The handle [http://hdl.handle.net/1887/22055](http://hdl.handle.net/1887/22055) holds various files of this Leiden University dissertation.

**Author:** Koch, Patrick  
**Title:** Efficient tuning in supervised machine learning  
**Issue Date:** 2013-10-29
Appendix A

Analysis of noisy multi-criteria optimization

In this chapter we analyze the influence of noisy evaluations in multi-criteria optimization. Here, we focus on the approximation of the Pareto set, when the real Pareto front is known and noise is artificially added to the solutions.

A.1 Estimating noise in multi-criteria machine learning experiments

In the multi-criteria tuning in Chapter 7 we observed, that very high classification accuracies were obtained in some runs. While this seems to be a good result at first, it is obvious, that these solutions can be biased due to some lucky evaluations. Possible reasons for this can be beneficial training and test set splittings, while the real generalization error of such solutions might be worse. This means, that the parameter settings need not necessarily perform equally well on other sub-samples of the data. Unfortunately this drawback can not be prevented without making restrictions elsewhere, e.g., repeated evaluations leading to inferior exploration. In fact each solution is uncertain, but is always regarded as a real observation, which is a necessary requirement for the expected improvement criterion. It is important to know, that the solutions will not be discarded by the optimization procedure, unless they are dominated by other solutions. During the optimization run using the re-interpolation procedure by Forrester et al. [82], a solution is evaluated only once on a holdout set, which is randomly sampled from a larger test set. A simple solution for avoiding biases would be to perform repeated evaluations in order to stabilize the tuning error. But as we showed earlier, repeated evaluations also lead to less exploration of the search space, since the budget is limited. As a result of Jin and Branke [129] \( N \) repeated evaluations reduce the noise by a factor of \( \sqrt{N} \), but at the same time increase the total budget by the factor \( N \). For this reason we do not recommend this as a real alternative.

A frequently used option to alleviate biased solutions is to consider alternative
performance measures, like cross-validation or leave-one-out CV. An overview about resampling strategies has been given in [20]. These resampling strategies yield a good estimation of the real generalization performance. Unfortunately most measures are computationally expensive and quickly become infeasible for large-scale data. For this reason we propose to estimate the generalization performance with SVM in the following section, followed by an overview about strategies for noise handling in multi-criteria optimization.

A.1.1 Related work
In this section we give a short overview about existing approaches measuring the generalization performance of SVMs and multi-criteria optimization in the presence of noise.

A.1.1.1 Estimating the generalization performance with SVM
There exists a large body of work for analyzing the generalization performance of SVM according to bias and parameter settings: Cherkassky and Ma [41] proposed an analytical method to select hyperparameters $C$ and $\epsilon$ without direct resampling, but they did not incorporate a systematic tuning of the kernel parameter $\gamma$. We think that for this reason the approach is very limited and rather suited for simpler learning tasks. The approach only aims at setting SVM specific parameters, but it can not be extended for additional learning parameters. In our experiments we always obtained a very high sensitivity with regard to parameter $\gamma$, and it was also shown that the performance of heuristical $\gamma$ values is not competitive with tuned $\gamma$ values. Joachims [131] recommends a bound for the leave-one-out error of SVM which is more efficient to calculate. This is promising, since the leave-one-out error is a good estimator for the unbiased generalization error. Although Joachims reports a very strong correlation with the leave-one-out error, the method was only tested on text classification problems. Therefore it remains unclear, if the method also works for other classification and regression tasks. Valentini and Dietterich [243] analyzed bias and variance of the error to gain insights into SVM learning. They discovered patterns of bias and variance related to different parameter settings and kernel functions of SVMs. However, they did not incorporate noisy data in their experimental study, which makes it difficult to transfer the effects to real situations.

A.1.1.2 Noisy multi-criteria optimization
Some research has been undertaken for coping with noise directly inside the optimization procedure. Comprehensive overviews about noise in multi-criteria optimization have been given by Goh and Tan [96] and Eskandari and Geiger [71]. In more detail, the following approaches have been proposed to handle noise in multi-criteria optimization problems.

A theoretical foundation for noisy multi-criteria optimization has been advocated by Hughes [120]. Hughes discovered, that the probability for finding a non-dominated solution
can be analytically quantified, when the noise is normally distributed and the variance of
the noise is known. For two solutions $a$ and $b$ the probability that $b$ dominates $a$ can be
calculated as follows:

$$p(b \prec a) = \frac{1}{\sqrt{2\pi(s^2+1)}} \int_0^\infty e^{-\frac{(x-m)^2}{2(s^2+1)}} \, dx$$  \hspace{1cm} (A.1)

$$= \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{m}{\sqrt{2+2s^2}}\right)$$  \hspace{1cm} (A.2)

where $\operatorname{erf}$ is the error function (without loss of generality the probability density function
can be considered therefore). Variables $m$ and $s$ are defined as follows:

$$m = (\mu_a - \mu_b) / \sigma_b,$$

$$s = \sigma_a / \sigma_b$$

where $\mu_a$ and $\mu_b$ are the mean fitness and $\sigma_a$ and $\sigma_b$ the variance of solutions $a$ and $b$ respectively.

Hughes finally ranked the solutions according to the probability given by Eq. A.2.
Independently, Teich [238] proposed a similar concept termed probabilistic dominance. For
these works Fieldsend and Everson [75] advocated a Bayesian algorithm for learning the
noise variance.

Bui et al. [34] compared the approaches of Hughes and Teich using the popular NSGA-
II [57] algorithm with a strategy termed fitness inheritance. Here, the fitness of solutions
is assigned by the mean fitness of the parents, and if the fitness of the offspring does not
satisfy a certain confidence level, the fitness of the offspring is re-evaluated a number of
times. In an empirical study Buy et al. [34] discovered, that the probabilistic dominance
concept is inferior to the re-evaluation strategy. As its main weakness, the probabilistic
dominance concepts suffers from a deteriorated diversity of the solution set. As another
disadvantage, also high computational costs occur, caused by the evaluation of the integral
formula.

For this reason Syberfeldt et al. [237] advocate an adaptive re-evaluation approach where
the number of replicates are adapted according to the amount of noise. In an experimental
study they compare the probability dominance concept of Teich [238] and Hughes [120]
with their re-evaluation strategy. Similarly Bui et al. [34] they come to the conclusion, that
the probabilistic dominance concept performs worse compared with re-evaluation. However,
they also indicate that the re-evaluation of inferior solutions is unsatisfactory. As a solution
to this problem they propose multi-criteria OCBA [163] for defining the number of repeats.
Unfortunately multi-criteria OCBA was not incorporated in their experimental study, where a
default setting of 2 evaluations for each design point was stated. In contrast to our approach
their multi-criteria optimization algorithm is model-assisted by Artificial Neural Networks,
but does not allow for internal noise estimation like Gaussian processes or Kriging.
A.1.2 Model-assisted noise handling

While the previously mentioned approaches all aim at estimating the noise by re-evaluations or additional considerations inside the learning algorithm, it is more convenient, to estimate the noise by the surrogate model itself. Kriging surrogate models have a built-in mechanism for estimating uncertainties, that can be ideally used for the selection of new infill points. We investigate in this section, if this prediction is sufficient, that is wrong decisions of the optimization procedure are avoided.

In the previous experiments the RI by Forrester et al. [82] revealed the best performance. In RI at first a non-interpolating Kriging model is built, which assumes a variance greater than zero for all predictions. Note, that all required values are given by the surrogate model: $y_{real}(\vec{x})$ is the observed value of a parameter setting $\vec{x}$, while $\hat{\sigma}(\vec{x})$ is the corresponding variance.

It is possible that Kriging fails in correctly predicting the variances $\hat{\sigma}$. It is also possible that this failure is not revealed, because the real variance always remains unknown. Only in some cases wrong variances can be detected by the users of ML. E.g., an estimated variance of $\hat{\sigma} = 0.1$ for a gain value of $y_{real} = 0.98$ prediction accuracy is infeasible for one side (yielding $0.98 + 0.10 = 1.08$). It is obvious, that the prediction accuracy can never be larger than 1, which equals 100% correctly predicted patterns. And even this value would be at least questionable as generalization performance in ML. An alternative to avoid such irregular variances is to incorporate other variance estimators inside Kriging, e.g., by using conditional distributions, like a truncated normal distribution. We think that this is a matter of further research, and rather concentrate on analyzing the quality of the variance estimation.

A.1.2.1 Research questions

The research questions in this section are defined as follows:

Q1 How many solutions are selected in the final non-dominated set because of beneficial noise, that is noise which improves the solutions artificially?

Q2 How large is the distance in the solution space between the noisy solutions in the non-dominated set and the same solutions without noise?

Q3 When re-evaluating the non-dominated points on the objective function without noise, are the solutions Pareto-optimal, or can they be dominated by other solutions?

A.1.2.2 Experimental analysis

In an experimental study we analyze the influence of noisy evaluations for SMS-EGO. Up to now we experimented on tuning ML processes. If reasonable training samples are selected in ML, the observed error should fluctuate around the true generalization error.
A good estimator of the true generalization error is the leave-one-out error. However, it is not possible to define a *true* Pareto front for ML processes, because of the prevailing uncertainties caused by the resampling of data. For this reason we analyze the behaviour of SMS-EGO on the deterministic test function ZDT2, instead of taking into account real ML data. For ZDT2, the Pareto front can be analytically calculated, which will help us in our analysis.

**A.1.2.3 ZDT2 test function**

The two-criteria function ZDT2 is defined as follows [45]:

\[
\begin{align*}
    f_1 &= x_1 \quad \text{(A.3)} \\
    f_2 &= g(\bar{x}) \cdot [1.0 - (x_1 / g(\bar{x}))^2] \quad \text{(A.4)} \\
    g(\bar{x}) &= 1 + \frac{9}{n-1} (\sum_{i=2}^{n} x_i) \quad \text{(A.5)}
\end{align*}
\]

subject to:

\[
0 \leq x_i \leq 1, \ i = 1, ..., n \quad \text{(A.6)}
\]

The shape of the Pareto front is non-convex. The Pareto front itself can be analytically calculated by

\[
\begin{align*}
    x_1 &= x_1 \quad \text{(A.7)} \\
    x_2 &= 1 - x_1^2 \quad \text{(A.8)}
\end{align*}
\]

This enables us to compare the attainment surface of SMS-EGO and the true Pareto front of ZDT2.

To make the experiment more comparable to ML experiments, we added normally distributed noise to the function values:

\[
y = y + \sigma N[0, 1] \quad \text{(A.9)}
\]

For ZDT2 we set a value of \( \sigma = 0.1 \) for both objectives \( f_1 \) and \( f_2 \). This adaptation of the standard deviation was incorporated, because in ML comparable noise levels have been observed in preliminary runs.

We ran SMS-EGO on the ZDT2 function with normally distributed additive noise, and saved the parameter settings, as well as the objective function values obtained during the runs. For SMS-EGO we defined the following settings: We limited the number of function evaluations by defining a maximal budget of 150 function evaluations, where 20 evaluations were spent for the initial LHS (initial design). The input space \( X \) was set to three input dimensions. With these settings it should be possible to attain the Pareto front. However, our main interest is not the solution quality of SMS-EGO, but whether the algorithm can model the noise correctly and does not retain.
A.1.2.4 Results

The approximation sets of the noisy ZDT2 and the ZDT2 function without noise are shown in Fig. A.1. The plot shows that the Pareto front of ZDT2 without noise could be well approximated by SMS-EGO (gray points). This was intended, to expose the effects of the approximation of the algorithm for the noisy ZDT2 function. The red dots picture the non-dominated solutions of ten independent runs, when Gaussian additive noise was added for ZDT2. It can be seen from the plot, that almost all red dots lie beyond the real ZDT2 Pareto front (in the case without noise). This is now the same effect as observed in some ML tuning runs, where parameter settings have been evaluated and the evaluation of the prediction model was too optimistic.

![Figure A.1: Approximation of ZDT2. The plot shows the approximated points for the first and second objective on the x- and y-axis respectively. The gray points show the non-dominated solutions obtained after a non-dominated filtering for the solutions of ten runs on ZDT2 without noise. The budget for each run was 150 evaluations with an initial design size of 20. Instead the red points show the non-dominated solutions of ten runs on the noisy ZDT2. They seem to dominate the gray solution set, because (possibly negative) Gaussian noise was added to the solutions. The blue points then show the corrected solutions, when the noise is subtracted again.](image)

Summarizing all ten runs, we obtain the non-dominated solutions as shown in Fig. A.2. Therefore we applied a non-dominated filter for all solutions of the ten runs with SMS-EGO. Again, the points shown in red depict the (now aggregated) non-dominated solution set for
ZDT2 with additive noise, while the blue points depict the corresponding function values without this noise. It can be seen in the plot that the blue solutions are again distributed over the whole Pareto front. Also “corner” solutions, that are solutions when optimizing one single objective of the multi-criteria problem, have been obtained and remain non-dominated.

Figure A.2: Same plot for ZDT2 as in Fig. A.1 but now with a non-dominated filtering applied to the aggregated solutions of ten runs.

A.1.2.5 Conclusion

We can conclude that SMS-EGO can deal with misleading noise. Of course the algorithm selects noisy solutions, which appear to be better because of the additive noise term. The solutions with additive noise are improved by the preset standard deviation of the additive normal distribution \( N(0, 1) \), which was \( \sigma = 0.1 \) in our experiments. This is a direct answer to research question Q2. Surprisingly no wrong decisions could be observed, e.g., it did not occur, that too many solutions were selected which are not close to the Pareto front of the non-noisy function. Instead, the obtained solution set is well distributed over the Pareto front. We can state, that the main objectives of multi-criteria optimization are respected with additional noise. Our experiment shows, that although noisy solutions are taken in the solution set of SMS-EGO, the solutions are nevertheless close to the real Pareto front after substracting the noise. As a consequence the question deployed in Q1 can be answered
positively, stating that although the solutions are influenced by the additive noise term, most solutions are still Pareto optimal when the noise is neglected. For this reason we can also affirmate research question Q3 which can be seen as a good result for the noise handling inside model-assisted optimization using Kriging surrogate models.