Type of document Technical Report	*Classification* Confidential	*Document number* 9ADB003856-005	*Rev.* 0.8
Project Title Design Optimization for CHTUS	*CRID* 10656	*Date* November 17, 2014	*Pages* 26

Place Dä	*from Dept.* CH-RD.C1	*Authors* Jan Poland, Achin Jain and Kwok-Kai So (CHTUS)

Document Title

Ordinal Regression for Meta-Modeling in Optimization

Place	*to Dept.*	*Distributed to*

Summary

This report studies ordinal regression models, in particular large margin rank learners, and their use as meta-models for optimization of expensive and noisy functions. After introducing the topic and reviewing the role of regression and ordinal regression in meta-model based optimization, a new large margin rank learner, the *p5 model*, is introduced and compared to established approaches. The second part of the report is dedicated to experimental studies with meta-model based optimization, comparing the different learners. The cases range from toy problems to an application from turbocharger design, based on Computational Fluid Dynamics simulations.

Keywords: Meta-model based optimization, surrogate based optimization, black-box optimization, response surface, ordinal regression, Support Vector Machine, design optimization, turbo charger, gas inlet casing, CFD, computational fluid dynamics

Contents

1 Introduction

Optimization of expensive and noisy functions is one of the common challenges in applied optimization. Since engineers are frequently using numerical simulation like Finite Element Methods (FEM), Computational Fluid Dynamics (CFD), etc. in order to virtually prototype their systems, numerical design optimization of these virtual prototypes is the next step to advance the engineering tool chain. Depending on the complexity and degree of fidelity, these virtual prototypes are often expensive to evaluate, taking computation times from minutes up to months even on modern parallel hardware. There are other applications where expensive functions need to be optimized, e.g. when a function evaluation involves physical measurements at a test bed, or building a throw-away prototype, or treating a patient, etc.

Using established optimization approaches like gradient based optimization to these kinds of problems typically fails in practice. This has two reasons: (1) most "classical" optimization approaches (methods as described in [1]) are sensitive to noise, and (2) they are not designed for expensive function evaluations. On the other hand, they are designed for good convergence properties, a feature that is barely interesting for the applications mentioned. A solution which is just even close to a local optimum is usually a great achievement, often some reasonable improvement over the initial solution is sufficient in practice.

Response surface methods for optimization base on a different approach: From a set of function evaluations (*experimental design*), a statistical *meta-model* is constructed (*trained*). The meta-model is used to predict optimal or good solutions. Meta-model based optimization has its roots in the statistical literature about the early 1950s [2]. Since then, a significant amount of research has been devoted to the design of meta-model based optimization strategies with various working principles [3], [4], [5], [6], [7], [8], [9], [10], [11] (see Section 2.9 for details). Sophisticated algorithmic designs have been proposed that make optimal use of the information gained from function evaluations [5], [11], [4], [8].

1.1 Manual application of meta-model based optimization

In practice, a manual approach to meta-model based optimization is often applied. Starting from an experimental design, a meta-model is trained, often a polynomial regression or an Artificial Neural Network. Then the meta-model's predicted optima are located and validated. If the result is not yet satisfactory, the design space is narrowed down, extended, or otherwise changed, and the process is iterated.

This manual procedure has some advantages over automatic approaches:

- The user / engineer may change the parametrization of the design on the fly. Often it is convenient to start with a small parameter space and extend it according to the results of the simulations done so far, while defining a big parameter space in the beginning would be harder (e.g. because of conflicting parameter extensions).

- The user has much better control of the optimization process. In particular, he can early correct errors and detect and avoid "un-physical" solutions, that are solutions suggested by the optimization which are infeasible because of a factor which was not modelled. Also, he can decide between multiple optima based on additional expert knowledge not contained in the model.
- The user obtains more insight into characteristics of the problem, in particular sensitivities, which often gives valuable hints of how to improve the parametrization or even fundamentally change the design in a favorable way.

The focus of this work is the meta-models' predictive performance w.r.t. optimization, for a single iteration of training-optimizing-validating. As argued above, this is very relevant for practical applications. In addition, this focus facilitates study of the meta-models' predictive qualities in a clearer way than within the standard algorithmic framework of surrogate based optimization. Since only limited theoretical statements about predictive performance can be made for the models studied, the focus will be on empirical evaluation via computer experiments.

1.2 Outline

This work focuses on *ordinal regression* for meta-model based optimization. Except for recent work in the area of Evolutionary Computation [7], [8], [9], most meta-models studied in literature for optimizing expensive and noisy functions are common regression models, often Gaussian process models [3], [4], [5], [11]. Section 2, deals with ordinal regression, in particular large margin rank learners. A popular and simple instance, the *rank SVM*, is introduced in Section 2.3. Section 2.4 will propose a new large margin learner for ordinal regression, the *p5 model*, which has some advantages over existing models. In Section 2.5, the dual Quadratic Programs of rank SVM and p5 model will be compared, they turn out to be closely related. The remainder of Section 2 covers various aspects like choice of the kernel, cross validation, etc. Section 3 is dedicated to experimental studies. Starting variants of the sphere function in Section 3.2, we proceed to more realistic toy problems based on Finite Element simulation (Sections 3.3 and 3.4). Results from an application in turbocharger development based on Computational Fluid Dynamics simulation are given in Section 3.5. Section 4 summarizes and concludes.

2 Ordinal regression

We speak of *ordinal regression* when approximating a function f with a function g in an order preserving way. Let $\mathcal{X}$ be the space on which f is defined, this work restricts to $\mathcal{X} \subset \mathbb{R}^d$. $\mathcal{X}$ will be referred to as the *data space* since it is the domain of the training data (see Section 2.3), or as the *parameter space* or *search space* since parametrization of the underlying optimization problem is defined on $\mathcal{X}$. Then, an ordinal regression $g : \mathcal{X} \to \mathbb{R}$ for $f : \mathcal{X} \to \mathbb{R}$ is trained to satisfy $g(\tilde{x}) \leq g(x)$ whenever $f(\tilde{x}) \leq f(x)$ for the largest possible set of pairs $(x, \tilde{x})$.

The quality of an ordinal regression model is not changed by a monotonic transformation of its output. This is different for a common regression model $\tilde{f} : \mathcal{X} \to \mathbb{R}$ which is trained to be close to f, usually in the least squares sense (i.e. $\int_{\mathcal{X}} \|\tilde{f} - f\|^2$ should be small).

The natural success criterion of optimization, namely if an optimum has been attained or not, is invariant under monotonic transformation [12]. Yet, ordinal regression has been proposed for optimization only relatively recently [9], [7]. In contrast, there is a large body of literature on optimization based on common regression meta-models. Ordinal regression models extract information from data in a different way than common regression, and being neutral to monotonic transformations may avoid overfitting to the training data. In practice it always depends on the characteristics of the application if regression, order regression or another tool is going to deliver the best performance.

2.1 How much quantitative information should be used

The argument that optimization is invariant under monotonic transformation and hence also meta-models should be, is plausible but short-sighted to a certain extent. In case that a common regression model is a good or even perfect fit (e.g. a quadratic regression), the full quantitative information in the function evaluations is valuable. Just when the regression model is a less good fit, ordinal regression may be the stronger option.

Lack of fit may be also because of the noise[1] part of the model. Standard regression models implicitly assume Gaussian noise, which is often present in physical measurements, sensor data, etc. In different situations, noise may exhibit different characteristics. For instance, a part of the available data may be completely corrupted. In numerical simulations like FEM, CFD, etc., *computational noise from meshing* often occurs: These methods are in most cases based on irregular meshes, which are generated for the shapes to be simulated. If these shapes are subject to optimization, then discontinuities in the meshing algorithms introduce significant noise on the function evaluations, which may be even amplified by other artifacts of the numerical simulation like solver convergence and tolerances. An examples for such noise is shown in Fig. 7. With any kind of noise, one important function of a meta-model in optimization is reduction / filtering of the noise.

Even if ordinal regression is preferred over common regression, this does not necessarily imply that just order information is used for modeling and all further quantitative information is thrown away. In (5), we will introduce a way to still use quantitative information, which intuitively corresponds to cases with noise of limited amplitude.

[1] In the absence of noise, derivative free optimization [13] is a strong alternative.

2.2 Large margin learners for ranking

This work focuses on large margin learners for ordinal regression. Large margin learners have gained high popularity with the Support Vector Machine (SVM) [14]. They operate by establishing a geometric maximization criterion in the linear data space $\mathcal{X}$. Often, instead of working in the original data space $\mathcal{X}$, the data is mapped into a feature space $\tilde{\mathcal{X}}$. The corresponding (linear or nonlinear) mapping $\Phi : \mathcal{X} \to \tilde{\mathcal{X}}$ usually need not be explicitly known, but is used implicitly via a *kernel*, which is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that any generated Gram matrix is positive semi-definite [14]. In order to enable the kernel-based computation, the geometric margin maximization problem is not solved directly, but transformed to its *dual problem* (see Section 2.5) which contains the data only in form of pairwise inner products.

2.3 Rank SVM

One of the earliest large margin learners for ranking has been described in [15]. It is the rank learner that has been most widely used with evolutionary computation [9], [7]. Following [7], it is referred to as *rank SVM* here. The following paragraph introduces a slight generalization of the variant in [7].

Let training data $(x_i, y_i)_{i=1}^N$ be given, $x_i \in \mathcal{X} \subset \mathbb{R}^d$ and the values y_i are the function evaluations (i.e. objective function evaluations in case of an optimization, but the learner can be used generally outside the framework of optimization). Assume that the indexing is such that $y_1, y_2, \ldots$ are in non-descending order. The best and worst solutions are grouped into two sets $\mathcal{S}^{\text{best}} = \{1, \ldots, i^{\text{best}}\}$ and $\mathcal{S}^{\text{worst}} = \{i^{\text{worst}}+1, \ldots, N\}$. The training of the rank SVM (and later the p5 model as well) is going to disregard rank differences within these sets. For optimization, small rank differences are essential, hence trivial $\mathcal{S}^{\text{best}} = \{1\}$ is always used. $\mathcal{S}^{\text{worst}}$ may be non-trivial, i.e. it may be larger than the trivial $\mathcal{S}^{\text{worst}} = \{N\}$.

For notational convenience when dealing with $\mathcal{S}^{\text{best}}$ and $\mathcal{S}^{\text{worst}}$, two index mappings j^+ and j^- are introduced as

$$j^+(i) = \begin{cases} i^{\text{best}} + 1 & \text{if } i \in \mathcal{S}^{\text{best}} \\ i + 1 & \text{otherwise} \end{cases} \quad \text{and} \quad j^-(i) = \begin{cases} i^{\text{worst}} & \text{if } i \in \mathcal{S}^{\text{worst}} \\ i - 1 & \text{otherwise.} \end{cases} \tag{1}$$

In the trivial case of a one-element set $\mathcal{S}^{\text{best}}$ or $\mathcal{S}^{\text{worst}}$, the corresponding index maps are $j^+(i) = i + 1$ and $j^-(i) = i - 1$, respectively.

The rank SVM learner is defined by maximizing the margin of the pairwise differences of subsequent data points, possibly transformed into a feature space. By a standard argument

[14], this is equivalent to solving the following Quadratic Program (QP)[2]:

$$\min_{w\in\mathbb{R}^{\tilde{d}},\xi\in\mathbb{R}^{N-1}} \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{N-1}\xi_i \tag{2}$$

$$\text{s.t. } \xi_i \geq 0 \text{ for all } 1 \leq i \leq N-1 \tag{3}$$

$$\langle w, \Phi(x_{j^+(i)}) - \Phi(x_{j^-(i+1)})\rangle \geq \delta_i - \xi_i \text{ for } 1 \leq i \leq N-1. \tag{4}$$

Here, w is the weight vector to be optimized in the feature space $\tilde{\mathcal{X}} = \mathbb{R}^{\tilde{d}}$ (working in the original space with Φ being the identity is possible), and ξ_i are the slack variables which are introduced to allow for violation of the pairwise ranking. C is the penalty for the slack variables, often chosen to be a large value, e.g. $C = 1000$ (see also Section 2.8).

The desired margins are stored in $\delta_i \geq 0$ ($1 \leq i < N$). For a ordinal regression, usually $\delta_i = 1$ uniformly for all i is chosen. Alternatively, in this work also the values individually as

$$\delta_i = y_{j^+(i)} - y_{j^-(i+1)} \tag{5}$$

(equivalent to $\delta_i = y_{i+1} - y_i$ except for indices in $\mathcal{S}^{\text{best}}$ and $\mathcal{S}^{\text{worst}}$) are used. Thus deviations from the ideal ranking for function values which are close together are encouraged. This is reasonable since the ideal ranking is usually corrupted by noise, in particular when the noise has small amplitude (see also the discussion in Section 2.1).

The rank SVM learner is illustrated in Fig. 1 (left) for a small 4-points training set in 2D, trained with $C = 1$. There, one slack variable is needed to correct the mis-ranked pair $2 \rightarrow 3$.

2.4 P5 model

The rank SVM is trained to establish a large margin on the pairwise differences of the data $\Phi(x_i) - \Phi(x_{i+1})$. In contrast, the common SVM for binary classification is formulated to maximize the margin of the training data relative to a given separating hyperplane in the feature space, specified by an offset b. Here, the slack variables introduced into the QP correspond to individual data points instead of pairwise differences. Rank learners based on this principle have been suggested in [16], [17], [18]. In order to distinguish K different rank classes, $K-1$ separating hyperplanes are required, defined by offsets $b_1, b_2, \ldots, b_{K-1}$. For a data point x_i, two slack variables ξ_i and η_i are introduced in order to correct the projection of the point onto the maximal margin vector w both wrt. the next higher and the next lower rank, if necessary. However, the formulations proposed so far *decouple* these two slack variables: $\langle w, \Phi(x_i)\rangle - b_{k+1} \geq 1 - \xi_i$ and $\langle w, \Phi(x_i)\rangle - b_k \leq -1 + \eta_i$ is requested in equation [17]equation (3). A problem of this formulation is that the b_k

[2] In order to keep the terminology unambiguous, the term *optimization* is reserved for the initial task of meta-model based optimization. The step of solving a minimization problem in order to train a large margin learner is always referred to as Quadratic Programming, short QP.

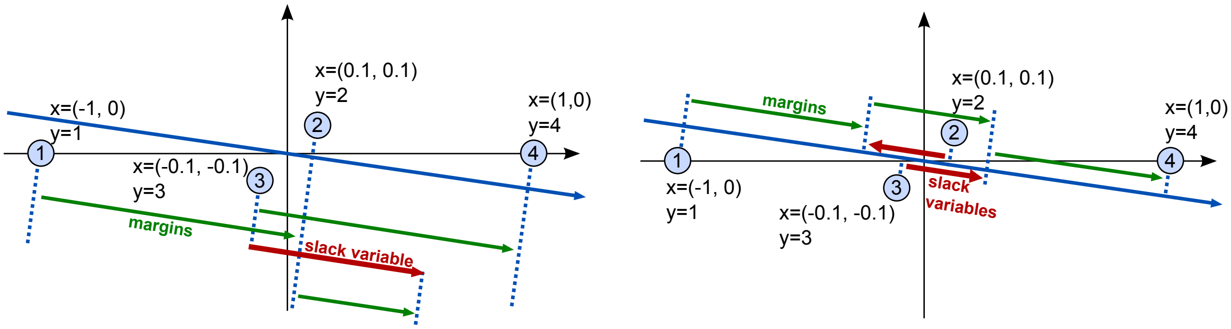


Figure 1: Rank SVM (left) and p5 model (right). Separation is achieved along the blue vector, the green arrows show the margins. In the solution obtained by the rank SVM, one of the pairs 2 → 3 needs to be corrected by a slack variable (red arrow) in order to achieve the required margins. In contrast, the p5 model solution corrects the positions of both points 2 and 3.

may be incorrectly ordered [17]. Correct ordering of the separating hyperplanes can be guaranteed if all pairwise relations between two data points have are included into the QP, instead of just those of rank adjacent data points. However, this significantly increases the size and complexity of the QP.

Instead of decoupling, we suggest to couple ξ_i and η_i as specified below (10)-(12). We call this formulation of a large margin rank learner the *precedence preserving pairwise partitioning pointer model*, short *p5 model*. Training data is given as $(x_i, y_i)_{i=1}^N$, sorted such that $y_1, y_2, \ldots, y_N$ are non-decreasing. Sets $\mathcal{S}^{\text{best}} = \{1, \ldots, i^{\text{best}}\}$ and $\mathcal{S}^{\text{worst}} = \{i^{\text{worst}} + 1, \ldots, N\}$ of cardinality ≥ 1 are admitted, and training is formulated such that rank differences within these sets are disregarded. The same index maps j^+ and j^- as defined in (1) are used again. The QP corresponding to the p5 model now reads as follows:

$$\begin{aligned}
&\min_{w,b,\xi,\eta} \tfrac{1}{2}\|w\|^2 + C\sum_{i=2}^{N}\xi_i + C\sum_{i=1}^{N-1}\eta_i && (6)\\
&\text{s.t.} \qquad \xi_i \geq 0 \text{ for all } 2 \leq i \leq N && (7)\\
&\qquad\qquad \eta_i \geq 0 \text{ for all } 1 \leq i \leq N-1 && (8)\\
&\langle w, \Phi(x_1)\rangle - \eta_1 \leq b_{j^+(1)-1} - \tfrac{1}{2}\delta_1 && (9)\\
&\langle w, \Phi(x_i)\rangle + \xi_i - \eta_i \leq b_{j^+(i)-1} - \tfrac{1}{2}\delta_i \text{ for all } 2 \leq i \leq N-1 && (10)\\
&\langle w, \Phi(x_i)\rangle + \xi_i - \eta_i \geq b_{j^-(i)} + \tfrac{1}{2}\delta_{i-1} \text{ for all } 2 \leq i \leq N-1 && (11)\\
&\langle w, \Phi(x_N)\rangle + \xi_N \geq b_{j^-(N)} + \tfrac{1}{2}\delta_{N-1}. && (12)
\end{aligned}$$

Here, $w \in \mathbb{R}^{\tilde{d}}$ and $\xi, \eta \in \mathbb{R}^{N-1}$, $b \in \mathbb{R}^{N+1-|\mathcal{S}^{\text{best}}|-|\mathcal{S}^{\text{worst}}|}$, and for notational convenience, indexing of ξ starts at 2 and indexing of b starts at i^{best}. C is the penalty for the slack variables, often chosen to be a large value, e.g. $C = 1000$ (see also Section 2.8). The desired margins are stored in $\delta_i \geq 0$ $(1 \leq i < N)$. As for the rank SVM, they may be chosen uniformly to be 1 or individually according to (5). For modeling an objective function, always $\mathcal{S}^{\text{best}} = \{1\}$.

The p5 model is illustrated in Fig. 1 (right) for a small 4-points training set in 2D, trained with $C = 1$. Both points in the middle need to be corrected with the corresponding slack variables in order to satisfy the constraints of the QP.

In contrast to existing support vector ordinal regression [16], [17], the b_i of a trained p5 model are always correctly ordered, without introducing additional constraints to force them to be so as in [17].

Proposition 1 *If the (7)-(12) are satisfied, then* $b_{i^{best}} \leq b_{i^{best}+1} \leq \ldots$ *always holds.*

Proof: The assertion $b_i \leq b_{i+1}$ follows directly by chaining the pairs of inequalities (10) and (11) for an index $i \geq i^{\text{best}}$, since the l.h.s. of these constraints are equal and all $\delta_i \geq 0$. This is true for all $i \leq N - |\mathcal{S}^{\text{best}}| - |\mathcal{S}^{\text{worst}}|$. □

2.5 Dual problems

Transforming a QP for training a large margin learner into its dual is a standard technique [14]. It has two advantages: In the dual QP, the training data occurs only in the form of inner products. Thus, no explicit computation in the feature space $\tilde{\mathcal{X}}$ induced by the mapping Φ is necessary, but this mapping is implicitly dealt with via a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Secondly, the dual QP is often even simpler than the primal QP.

The dual QP is established by maximizing the Lagrangian of the primal QP. For the rank SVM, the Lagrangian of (2)-(4) is

$$\mathcal{L} = \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{N-1}\xi_i - \sum_{i=1}^{N-1}\beta_i\xi_i - \sum_{i=1}^{N-1}\alpha_i\Big(\langle w, \Phi(x_{j^+(i)}) - \Phi(x_{j^-(i+1)})\rangle - \delta_i + \xi_i\Big) \quad (13)$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are the Lagrangian variables and the last two sums in the r.h.s. come from the constraints (3) and (4). At the QP solution, derivatives w.r.t. all variables must be zero. From $\frac{d\mathcal{L}}{d\xi_i} = 0$, it follows immediately that $\alpha_i \leq C$ for all $1 \leq i \leq N-1$. Similarly, $\nabla_w \mathcal{L} = 0$ implies

$$w = \sum_{i=1}^{N-1}\alpha_i\big(\Phi(x_{j^+(i)}) - \Phi(x_{j^-(i+1)})\big). \quad (14)$$

After all possible simplifications, the dual QP for the rank SVM summarizes to

$$\min_{\alpha} \tfrac{1}{2}\alpha^T K\alpha - \delta^T\alpha \quad (15)$$

$$\text{s.t. } 0 \leq \alpha_i \leq C \text{ for all } 1 \leq i \leq N-1. \quad (16)$$

Here, $K \in \mathbb{R}^{(N-1)\times(N-1)}$ is the Gram matrix having the entries

$$\begin{aligned} K_{i,j} &= \langle \Phi(x_{j^+(i)}) - \Phi(x_{j^-(i+1)}, \Phi(x_{j^+(j)}) - \Phi(x_{j^-(j+1)}\rangle & (17) \\ &= \langle \Phi(x_{j^+(i)}), \Phi(x_{j^+(j)})\rangle - \langle \Phi(x_{j^+(i)}), \Phi(x_{j^-(j+1)})\rangle - & (18) \\ & \quad \langle \Phi(x_{j^-(i+1)}), \Phi(x_{j^+(j)})\rangle + \langle \Phi(x_{j^-(i+1)}), \Phi(x_{j^-(j+1)})\rangle. & (19) \end{aligned}$$

The dual QP for the p5 model can be derived in the same way. The Lagrangian contains more terms, and the algebraic manipulations are a bit more lengthy. The resulting simplified QP however strongly resembles the dual QP of the rank SVM (15)-(16), with K being the same Gram matrix (17):

$$\min_{\alpha} \tfrac{1}{2}\alpha^T K \alpha - \delta^T \alpha \qquad (20)$$

$$\begin{aligned} \text{s.t.} \qquad \alpha_i &\geq 0 \text{ for all } 1 \leq i \leq N-1 & (21) \\ \alpha_i &\leq C \text{ if } i \in \mathcal{S}^{\text{best}} \text{ or } i+1 \in \mathcal{S}^{\text{worst}} & (22) \\ \alpha_{i+1} - \alpha_i &\leq C \text{ if } i+1 \notin \mathcal{S}^{\text{best}} \text{ and } i \notin \mathcal{S}^{\text{worst}} & (23) \\ \alpha_i - \alpha_{i+1} &\leq C \text{ if } i+1 \notin \mathcal{S}^{\text{best}} \text{ and } i \notin \mathcal{S}^{\text{worst}}. & (24) \end{aligned}$$

A solution to the dual QP delivers only the Lagrangian multipliers α_i, the other primal variables w, b, ξ, η are not directly recovered. However, for our purpose of evaluating the rank model, knowledge of the α_i is sufficient. For any $x \in \mathcal{X}$, $\langle w, \Phi(x)\rangle$ can be evaluated by means of the kernel function and (14).

In order to practically solve the dual QP, the experimental studies in this work make use of IpOpt [19].

2.6 Kernel choice

Kernel choice is a important degree of freedom with SVM related methods. Choosing an appropriate kernel for the application at hand has the potential significantly improve the performance. Here, this opportunity is explored only to a limited extent for meta-model based optimization. The remainder of this work restricts to inhomogeneous polynomial kernels in $\mathbb{R}^d$,

$$k(x,y) = \big(1 + \langle x, y\rangle\big)^p \text{ for } x, y \in \mathbb{R}^d, \qquad (25)$$

where p is the polynomial degree of the kernel.

Proposition 2 *If $f : \mathbb{R}^d \to \mathbb{R}$ is a polynomial function of degree p and $g : \mathbb{R} \to \mathbb{R}$ is a monotonic function, then a rank SVM or a p5 model with inhomogeneous polynomial kernel of degree p can perfectly represent $g \circ f : \mathbb{R}^d \ni x \mapsto g(f(x))$.*

Proof: It is well-known that the polynomial kernel of degree p corresponds to a feature space that contains all monomials up to degree p. Hence there are rank SVM and p5 model using this kernel which perfectly represent f. Since g is monotonic, the same learners also perfectly represent $g \circ f$. □

Depending on the way models are used in a meta-model based optimization, different polynomial degrees p may seem reasonable. If one tries to reproduce the behavior of a first order method, one may choose $p = 1$. This choice has the advantage that little data is necessary to robustly train first order models. In our experiments, we found however that $p = 1$ did not perform well with the way we used the models (compare Section 2.9).

A quadratic kernel ($p = 2$) performs better in our experiments. Because of Proposition 2, a rank learner using this kernel can in principle perfectly represent a second order approximation of the function to be optimized. Also higher order kernels are interesting. Cubic polynomial regression is often used in practical applications. In the following experiments, up to 4th order polynomial kernels have been used.

2.7 Cross validation

In the experimental part of the report, cross validation is studied as a tool for model / kernel selection. Since computation time for training is considered to be of minor importance here, always *leave one out cross validation* (loo-cv) is used, where for a data set of size N, N different models are trained with in turn one data point taken away from the training set. The loo-cv score is computed by evaluating each model on the removed data point and averaging over the N cases.

In the experiments, two versions of the cross validation will be compared:

- common loo-cv for regression, where the cross validation score is evaluated as $\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{m}_i(x_i) - y_i\|^2$ with $\boldsymbol{m}_i$ being the model trained without (x_i, y_i),
- order loo-cv, where the score is $\frac{1}{N}\sum_{i=1}^{N} |rank(\boldsymbol{m}_i(x_i)) - rank(y_i)|$ and $rank(\cdot)$ denotes the rank within the corresponding sets $\{\boldsymbol{m}_i(x_j), 1 \le j \le N\}$ and $\{y_j, 1 \le j \le N\}$.

The experiments presented later indicate that order cross validation is the better tool than standard cross validation for predicting the meta-model's performance in optimization. However, there are cases encountered where the quality of this prediction is highly questionable. Any further opportunities to modify the order cv score, e.g. by weighting of data with good rank stronger than data with bad rank, is not explored in this work.

2.8 Balanced choice of the slack variables penalty C

In the standard SVM, the penalty C on the slack variables is usually chosen to be a large value, but not too large in order not to affect the robustness of QP solving. In practice,

$C = 1000$ is a common choice giving good results for many applications. This is also a good choice for large margin rank learners and will be used in all experimental studies in Section 3. Here, a different strategy to choose C for the p5 model (or generally rank learners) is briefly discussed.

By construction (5) and (6)-(12), the QP becomes invariant under change of scaling of the x-part or y-part of the training data if

$$C = C_0 \frac{\sigma_y}{\sigma_{\tilde{X}}^2} \tag{26}$$

is chosen. Here, σ_y is the scaling of the y-part of the training data, and $\sigma_{\tilde{X}}$ is the scaling of the data in the feature space. If estimates for σ_y and $\sigma_{\tilde{X}}$ are available, then choosing C according to (26) is a scale invariant and hence possibly better choice than a fixed value. An estimate for σ_y can be easily obtained from the y-part of the training data or the derived δ_i. On the other hand, $\sigma_{\tilde{X}}$ can be estimated by using the kernel matrix (17).

The baseline penalty C_0 can again be chosen to be a large value. Alternatively, decreasing C_0 may be reasonable: This facilitates violation of the ranking and in turn yields a larger margin $\|w\|$. If the application scope of the rank learner is known (not necessarily meta-model based optimization), C_0 may be determined by a experimental study. Within the present work, this has been tried in order to achieve best fit in the linear regression sense in case of learning a linear function subject to various types of noise. The resulting best C_0 under this assumption is approximately $C_0 \approx 1$.

2.9 Use of models in meta-model based optimization

In this work, the main focus is on using meta-models as global models for the currently defined search space $\mathcal{X}$ (which is possibly subsequently refined and a new meta-model is trained, see Section 1.1). Accordingly, the largest part of the following experiments uses the meta-models globally. For increasing dimension d however, global meta-models may become less trustworthy. An alternative is to use local meta-models, i.e. train the meta-models on local data and using them locally only. In particular by working directly in the data space $\mathcal{X}$, that is using a linear kernel, a large margin rank learner yields an approximation to the steepest ascent direction (gradient) of the function to be learned. Among the experimental studies, only the real application study in Section 3.5 uses the models in a more local way. By choosing an appropriate set $\mathcal{S}^{\text{worst}}$ which may also depend on the location of the points in $\mathcal{X}$ (this possibility is not further discussed in the report), the rank learners discussed here offer good ways to tune their degree of locality.

There is another, fundamentally different way to use meta-models in optimization, which shall be briefly highlighted in this paragraph. In this work as in large parts of the literature ([3], [6], [7], [9] and others), meta-models are trained and directly evaluated in order to predict the quality of new solution candidates. In contrast, [4], [5], [11] and others propose to base the optimization on some derived quantity based on a trained meta-model, e.g. the expected improvement over best function value found so far [5]. To the author's knowledge, no such derived quantity has been suggested so far for ordinal regression meta-models. This could be an interesting topic for future research.

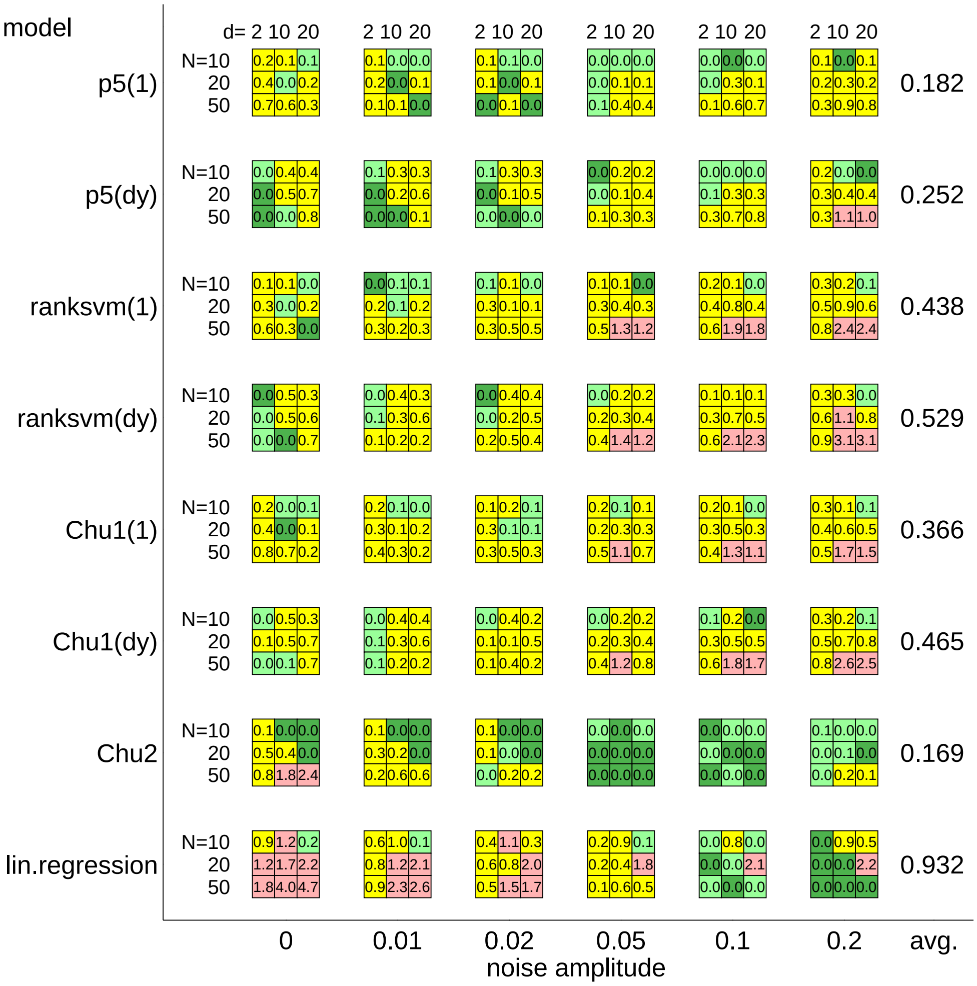
3 Experimental studies

In the following, various tasks of meta-model based optimization are used to compare the performance of different large margin rank learners and standard regression models. The manual optimization procedure (one single iteration as described in Section 1.1) is always used to optimize a function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^d$ is bounded (usually box-constraint). The dimension d ranges between 2 and 20. In practice, meta-model based optimization is often applied for dimensions $d \geq 3$ (where simple gridding and plotting meets its limits) and up to $d \leq 20$. With larger d, meta-model based optimization is typically not very useful in practice except for especially benign cases. All large margin learners are trained with slack variable penalty $C = 1000$. Statistical significance of the observed differences is tested with a 2-samples t-test with confidence level of 95%.

3.1 Learning a linear ordering

This task is a basic benchmark for linear ordinal regression. A vector $v \in \mathbb{R}^d$ is pre-chosen (we use simply the first unit vector). A set of training data $(x_i, y_i)_{i=1}^N$ where $x_i \in [0,1]^d$ and $y_i \in [0,1]$ is uniformly randomly generated such that both sequences (y_i) and $(\langle x_i, v\rangle)$ are monotonically increasing. The labels y_i may be additionally corrupted by noise. Then, the learners are trained on the training set, and the quality of the learners is evaluated by $\frac{|\langle w,v\rangle|}{\sqrt{\|w\|^2\|v\|^2}}$, where w is the corresponding weight vector from the primal problem formulation. Fig. 2 shows the average performance over 500 independent training sets of a number of large margin rank learners as well as linear regression. The labels y_i are corrupted by uniformly distributed noise of different amplitudes, and the experiments are done in different dimension d and with differently many training points N. For each large margin learner, the slack variable penalty C was individually optimized. The learners compared are the following:

- p5 models with uniform $\delta_i = 1$ and individual δ_i as in (5),
- rank SVMs with uniform $\delta_i = 1$ and individual δ_i as in (5),
- the explicit approach from [17] with uniform $\delta_i = 1$ and individual δ_i as in (5) and
- the implicit approach from [17], with quadratically many slack variables and constraints.



model		0			0.01			0.02			0.05			0.1			0.2			avg.
	d=	2	10	20	2	10	20	2	10	20	2	10	20	2	10	20	2	10	20	
p5(1)	N=10	0.2	0.1	0.1	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	
	20	0.4	0.0	0.2	0.2	0.0	0.1	0.1	0.0	0.1	0.0	0.1	0.1	0.0	0.3	0.1	0.2	0.3	0.2	0.182
	50	0.7	0.6	0.3	0.1	0.1	0.0	0.0	0.1	0.0	0.1	0.4	0.4	0.1	0.6	0.7	0.3	0.9	0.8	
p5(dy)	N=10	0.0	0.4	0.4	0.1	0.3	0.3	0.1	0.3	0.3	0.0	0.2	0.2	0.0	0.0	0.0	0.2	0.0	0.0	
	20	0.0	0.5	0.7	0.0	0.2	0.6	0.0	0.1	0.5	0.0	0.1	0.4	0.1	0.3	0.3	0.3	0.4	0.4	0.252
	50	0.0	0.0	0.8	0.0	0.0	0.1	0.0	0.0	0.0	0.1	0.3	0.3	0.3	0.7	0.8	0.3	1.1	1.0	
ranksvm(1)	N=10	0.1	0.1	0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.0	0.2	0.1	0.0	0.3	0.2	0.1	
	20	0.3	0.0	0.2	0.2	0.1	0.2	0.3	0.1	0.1	0.3	0.4	0.3	0.4	0.8	0.4	0.5	0.9	0.6	0.438
	50	0.6	0.3	0.0	0.3	0.2	0.3	0.3	0.5	0.5	0.5	1.3	1.2	0.6	1.9	1.8	0.8	2.4	2.4	
ranksvm(dy)	N=10	0.0	0.5	0.3	0.0	0.4	0.3	0.0	0.4	0.4	0.0	0.2	0.2	0.1	0.1	0.1	0.3	0.3	0.0	
	20	0.0	0.5	0.6	0.1	0.3	0.6	0.0	0.2	0.5	0.2	0.3	0.4	0.3	0.7	0.5	0.6	1.1	0.8	0.529
	50	0.0	0.0	0.7	0.1	0.2	0.2	0.2	0.5	0.4	0.4	1.4	1.2	0.6	2.1	2.3	0.9	3.1	3.1	
Chu1(1)	N=10	0.2	0.0	0.1	0.2	0.1	0.0	0.1	0.2	0.1	0.2	0.1	0.1	0.2	0.1	0.0	0.3	0.1	0.1	
	20	0.4	0.0	0.1	0.3	0.1	0.2	0.3	0.1	0.1	0.2	0.3	0.3	0.3	0.5	0.3	0.4	0.6	0.5	0.366
	50	0.8	0.7	0.2	0.4	0.3	0.2	0.3	0.5	0.3	0.5	1.1	0.7	0.4	1.3	1.1	0.5	1.7	1.5	
Chu1(dy)	N=10	0.0	0.5	0.3	0.0	0.4	0.4	0.0	0.4	0.2	0.0	0.2	0.2	0.1	0.2	0.0	0.3	0.2	0.1	
	20	0.1	0.5	0.7	0.1	0.3	0.6	0.1	0.1	0.5	0.2	0.3	0.4	0.3	0.5	0.5	0.5	0.7	0.8	0.465
	50	0.0	0.1	0.7	0.1	0.2	0.2	0.1	0.4	0.2	0.4	1.2	0.8	0.6	1.8	1.7	0.8	2.6	2.5	
Chu2	N=10	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	
	20	0.5	0.4	0.0	0.3	0.2	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.169
	50	0.8	1.8	2.4	0.2	0.6	0.6	0.0	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.1	
lin.regression	N=10	0.9	1.2	0.2	0.6	1.0	0.1	0.4	1.1	0.3	0.2	0.9	0.1	0.0	0.8	0.0	0.0	0.9	0.5	
	20	1.2	1.7	2.2	0.8	1.2	2.1	0.6	0.8	2.0	0.2	0.4	1.8	0.0	0.0	2.1	0.0	0.0	2.2	0.932
	50	1.8	4.0	4.7	0.9	2.3	2.6	0.5	1.5	1.7	0.1	0.6	0.5	0.0	0.0	0.0	0.0	0.0	0.0	

Figure 2: Comparison of a large number of models on the task of learning a linear ordering. The numbers show the regret w.r.t. the best performing learner on each task, normalized by the standard deviations. The color is dark green for the best performing learner on each task, light green if the performance is statistically not significantly worse (t-test, 0.95% confidence level), yellow if the performance is within one standard deviation from best, and a red for worse performance.

On this task and for mid range noise, the implicit approach with constraints and slack variables on each pair of data points performs better than all other models. Out of the large margin learners with linearly scaling QP, the p5 model performs best. Choosing the margin to be uniformly 1 performs better with high noise levels, while individual margins (5) work sightly better on low noise levels.

The p5 model rarely performs much worse than the implicit approach, but it scales better in the problem size. Hence, in the following, the focus is restricted to large margin learners

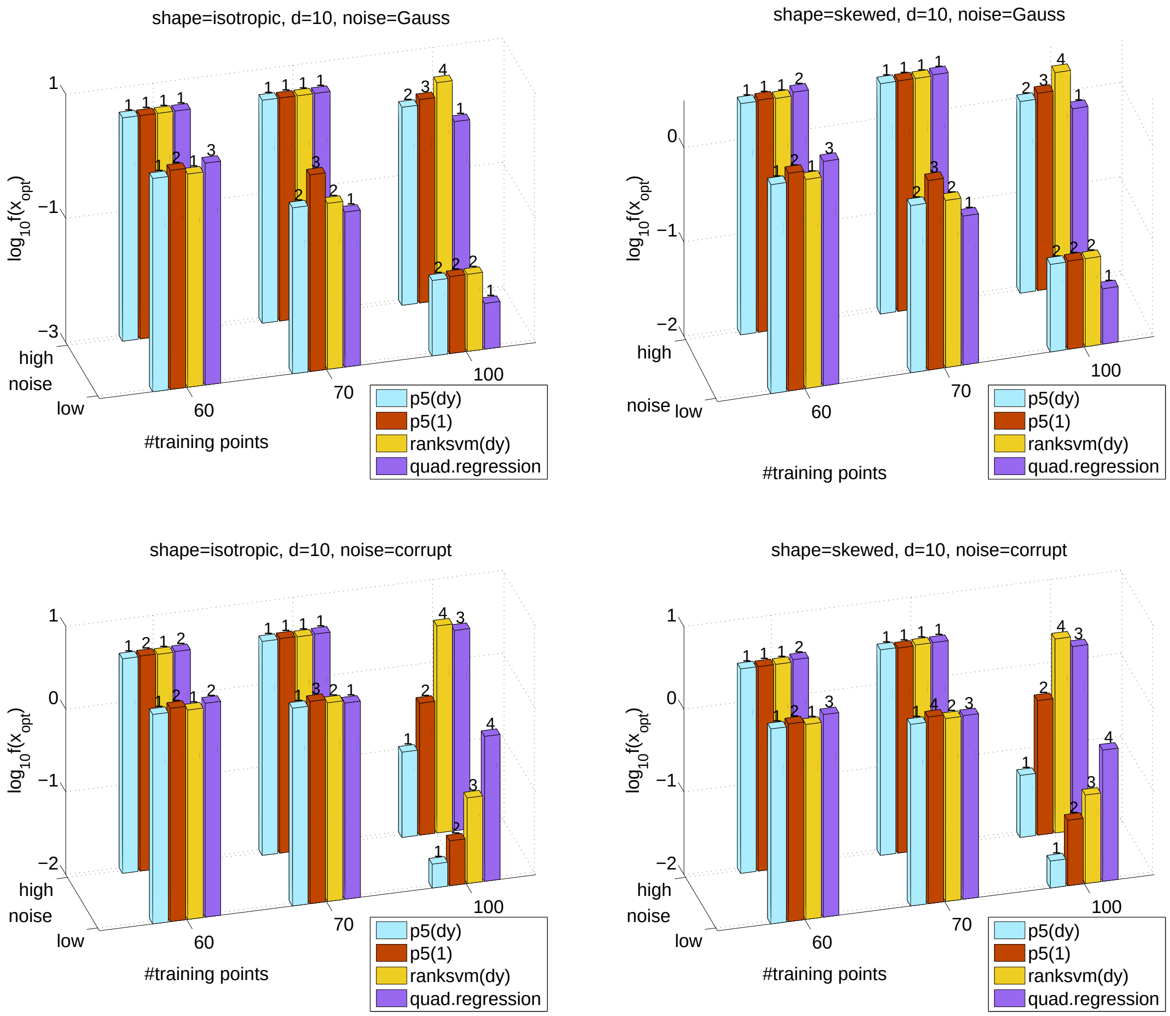


Figure 3: Performance comparison on the sphere in dimension $d = 10$: Simulations with Gaussian noise (top) vs. a fraction of the training data corrupted (bottom), isotropic (left) vs. skewed sphere (right). The numbers above the bars display the rank confirmed by a t-test. Average function values of 10^0 and above indicate poor performance of the corresponding meta-model in this case.

that scale linearly in the problem size.

3.2 Sphere function

The sphere function $f(x) = (x - x_0)^T Q(x - x_0)$ for $x \in \mathbb{R}^d$ and positive definite matrix $Q \in \mathbb{R}^{d \times d}$ is one of the most frequently used test functions for optimization. In the following experiments, an isotropic sphere with $Q = I$ and a skewed sphere with one eigenvalue of Q being 0.1 and the other eigenvalues 1 are used. The eigenvector corresponding to the

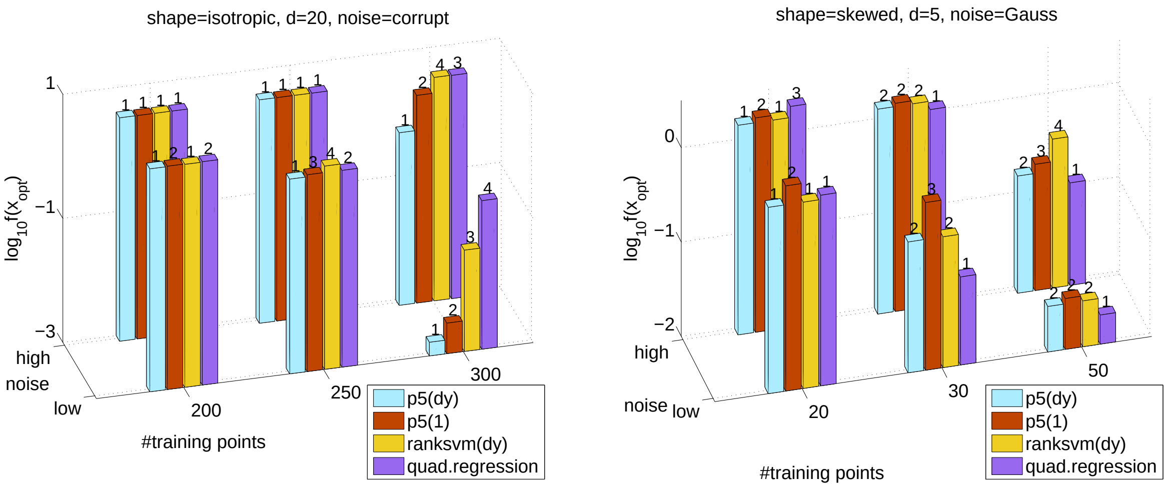


Figure 4: Simulations for the isotropic sphere in $d = 20$ with corrupting noise (left) and the skewed sphere in $d = 5$ with Gaussian noise (right). The numbers above the bars indicate the rank confirmed by a t-test.

small eigenvalue is uniformly randomly rotated, and the location of the optimum x_0 is standard Gaussian distributed. The training sets $(x_i, y_i)_{i=1}^{N}$ are also independently sampled from a standard Gaussian distribution. In one set of the experiments, y_i is affected by Gaussian noise of amplitude 1% (low noise) and 5% (high noise). In a different set of experiments, a single data point (low noise) and a fraction of 5% of the data (high noise) is completely corrupted (see Section 2.1). Each individual experiment is repeated 500 times with independently sampled randomness.

Quadratic kernels are used for the large margin learners. The meta-models are constructed from training sets of different size N. Fig. 3 shows the results in dimension $d = 10$. There, the full quadratic model has 66 free parameters. Hence, with the smallest training set of 60 points, the model is under-determined, while with the larger training sets, the model is over-determined. In order to deliver valid results also in the under-determined case, the quadratic regression is equipped with a Gaussian prior with mean 0 and standard deviation σ^2 satisfying $\frac{1}{2\sigma^2} = 10^{-3}$, corresponding to a quadratic penalty with weight 10^{-3} on the model coefficients.

Each model $\boldsymbol{m}(\cdot)$ is evaluated by locating its predicted optimum:

$$\min_{x} \boldsymbol{m}(x) \text{ s.t. } \|x - x_0\| \leq 2. \tag{27}$$

The constraint is introduced to prevent outliers. Because of the constraint, average function values of 10^0 and above indicate poor performance.

Evaluation criterion is the average true function value at the models' optima. The comparison in Fig. 3 shows that for Gaussian noise, not surprisingly the quadratic regression

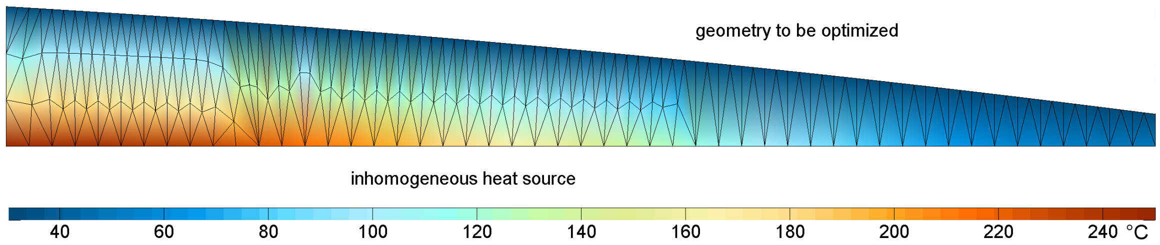


Figure 5: Illustration of the thermal insulator design task. The shape of the upper boundary is being optimized.

works best. In contrast, for a completely corrupted part of the training data, the p5 model with margins chosen individually according to (5) works best. This p5 model variant is consistently equally good or better than both the rank SVM (individual margins) and the p5 model with uniform margin. The numbers above the bars in the plot indicate the rank confirmed by a t-test: If the difference of two values is not statistically significant, the corresponding bars are labeled with the same rank.

Repeating the simulations in dimension $d = 5$ and $d = 20$ confirms these results. In $d = 5$, the high Gaussian noise case is approximated easier, and relatively less training data already gives good results (shown exemplarily for the skewed sphere function Fig. 4 right). In $d = 20$, the necessary amount of training data increases (shown exemplarily for the isotropic sphere function with corrupting noise Fig. 4 right). While there are 231 parameters in the full quadratic model, even a training set of size 250 barely suffices to learn the low noise case.

In the remainder of the report, all large margin learners will be trained with individual margins.

3.3 Thermal insulator design in 2D

The next study is a still a toy problem, but a more realistic one. The task is the design of a thermal insulator, illustrated in Fig. 5. A two-dimensional shape is designed, where the boundaries at the bottom and the sides are given as straight lines and the top boundary is optimized. A non-uniform known heat source is attached to the bottom boundary. The objective is to achieve best thermal insulation properties with a given amount of insulating material (fixed surface area of the shape). A practical example for this task is the design of an insulator for a thermally stratified tank.

Function evaluations are based Finite Element (FE) simulations using the FAESOR toolbox for Matlab [20]. A low complexity model is chosen, hence function evaluation are very fast (a few seconds on a desktop computer). In this section, results for a search space parameterized with 2 dimensions are presented. In the next section, a 5 dimensional parametrization is considered.

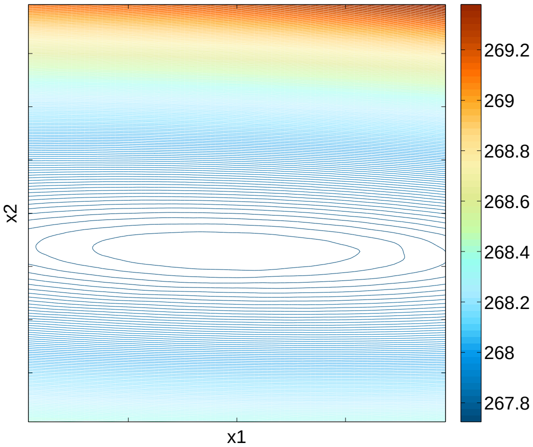


Figure 6: Contours of $f = f_{\text{FE}} \circ f_{\text{pre}}$ on the small parameter space $\mathcal{X}_{\text{small}}$.

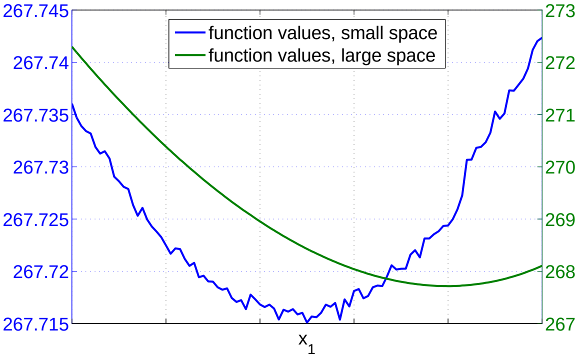


Figure 7: Noise of f when one parameter is modified, large vs. small space.

Here, a quadratic function is used to parameterize the upper boundary of the shape. This has 3 free parameters, one of which is determined by the given total amount of insulating material. This is done within a preprocessing function $f_{\text{pre}} : \mathcal{X} \to \mathcal{X}_{\text{FE}}$, where $\mathcal{X}$ is a box constraint subset of $\mathbb{R}^2$ and $\mathcal{X}_{\text{FE}}$ is the input space for the FE simulation. The FE evaluation is computed by a function $f_{\text{FE}} : \mathcal{X}_{\text{FE}} \to \mathbb{R}^+$ which evaluated the heat losses for a given shape. The function to optimize is hence

$$f = f_{\text{FE}} \circ f_{\text{pre}} : \mathcal{X} \to \mathbb{R}^+. \tag{28}$$

The problem is studied on two different parameter spaces: An initial large parameter space $\mathcal{X}_{\text{large}} \subset \mathbb{R}^2$ and a refined small space $\mathcal{X}_{\text{small}} \subset \mathbb{R}^2$ in which the optimum is located. The contours of a smoothed version of f in $\mathcal{X}_{\text{small}} \subset \mathbb{R}^2$ are shown in Fig. 6. The function has only one minimum (this is true also for the big parameter space) and resembles a skewed sphere. When varying one single parameter over its whole range, the noise characteristics can be explored (Fig. 7): The scaling is such that on the large space, the noise is not even visible, in contrast to the small space. Another piece of insight into this toy problem is given in Fig. 11 which shows the histograms of random function evaluations in the different parameter spaces.

Now the learning performance of different models is compared for both $\mathcal{X}_{\text{large}}$ and $\mathcal{X}_{\text{small}}$. The models tested are p5 models and rank SVMs with individual margins and inhomogeneous polynomial kernels of order 2 to 4, and quadratic regression. Each experiment is repeated 500 times, with a 30 element training set constructed from an independently random latin hypercube sample. For each trained model, the location of the optimum is computed. If the optimum is located within the small space (not on the border), it is counted as a success, otherwise a failure. The plots in Fig. 8 show the failure rates, average smoothed function values at the model optima of the successful runs, and common and order cross validation scores (see Section 2.7). The following observations are made:

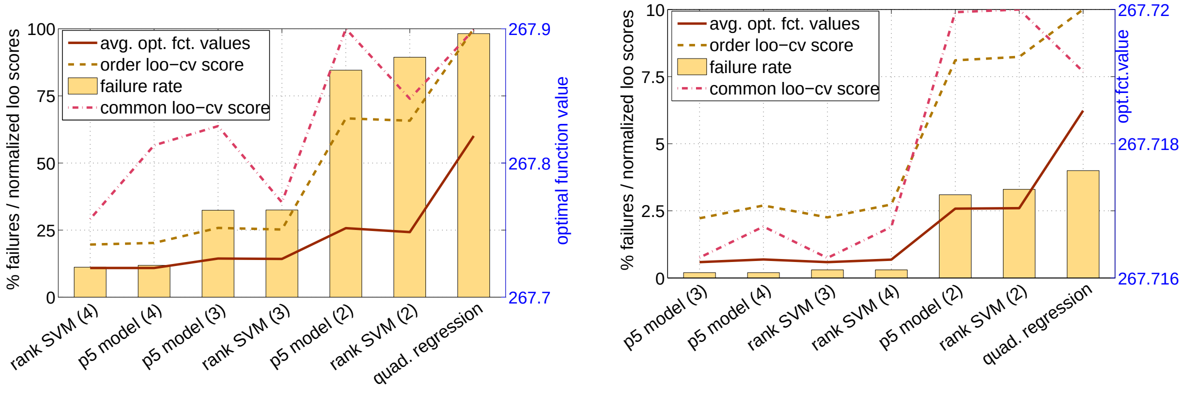


Figure 8: Average performance of different models on the large space (left) and small space (right). Orders of the kernels are indicated in brackets. In both plots, the left-most two average function values are significantly lower than the rest according to a t-test, while the difference between them is not statistically significant.

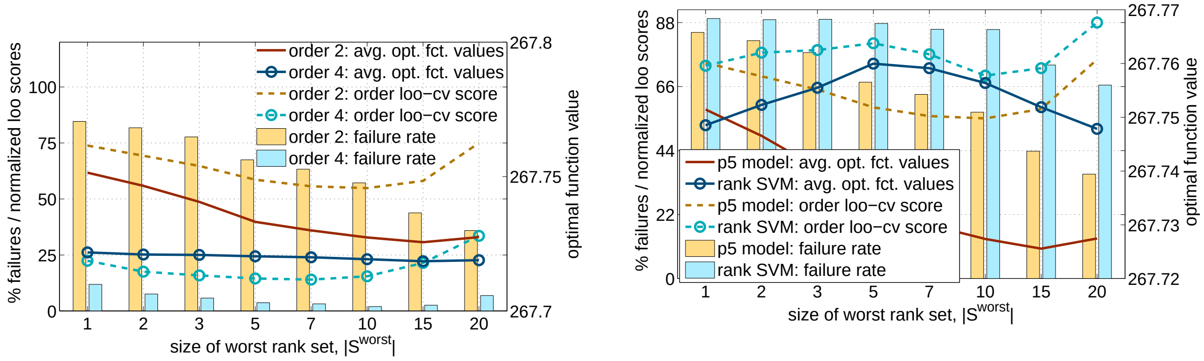


Figure 9: Large space, varying the size of $\mathcal{S}^{\text{worst}}$. Left a comparison of the quadratic and 4th order kernel p5 model, right a comparison of quadratic p5 model and rank SVM. All differences in average function value between the compared models are statistically significant under a t-test except for $|\mathcal{S}^{\text{worst}}| = 1$ and $|\mathcal{S}^{\text{worst}}| = 2$ in the right plot.

- The p5 models perform always at least as well as the rank SVM learners, sometimes slightly better.
- On the large space, a 4th order kernel is advantageous. (The same holds for regression: 3rd and 4th order polynomial regression do deliver better results than quadratic regression here.)
- Success rates are well correlated with the average function values on successful runs and with average order cross validation scores, but less well with common cross validation scores.

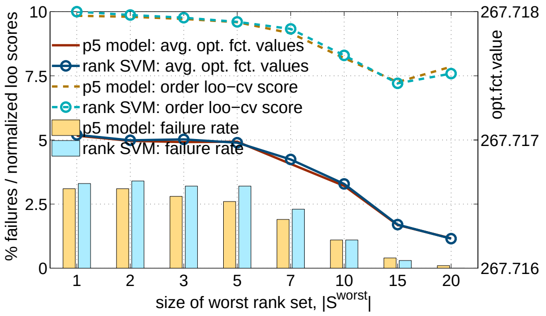


Figure 10: Small space, varying the size of $\mathcal{S}^{\text{worst}}$: comparison of quadratic kernel p5 model and rank SVM. Performance is in terms of average function value obtained is almost identical, and the difference is statistically insignificant.

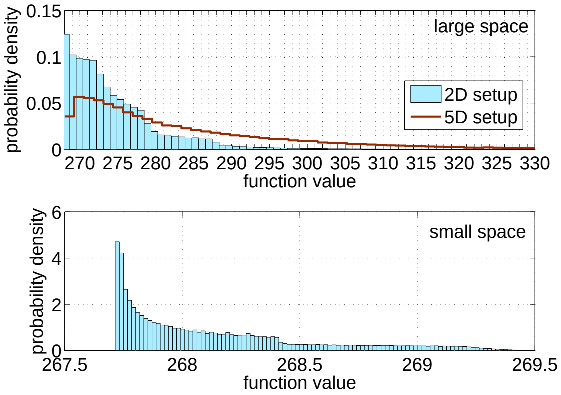


Figure 11: Histograms of random function evaluations for the 2 dimensional parametrization in both spaces $\mathcal{X}_{\text{large}}$ and $\mathcal{X}_{\text{small}}$ and for the 5 dimensional parametrization.

The next set of experiments explores the opportunities of introducing a non-trivial worst case rank set $\mathcal{S}^{\text{worst}}$ (see Section 2.3). Results for the large parameter space are shown in Fig.9, results for the small parameter space are shown in Fig.10. Here, the observations are the following:

- Correlation between failure rates, average function values at the model optimum and order cross validation scores is again good.
- Again the rank SVM never outperforms the p5 model, but sometimes clearly underperforms.
- In the cases analyzed here, a non-trivial set $\mathcal{S}^{\text{worst}}$ indeed improves the performance of large margin rank learners.

3.4 Thermal insulator design in 5D

A 5 dimensional parametrization of the thermal insulator design task is evaluated in this section. Instead of a quadratic curve for the upper shape boundary (Fig. 5), a cubic Bezier curve is used. Cubic Bezier curves are specified by 4 supporting points in 2D, hence 8 parameters. The abscissa of the end points are fixed, and one parameter is again inferred from the fixed amount of insulating material. Hence, the optimization problem is defined on a box constraint set $\mathcal{X} \subset \mathbb{R}^5$. There is no refinement of the parameter space considered here.

The function to be optimized is again defined as in (28). Comparing the histograms of random function evaluations (Fig. 11) indicates that this parametrization is less favorable

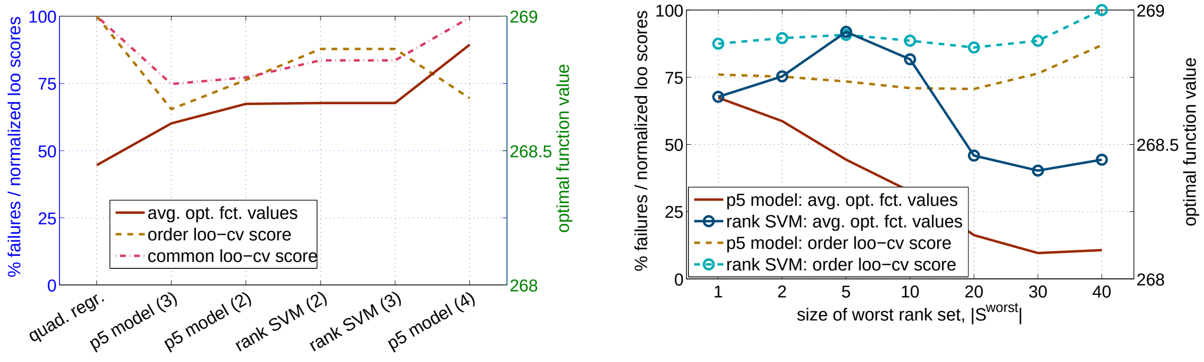


Figure 12: Results for the insulator design task on the 5 dimensional parameter space. Left: comparison of different models (kernel orders specified in brackets). Superiority of the quadratic regression is significantly under a t-test. Right: comparison of quadratic kernel p5 model and rank SVM (individual margins) under varying size of $\mathcal{S}^{\text{worst}}$. The differences are statistically significant for $|\mathcal{S}^{\text{worst}}| \geq 5$, and for $|\mathcal{S}^{\text{worst}}| \geq 10$, the performance is significantly better than the quadratic model on the left plot.

than the 2 dimensional one discussed before. Indeed, the function value achieved by the optimization here is never below 268 on average (Fig. 12).

The models tested are p5 models and rank SVMs with different kernels, and quadratic regression. Each experiment is repeated 500 times, with a 75 element training set constructed from an independently random latin hypercube sample. For each trained model, the location of the optimum is computed. In contrast to before, the model optima are evaluated just by their average function value, there is no target set. The results are presented in Fig. 12. The following observations are made:

- Again, the p5 model performs always equally well or better than the rank SVM.
- Cross validation scores are less clearly correlated to the performance, order cross validation does not exhibit a better correlation than common cross validation.
- A non-trivial set $\mathcal{S}^{\text{worst}}$ indeed improves the performance of the p5 models, but not necessarily that of the rank SVM.

3.5 Turbocharger gas inlet casing design

The last case study is on a a radial turbocharger gas inlet casing design. The results presented here are a small part of a larger multi-objective design optimization study at ABB Turbo Systems [21], based on Computational Fluid Dynamical (CFD) simulation. Because of the slightly different focus on multiple objectives, only parts of the results are meaningful within the present context. In that study, quadratic regression and some variants of the p5 model have been evaluated. Experiments with rank SVM and with

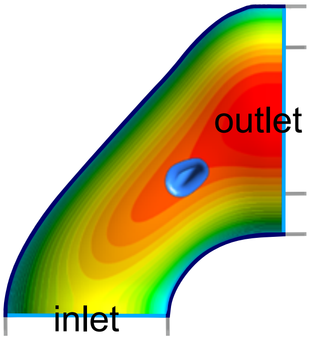


Figure 13: Illustration of the gas inlet casing design problem. Here, a horizontal cross section of the gas inlet casing is shown. The shape of the dark blue, curved boundaries is to be optimized. The color codes the normal extension of the boundary (3rd dimension).

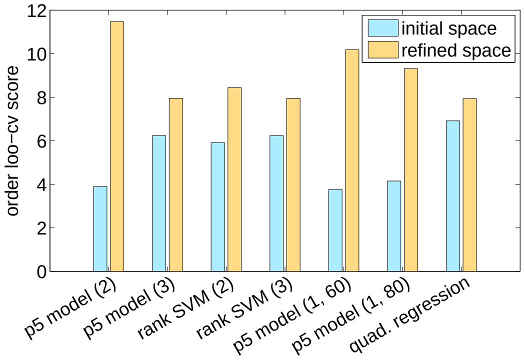


Figure 14: Order cross validation scores for different models in the initial and the refined parameter space. Kernel orders and $|\mathcal{S}^{\text{worst}}|$ (if applicable) are specified in brackets.

different size of $\mathcal{S}^{\text{worst}}$ for nonlinear kernels are missing. The meta-models are used in a more local way (compare Section 2.9), by optimizing them under a locality constraint (29).

The task is illustrated in Fig. 13. The shapes of the boundaries (dark blue) of the gas inlet casing are parameterized with 11 parameters. Two box constraint parameter spaces are considered: The initial one and a refined one, the latter is defined after evaluating the initial optimization. For training and optimizing meta-models, both spaces are normalized to the unit box, i.e. the data space is $\mathcal{X} = [0,1]^{11}$ in both cases.

Out of the multiple objectives considered in the full case study, the focus of the presentation here is minimization of the total pressure loss. Scalings, units, etc. of the real turbocharger cannot be precisely specified here. Evaluating the objective function via a CFD simulation takes about 1 hour of computation on a modern multi-core hardware (using ANSYS CFX).

The training set in the initial parameter space $\mathcal{X}_{\text{initial}} \subset \mathbb{R}^{11}$ contains 118 points, the training set in $\mathcal{X}_{\text{refined}} \subset \mathbb{R}^{11}$ contains 114 points, both constructed with a latin hypercube sampling. Evaluations are done differently than before: After training the candidate models, the model optima are computed, starting from the best solution within the training set x_{start}, and under the additional constraint

$$\|x - x_{\text{start}}\| \leq r, \tag{29}$$

that is, the model optimum may move away from the starting point by at most r, in terms of the Euclidean distance in the normalized parameter space. The maximum distance r is varied within $0 \ldots 0.5$. This evaluation method highlights more a local than global quality, namely the models' capabilities of proposing good search directions starting from the best known solution so far.

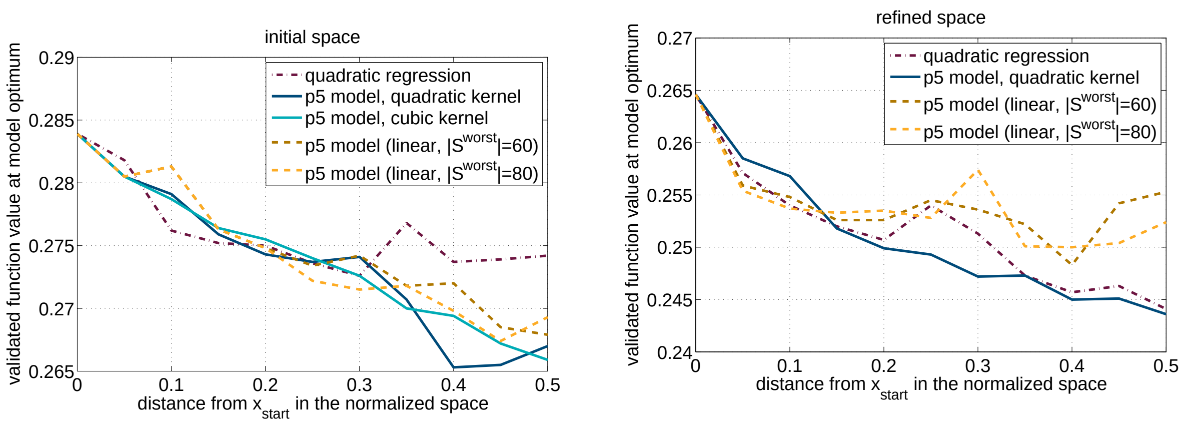


Figure 15: Comparison of model optima for the gas inlet design task on the initial parameter space (left) and the refined parameter space (right). The models are optimized under the additional bound on the difference between the solution x and the starting point x_{start}.

Fig. 15 compares the results for different models. Since each line corresponds to just a single run, conclusions have to be drawn with care. For the big parameter space (left plot), perhaps the quadratic regression performs worse than the other models. For the refined parameter space (right), the linear kernels seem to perform worse than the other models. Fig. 14 finally compares the order cross validation scores (rank SVM just added for the reader's reference, it has not been used in the study). Correlation between the cross validation scores and the models' performance is unclear in this case.

The best design resulting from this optimization improves the efficiency of the turbocharger by up to 1% in partial load compared to the initial design.

4 Discussion and conclusions

Ordinal regression offers opportunities for meta-model based optimization. At the very least, ranking learners enrich the available tools: They extract different information from given training data than common regression, and therefore will perform better for certain applications and worse for others. They offer different tuning opportunities (kernel, slack variables penalty, worst rank set). It is tempting to argue that ranking learners *in principle* are more suitable for optimization than common regression, since the property of optimality does not depend on absolute function values, but only on ranks. However, this argument must be taken with care, see the discussion in Section 2.1. On the other hand, over the range of experimental studies presented in this work, ranking learners often compare favorably to common regression.

A particular difference between large margin learners and common regression arises limited amount of training data in high dimensions or with high order models. An under-determined regression needs some regularization, e.g. a Gaussian prior. A large margin learner implicitly regularizes by the maximum margin principle and hence can be applied in arbitrarily high dimensional feature spaces without particular care. Better understanding the differences and implications for various model / kernel complexities and sample sizes and deriving practical bounds on sample complexity, is a topic of further research with high relevance for applications.

The p5 model introduced here performs well for meta-model based optimization, in particular it always compares favorably to the rank SVM. But its application is not restricted to this. It can be used for any order regression problem. By construction, it avoids some of the problems of other large margin rank learners.

References

[1] J. Nocedal and S. J. Wright, *Numerical optimization.* New York: Springer, 2006.

[2] G. E. P. Box and K. B. Wilson, “On the experimental attainment of optimum conditions,” *J . Roy. Statist. Soc. Ser. B*, vol. 13, pp. 1–45, 1951.

[3] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, “Accelerating evolutionary algorithms with gaussian process fitness function models,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 35, p. 183–194, 2005.

[4] P. Hennig and C. J. Schuler, “Entropy search for information-efficient global optimization,” *arXiv:1112.1217 [cs, stat]*, Dec. 2011. [Online]. Available: http://arxiv.org/abs/1112.1217

[5] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998.

[6] S. Kern, N. Hansen, and P. Koumoutsakos, “Local meta-models for optimization using evolution strategies,” in *Parallel Problem Solving from Nature - PPSN IX*, ser. Lecture Notes in Computer Science, T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds. Springer Berlin Heidelberg, Jan. 2006, no. 4193, pp. 939–948.

[7] I. Loshchilov, M. Schoenauer, and M. Sebag, “Comparison-based optimizers need comparison-based surrogates,” in *Parallel Problem Solving from Nature, PPSN XI*, ser. Lecture Notes in Computer Science, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds. Springer Berlin Heidelberg, Jan. 2010, no. 6238, pp. 364–373.

[8] I. Loshchilov, M. Schoenauer, and M. Sèbag, “Bi-population CMA-ES agorithms with surrogate models and line searches,” in *Proceedings of the 15th Annual*

Conference Companion on Genetic and Evolutionary Computation, ser. GECCO '13 Companion. New York, NY, USA: ACM, 2013, p. 1177–1184. [Online]. Available: http://doi.acm.org/10.1145/2464576.2482696

[9] T. P. Runarsson, "Ordinal regression in evolutionary computation," in *Parallel Problem Solving from Nature - PPSN IX*, ser. Lecture Notes in Computer Science, T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds. Springer Berlin Heidelberg, Jan. 2006, no. 4193, pp. 1048–1057.

[10] A. Sóbester, S. J. Leary, and A. J. Keane, "On the design of optimization strategies based on global response surface approximation models," *Journal of Global Optimization*, vol. 33, no. 1, pp. 31–59, Sep. 2005.

[11] J. Villemonteix, E. Vazquez, and E. Walter, "An informational approach to the global optimization of expensive-to-evaluate functions," *Journal of Global Optimization*, vol. 44, no. 4, pp. 509–534, Aug. 2009.

[12] N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger, "Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems," *Appl. Soft Comput.*, vol. 11, no. 8, p. 5755–5769, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2011.03.001

[13] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*. Philadelphia: Society for Industrial and Applied Mathematics/Mathematical Programming Society, 2009.

[14] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge, UK; New York: Cambridge University Press, 2004.

[15] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, Smola, Bartlett, Schoelkopf, and Schuurmans, Eds. MIT Press, Cambridge, MA, 2000, p. 115–132.

[16] A. Shashua and A. Levin, "Ranking with large margin principle: Two approaches," in *In Proceedings of Advances in Neural Information Processing Systems*, 2002.

[17] W. Chu and S. S. Keerthi, "Support vector ordinal regression," *Neural Computation*, vol. 19, p. 2007, 2007.

[18] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, p. 377–384. [Online]. Available: http://doi.acm.org/10.1145/1102351.1102399

[19] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[20] P. Krysl, *A pragmatic introduction to the finite element method for thermal and stress analysis: with the matlab toolkit SOFEA*. Singapore; Hackensack, N.J.: World Scientific, 2006.

[21] K.-K. So, “EP80: Aerodynamische Optimierung und Auslegung des radialen und axialen Gaseintrittsgehäuses des Z200,” ABB Turbo Systems, Tech. Rep. CHTUS TB11003, May 2013.