid \alpha_i = 1\}.
\end{aligned}$$

The set $\mathcal{I}_0$ represents the satisfied constraints whereas $\mathcal{I}_1$ is devoted to the violated constraints and $\mathcal{I}_\alpha$ includes the "margin constraints".

Similarly, we will denote by $\boldsymbol{\alpha}^t$ and $\mathsf{f}^t(\mathbf{x})$ the optimal solution of the dual Problem (6) for the regularization parameter λ^t. Note that we assume the above sets $(\mathcal{I}_\alpha^t, \mathcal{I}_1^t, \mathcal{I}_0^t)$ induced by the solution of the optimization problem for λ^t remain unchanged for $\lambda \in [\lambda^{t+1}, \lambda^t]$, i.e., $(\mathcal{I}_\alpha^t, \mathcal{I}_1^t, \mathcal{I}_0^t) = (\mathcal{I}_\alpha, \mathcal{I}_1, \mathcal{I}_0)$. Hence, $\alpha_i \in \mathcal{I}_\alpha$ depends linearly on λ, This can be seen by writing $f(\mathbf{x})$ as follows :

$$\begin{aligned}
\mathsf{f}(\mathbf{x}) &= \left[\mathsf{f}(\mathbf{x}) - \frac{\lambda^t}{\lambda}\mathsf{f}^t(\mathbf{x})\right] + \frac{\lambda^t}{\lambda}\mathsf{f}^t(\mathbf{x}) \\
&= \frac{1}{\lambda}\left[(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^\top P\mathbf{k}(\mathbf{x}) + \lambda^t \mathsf{f}^t(\mathbf{x})\right] \qquad (9) \\
\mathsf{f}(\mathbf{x}) &= \frac{1}{\lambda}\left[(\boldsymbol{\alpha}_{\mathcal{I}_\alpha} - \boldsymbol{\alpha}_{\mathcal{I}_\alpha}^t)^\top P_{\mathcal{I}_\alpha}\mathbf{k}(\mathbf{x}) + \lambda^t \mathsf{f}^t(\mathbf{x})\right]
\end{aligned}$$

where the last line is true as $\alpha_i - \alpha_i^t$ for all $i \notin \mathcal{I}_\alpha$. $P_{\mathcal{I}_\alpha}$ is the submatrix of P containing the rows corresponding to $\mathcal{I}_\alpha$ and all columns. For all $i \in \mathcal{I}_\alpha$ we have that $1 = \mathsf{f}(\mathbf{x}_{k_i}) - \mathsf{f}(\mathbf{x}_{l_i}) = \mathsf{f}^t(\mathbf{x}_{k_i}) - \mathsf{f}^t(\mathbf{x}_{l_i})$, leading to

$$1 = \frac{1}{\lambda}\left[(\boldsymbol{\alpha}_{\mathcal{I}_\alpha} - \boldsymbol{\alpha}_{\mathcal{I}_\alpha}^t)^\top P_{\mathcal{I}_\alpha}(\mathbf{k}(\mathbf{x}_{k_i}) - \mathbf{k}(\mathbf{x}_{l_i})) + \lambda^t\right].$$

Therefore

$$\lambda - \lambda^t = (\boldsymbol{\alpha}_{\mathcal{I}_\alpha} - \boldsymbol{\alpha}_{\mathcal{I}_\alpha}^t)^\top P_{\mathcal{I}_\alpha}(\mathbf{k}(\mathbf{x}_{k_i}) - \mathbf{k}(\mathbf{x}_{l_i}))\ . \qquad (10)$$

This equation is valid for all pairs in $\mathcal{I}_\alpha$ for fixed sets $\mathcal{I}_\alpha, \mathcal{I}_0,$ and $\mathcal{I}_1$. It can be simplified by transposing Eq. (10) and using Eq. (7) in it, getting :

$$(\lambda - \lambda^t)\mathbb{1}_{\mathcal{I}_\alpha} = P_{\mathcal{I}_\alpha} K P_{\mathcal{I}_\alpha}^\top \left(\boldsymbol{\alpha}_{\mathcal{I}_\alpha} - \boldsymbol{\alpha}_{\mathcal{I}_\alpha}^t\right) \qquad (11)$$

If we define $\boldsymbol{\eta} = (P_{\mathcal{I}_\alpha} K P_{\mathcal{I}_\alpha}^\top)^{-1} \mathbb{1}_{\mathcal{I}_\alpha}$, with $\mathbb{1}_{\mathcal{I}_\alpha}$ a vector of ones of size $|\mathcal{I}_\alpha|$, then it can finally be seen that α_i, $i \in \mathcal{I}_\alpha$ changes piecewise linearly in λ as follows :

$$\alpha_i = \alpha_i^t - (\lambda^t - \lambda)\eta_i \qquad i \in \mathcal{I}_\alpha. \tag{12}$$

For all $\lambda \in [\lambda^{t+1}, \lambda^t]$, the optimal solution $\boldsymbol{\alpha}$ (and consequently the decision function f(**x**)) can be easily obtained until the sets change, i.e., an event occurs. From any optimal solution $\boldsymbol{\alpha}$ for λ, the corresponding sets $\mathcal{I}_\alpha$, $\mathcal{I}_0$, and $\mathcal{I}_1$ can be deduced and thereon the successive solutions for different λ.

5.1 Initialization

If λ is very large, $\boldsymbol{\beta} = \mathbf{0}$ minimizes Problem (6). This implies that $\xi_i = 1$, $\forall i$ and because of the strict complementary and KKT conditions, $\gamma_i = 0 \Rightarrow \alpha_i = 1$. To have at least one element in $\mathcal{I}_\alpha$, we need a pair $(\mathbf{x}_{k_i}, x_{l_i})$ that verifies $\boldsymbol{\beta}^\top(\mathbf{k}(x_{k_i}) - \mathbf{k}(x_{l_i})) = 1$. We know that $\boldsymbol{\beta} = \frac{1}{\lambda} P^\top \boldsymbol{\alpha}$ and therefore $\boldsymbol{\alpha} = 1$I solves , for all pairs, the equation

$$\lambda_{k_i,l_i} = \boldsymbol{\alpha}^\top P(\mathbf{k}(x_{k_i}) - \mathbf{k}(x_{k_i})).$$

Hence, initially all pairs will be in $\mathcal{I}_1$ and, as initial λ value, we take

$$\lambda^0 = \max\{\lambda_{k_i,l_i}\}.$$

The set $\mathcal{I}_\alpha$ will contain the pairs which maximize the value of λ_0.

5.2 Event detection

At step t the optimal solution $\boldsymbol{\alpha}^t$ defines a partition $\mathcal{I}_\alpha$, $\mathcal{I}_1$, $\mathcal{I}_0$. If these sets remain fixed for all λ in a given range then the optimal solution $\boldsymbol{\alpha}(\lambda)$ is a linear function of λ. If an event occurs, i.e., the sets change, then the linear equation has to be readjusted. Two types of events have to be determined:

- a pair in $\mathcal{I}_\alpha$ goes to $\mathcal{I}_1$ or $\mathcal{I}_0$
- a pair in $\mathcal{I}_1$ or $\mathcal{I}_0$ goes to $\mathcal{I}_\alpha$.

5.2.1 Pair in $\mathcal{I}_\alpha$ goes to $\mathcal{I}_1$ or $\mathcal{I}_0$

This event can be determined by analyzing at which value of λ the corresponding α_i turns zero or one. Eq. (12) is used and the following systems are solved for λ_i :

$$1 = \alpha_i^t - (\lambda^t - \lambda_i)\eta_i \qquad i \in \mathcal{I}_\alpha \tag{13}$$

$$0 = \alpha_i^t - (\lambda^t - \lambda_i)\eta_i \qquad i \in \mathcal{I}_\alpha. \tag{14}$$

Using these last equations, the exact values for λ_i that produces an event on pairs in $\mathcal{I}_\alpha$ moving to $\mathcal{I}_0 \cup \mathcal{I}_1$ can be determined.

5.2.2 Pair in $\mathcal{I}_1$ or $\mathcal{I}_0$ goes to $\mathcal{I}_\alpha$

To detect this event, note that Equation (11) can also be written as follows :

$$\begin{aligned}\left(\boldsymbol{\alpha}_{\mathcal{I}_\alpha} - \boldsymbol{\alpha}^t_{\mathcal{I}_\alpha}\right) &= (\lambda - \lambda^t)\left[\left(P_{\mathcal{I}_\alpha} K P^\top_{\mathcal{I}_\alpha}\right)^{-1} \mathbb{1}_{\mathcal{I}_\alpha}\right] \\ \left(\boldsymbol{\alpha}_{\mathcal{I}_\alpha} - \boldsymbol{\alpha}^t_{\mathcal{I}_\alpha}\right) &= (\lambda - \lambda^t)\boldsymbol{\eta}^\top.\end{aligned} \tag{15}$$

Plugging Eq. (15) in Eq. (9), we can write f(**x**) in a convenient manner:

$$\mathsf{f}(\mathbf{x}) = \frac{1}{\lambda}\left[\lambda^t \mathsf{f}^t(\mathbf{x}) + (\lambda - \lambda^t)\boldsymbol{\eta}^\top P_{\mathcal{I}_\alpha}\mathbf{k}(\mathbf{x})\right].$$

If we let $\mathsf{h}^t(\mathbf{x}) = \boldsymbol{\eta}^\top_{\mathcal{I}^t_\alpha} P_{\mathcal{I}_\alpha}\mathbf{k}(\mathbf{x})$, then

$$\mathsf{f}(\mathbf{x}) = \frac{1}{\lambda}\left[\lambda^t \mathsf{f}^t(\mathbf{x}) - \lambda^t \mathsf{h}^t(\mathbf{x}) + \lambda \mathsf{h}^t(\mathbf{x})\right] \tag{16}$$

An event on pair $(k_i, l_i) \in \mathcal{I}_0 \cup \mathcal{I}_1 \Vdash \mathcal{I}_\alpha$ means that $\mathsf{f}(\mathbf{x}_{k_i}) - \mathsf{f}(\mathbf{x}_{l_i}) = 1$ and can be detected by using Equation (16). The corresponding λ_i that generates this event is calculated as follows:

$$\lambda_i = \frac{\lambda^t\left[\left(\mathsf{f}^t(\mathbf{x}_{k_i}) - \mathsf{f}^t(\mathbf{x}_{l_i})\right) - \left(\mathsf{h}^t(\mathbf{x}_{k_i}) - \mathsf{h}^t(\mathbf{x}_{l_i})\right)\right]}{1 - \left(\mathsf{h}^t(\mathbf{x}_{k_i}) - \mathsf{h}^t(\mathbf{x}_{l_i})\right)} \tag{17}$$

Finally, λ^{t+1} will be the largest resulting $\lambda_i < \lambda^t$ from Equations (13), (14) and (17). In a cross validation framework, model selection can be done by learning the parameters in the training sets, an estimation of the generalization error (or ranking accuracy) can be taken by applying each model to the validation set.

The path computation is summarized by the pseudo-code of Algorithm 1.

5.3 Remarks and comments

Here we discuss briefly some issues of the algorithm related to the piecewise variation, the numerical complexity and how to address the emptiness of the set $\mathcal{I}_\alpha$.

On the functional piecewise variation

Let the function $g = \lambda f$ corresponding to the regularization parameter λ. In a similar manner, consider the function $g^t = \lambda^t f^t$ which corresponds to the solution for the value λ^t. From Eq. (16), one derives easily the relation $g = g^t + (\lambda - \lambda^t)h^t$. Therefore, we recover the piecewise linear variation stated in theorem 1. This linear variation formally concerns the function g instead of f. However the parameters $\boldsymbol{\alpha}$ involved in f evolves linearly with λ.

On the numerical complexity

The numerical complexity of the algorithm can be analyzed as follows. We assume the whole matrix $P\,K\,P^T$ is available beforehand as it can be built and stored at the beginning of

Algorithm 1 Pseudo-code of the RankingSVM path computation

Require: The set of m preferences $E = \{(\mathbf{x}_{k_i}, \mathbf{x}_{l_i}) \mid i \in [\![m]\!]\}$

Ensure: All solutions $(\mathsf{f}_\lambda,\ \lambda)$

Set $t = 0$

Compute the initial value λ^0 and estimation $\boldsymbol{\alpha}^0$.

Deduce the corresponding sets $(\mathcal{I}_\alpha^0, \mathcal{I}_1^0, \mathcal{I}_0^0)$

repeat

Compute the update direction of the parameters $\boldsymbol{\eta}^t = (P_{\mathcal{I}_\alpha^t} K P_{\mathcal{I}_\alpha^t}^\top)^{-1} \mathbb{1}_{\mathcal{I}_\alpha^t}$

Use $\boldsymbol{\eta}^t$ to detect all the necessary events and the corresponding regularization parameter values

Select the appropriate boundary value λ^{t+1}, save the event type (pair in $\mathcal{I}_1 \cup \mathcal{I}_0$ goes to $\mathcal{I}_\alpha$ or pair in $\mathcal{I}_\alpha$ goes to $\mathcal{I}_1$ or $\mathcal{I}_0$) and the related pair(s) of constraint

Update the parameters according to Eq. (12)

Update the sets $(\mathcal{I}_\alpha^t, \mathcal{I}_1^t, \mathcal{I}_0^t)$ according to the retained event and the concerned pair(s)

$t = t + 1$

until a termination criterion is reached

the algorithm and this computation requires $\mathcal{O}(mn^2)$ operations from the knowledge of the matrices P and K. At each iteration, solving the linear system (11) involves a cost of order $\mathcal{O}(|I_\alpha|^3)$. The calculation of all next values λ^{t+1} (using Eq. 13-14 and 17) has a numerical complexity of $\mathcal{O}(m|I_\alpha|)$ whereas the detection of the next event is of order $\mathcal{O}(m)$. Let $y_i = \mathsf{f}(\mathbf{x}_{k_i}) - \mathsf{f}(\mathbf{x}_{l_i})$ the evaluation of the preference $(\mathbf{x}_{k_i}, x_{l_i})$, $i \in [\![m]\!]$. According to (16), the update of all y_i is $\mathcal{O}(m)$. We can note that the computational complexity is essentially related to the cardinality of $I_\alpha|$. The cubic complexity of the linear system can be decreased to square complexity using a Sherman-Morrison rule to update the inverse of the matrix $P_{\mathcal{I}_\alpha} K P_{\mathcal{I}_\alpha}^\top$ or a Choleski update procedure. The exact complexity of the algorithm is hard to predict since the total number of events needed for exploring entirely the regularization path is data-dependent and the mean size of $|\mathcal{I}_\alpha|$ is difficult to guess beforehand. However, the total complexity is few multiples of the cost for solving directly the dual problem (8).

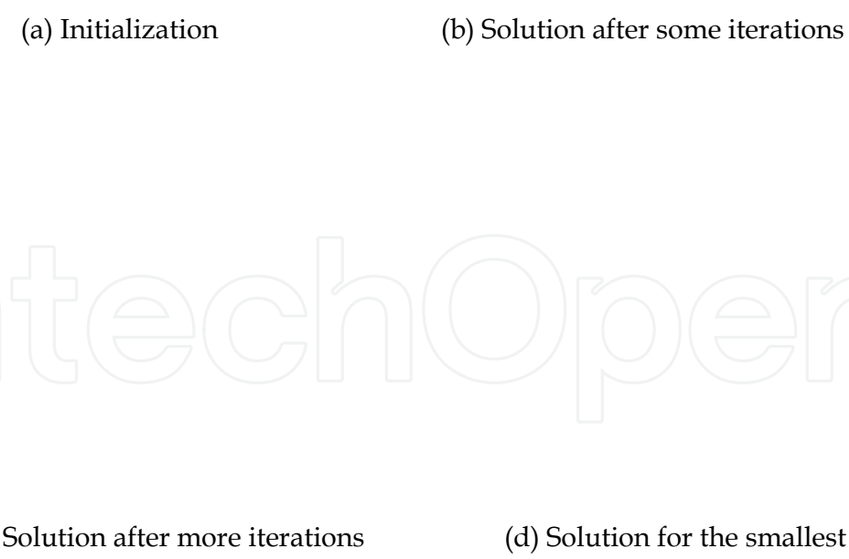
On the emptiness of $\mathcal{I}_\alpha$

It may happen during the algorithm that the set $\mathcal{I}_\alpha$ becomes empty. In such situation, a new initialization of the algorithm is needed. We apply the procedure developed in Subsection 5.1 except the fact we consider solely the pairs in $\mathcal{I}_1$ keeping unchanged the set $\mathcal{I}_0$.

6. Experimental results

Several datasets where used to measure the accuracy and time to process the regularization path for the RankSVM algorithm. Firstly, a toy example generated from Gaussian distributions (Hastie et al. (2001)) was applied. Some invetisgations on real life datasets taken from the UCI repository[1] are further presented.

The mixtures dataset of Hastie et al. (2004) was originally designed for binary classification with instances $\mathbf{x}_i$ and corresponding labels $y_i \in \{\pm 1\}$. However, it can be viewed as a ranking problem with $E = \{(\mathbf{x}_i, \mathbf{x}_j) \mid y_i > y_j\}$. It contains 100 positive and 100 negative points which would induce 10000 constraints. The regularization path was run on this dataset and a decision function was taken on zero. This decision boundary can still be improved by observing the generated ROC curve at each level. Figure (4) illustrates the decision function



(a) Initialization (b) Solution after some iterations

(c) Solution after more iterations (d) Solution for the smallest λ

Fig. 4. Illustration of the regularization path for the mixture dataset, all red points must be ranked higher than the blue points. As λ decreases, the margin gets smaller and the distance between pairs tends to be larger than one.

[1] http ://archive.ics.uci.edu/ml/datasets.html

for different breakpoints of the regularization path. The initial solution (a) is poor but after some iterations the results are improved as shown in subfigure (b). The most interesting solution is illustrated on subfigure (c) where almost constraints are satisfied.
The others datasets are regression problems and can also be viewed as ranking problems by letting $E = \{(\mathbf{x}_i, \mathbf{x}_j) \mid y_i > y_j\}$.
The number of induced constraints on the complete dataset and those obtained after following the graph design in Figure (3) are compared in Table 1.
For the experiments, a training, a validation and a test sets where built, being the last two of about half the size of the training set each. The number of involved features, training and test instances, and training and test constraints are summarized in Table 2.
Finally, the experiment was run 10 times, the error is measured as the percentage of misclassified samples. The size of A tells the number of support vectors and finally the time, is the average time (in seconds) to train a regularization path. The results are gathered in Table 3. We can see that the computation cost needed to obtain all possible soultions and their evaluation on test samples (in order to pick up the best one) is fairly cheaper making the approach particularly interesting.

7. Conclusions

Regularization parameter search for the ranking SVM can be efficiently done by calculating the regularization path. This approach calculates efficiently the optimal solution for all possible regularization parameters by solving (in practice) small linear problems. This approach has the advantage of overcoming local minimum of the regularization function. These advantages make the parameter selection considerably less time consuming and the obtained optimal solution for each model more robust.

Dataset	RanksSVM	Reduced RankSVM
Mixtures	10000	199
Auto	75245	656
Diabetes	134000	767
Cancer	75684	568

Table 1. Number of training instances under the original RankSVM and the ones obtained after the graph reformulation

Dataset	Features	Tr. Instances	Tr. Constraints	Test Instances	Test Constraints
Mixtures	2	99	100	50	50
Auto	7	305	196	98	98
Diabetes	8	383	384	192	192
Cancer	30	284	285	142	142

Table 2. Summary of the features of the training, validation and test sets

Dataset	Error	λ^*	Size A	Time
Mixtures	0.29 (0.12)	0.02 (0.04)	69 (18.48)	5.92 (0.41)
Auto	0.50 (0.31)	1.71 (4.37)	155.1 (50.54)	74.17 (15.95)
Diabetes	0.65 (1e-16)	0.77 (0.0003)	384 (0)	186.79 (240.33)
Cancer	0.63 (0)	0.79387 (1e-15)	285 (0)	0.27 (0.14)

Table 3. Obtained results by running the regularization path on the datasets described in Table 1. The results are averaged over 10 trials.

The numerical complexity of the algorithm depends on the number of iterations needed to explore the overall solution path and the mean size of $\mathcal{I}_\alpha$. At each iteration, a linear system is solved to get $\boldsymbol{\eta}$ which has complexity $\mathcal{O}(|\mathcal{I}_\alpha|^2)$. Empirically we observed that the number of iterations is typically only 2-3 times larger than the number of training pairs

Another key point is the determination of kernel hyper-parameter. This problem was not tackled here. However, one can seek to combine our regularisation path with the kernel parameter path developed in Gang Wang and Lochovsky (2007).

8. References

Boser, B. E., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152.

Cortes, C., Mohri, M., and Rastogi, A. (2007). An alternative ranking problem for search engines. In Demetrescu, C., editor, *WEA*, volume 4525 of *Lecture Notes in Computer Science*, pages 1–22. Springer.

Crammer, K. and Singer, Y. (2001). Pranking with ranking.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4 :933–969.

Gang Wang, D.-Y. Y. and Lochovsky, F. H. (2007). A kernel path algorithm for support vector machines. In *Proceedings of ICML'2007*.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5 :1391–1415.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning : Data Mining, Inference and Prediction*. Springer Verlag, New York.

Herbrich, R., Graepel, T., and Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. In Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*, pages 115–132, Cambridge, MA. MIT Press.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133– 142.

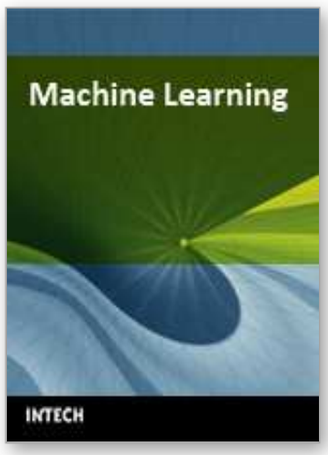
Markowitz, H. M. (1959). *Portfolio selection : Efficient diversification of investments*. John Wiley and Sons, Inc.

Rosset, S. and Zhu, J. (2007). Piecewise linear regularized solution paths. *Annals of Statistics*, 35(3) :1012–1030.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

IntechOpen

IntechOpen



Machine Learning

Edited by Abdelhamid Mellouk and Abdennacer Chebira

ISBN 978-953-7619-56-1
Hard cover, 450 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Machine Learning can be defined in various ways related to a scientific domain concerned with the design and development of theoretical and implementation tools that allow building systems with some Human Like intelligent behavior. Machine learning addresses more specifically the ability to improve automatically through experience.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Karina Zapien, Gilles Gasso, Thomas Gärtner and Stéphane Canu (2009). Model Selection for Ranking SVM Using Regularization Path, Machine Learning, Abdelhamid Mellouk and Abdennacer Chebira (Ed.), ISBN: 978-953-7619-56-1, InTech, Available from: http://www.intechopen.com/books/machine_learning/model_selection_for_ranking_svm_using_regularization_path

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen