Active Learning

for overlay prediction in semi-conductor manufacturing

by

K.A. van Garderen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday May 18, 2018 at 11:00 AM.

Student number:	4144384	
Project duration:	September 1, 2018 – May 18, 2018	
Thesis committee:	Dr. D.M.J. Tax,	TU Delft, Supervisor
	Prof. M.J.T. Reinders,	TU Delft
	Dr. A.E. Zaidman,	TU Delft
	Dr. A. Ypma,	ASML, Supervisor
	Dr. M. Larranaga,	ASML

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This thesis is the conclusion to my research internship at ASML and my time as a student at the TU Delft. The initial goal of this project was to design and implement an interactive visualization, with active sample selection and user interaction. A very broad goal which gave me many possibilities to choose my own contribution, and the challenge of deciding on more specific research questions. The path to this final thesis had a number of twists and turns, which made it difficult at times but also a very valuable learning experience. For this I want to express my gratitude to the people who stood by me for the duration of the project.

First, I would like to thank dr. Alexander Ypma for arranging this project and providing me with many different angles for research. Also, I would like to thank dr. David Tax for keeping me focused, providing me with regular feedback and always having a positive attitude. Also I would like to thank my colleagues at ASML for making my internship a very pleasant experience. Especially I would like to thank dr. Maialen Larranaga and dr. Feagheh Hasibi for the many productive discussions about data science, feminism and much more. Finally, I would like to thank my boyfriend for the ongoing support and for listening to my problems whenever I got stuck.

I wish you a pleasant read.

K.A. van Garderen
Eindhoven, May 7 2018

Contents

1 Introduction

The semi-conductor industry is driven by a demand for smaller devices and increasing yield in production. However, as the critical size of patterning on a chip decreases, the margin for error decreases as well. Inaccuracies in the pattern may lead to malfunctioning devices, which cause a decreasing yield of the fabrication facility ('fab'). Overlay between layers in the patterning is an important factor in production errors, and therefore maintaining a high yield depends on the ability to control and limit these errors. Typically a feedback loop is in place to correct any overlay errors as they appear in the process. To provide this feedback, additional costs are introduced to the system by time-consuming and costly overlay measurements.

There is increasing attention for data-driven solutions to reduce the amount of errors and streamline the process. Not only is the wafer scanner, produced by ASML, the driving force behind decreasing pattern sizes, it also offers unique possibilities for data analysis due to the multitude of sensors involved with precise lithography. Wafers going through multiple layers of processing will visit the scanner repeatedly and generate a wealth of data each time. Recent projects show that this data can offer great insight when analyzed using machine learning techniques [10] [11][15] [16].

This research is inspired by the demand to predict and understand overlay results through scanner data. Scanner data typically contains an abundance of measurements and this results in a high-dimensional problem. As high-dimensional problems typically require large amounts of data, many overlay measurements are required to train the regression parameters that describe this problem. These measurements are costly to obtain and therefore more effective models are needed to limit the amount of measurements required for an effective prediction. This thesis focuses on one specific way to achieve this, which is to make informed choices in the measurement strategy, also called 'active learning' [21].

The term active learning applies to a range of techniques where a sampling and labeling strategy is employed to improve the learning rate of a learning machine. Typically, there is a relatively large set of instances where the label is unknown, but acquiring these labels is possible at a cost. The active learning strategy can select one or more instances based on the current model, with the goal of achieving the largest possible gain in performance with a single measurement. An additional goal of this thesis is to use the predictive model for an informative visualization to an expert.

Active learning speaks to the imagination as it makes sense to choose costly measurements in an intelligent way, but doing so we take a risk of skewing the sample distribution in a way that is not representative of the true distribution. There can be real benefit in doing so, but the sampling bias can also be detrimental to performance. As sampling strategies are designed for a specific problem setting and learning model, and results may even differ between datasets, it is difficult to say whether these results generalize well.

In the next chapter the motivation and context of this thesis is refined in the form of a problem statement, and the research questions are defined. Chapter three introduces the context in terms of the regression and evaluation methods, which together form the framework for any active learning strategy. Chapter four gives an overview of active learning literature and discusses a number of general methods that are implemented and evaluated in chapter five. Finally, a comparison is made to a different regression model in chapter six, before drawing general conclusions and discussing opportunities for future research.

2

Problem Statement

This chapter describes the context and goals of this thesis. First, the source and shape of overlay errors is discussed in order to understand the regression targets. Next, the available data generated by the wafer scanner is described. The broader context of this research into active learning is discussed and finally the research goal and questions are defined.

2.1. Overlay

Overlay is a measure of the local displacement between two layers on a chip. It is impossible to completely avoid any displacement, but the ability to limit overlay is essential to produce semiconductor circuits on a nanometer scale. In a fab (fabrication facility), the overlay is measured on a regular basis during production and corrections are made in a continuous feedback loop. The actual policy for this overlay control is a company secret of the fab and therefore not known in this analysis.

Measurements for overlay are not performed in the scanner, but in a separate metrology tool. They are difficult and relatively time-consuming, and they induce an additional cost and potential delay into the production process. Therefore measurements are only performed for a fraction of the produced wafers.

In order to measure the overlay, a number of structures are printed with each layer, especially designed to enable an accurate measurement. The raw result of an overlay measurement is presented in a 'wafer map', which contains vectors (local displacement in 2D) for each location (see figure 2.1). To perform overlay control, the local overlay measurements must be translated to parameters that correspond to possible corrections in the process of alignment and exposure in the scanner. Alignment refers to the wafer positioning in the machine. During exposure the actual pattern, which is recorded on a mask called a 'reticle', is printed on each field of the wafer. If there are inaccuracies in the exposure, such as a rotated reticle, these will lead to an overlay pattern that is repeated for each field.

A global polynomial fit is performed that approximates the measurements through correctable param-

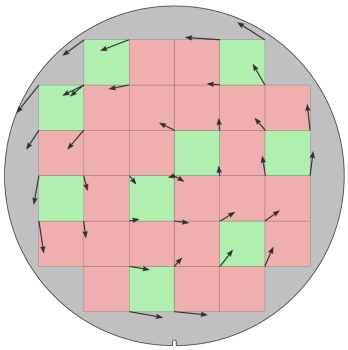

Figure 2.1: Wafer map showing sparsely measured local overlay errors. Image by Cepheiden (Own work) CC BY-SA 3.0 , via Wikimedia Commons

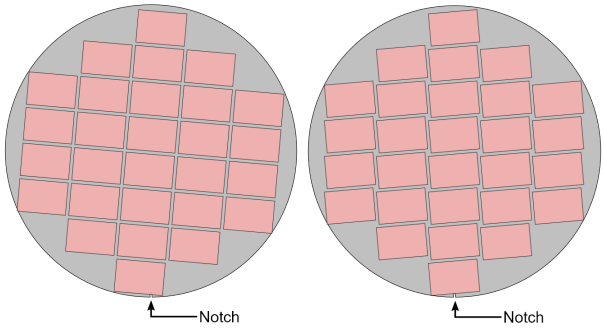

Figure 2.2: Illustration of error due to wafer rotation (left) and field rotation (right). A notch is used to indicate wafer orientation. Image by Cepheiden (Own work) CC BY-SA 3.0 , via Wikimedia Commons

eters such as linear displacement, rotation and magnification. These parameters contain global wafer parameters, which may be corrected during alignment, and field terms corresponding to corrections during exposure. The difference between the wafer terms and field terms is illustrated for rotation in Figure 2.2. The ten overlay parameters used in this project are:

1. Translation in X ('Tx')
2. Translation in Y ('Ty')
3. Wafer rotation in X ('WRotX')
4. Wafer rotation in Y ('WRotY')
5. Wafer expansion in X ('WExpX')
6. Wafer expansion in Y ('WExpY')
7. Field rotation in X ('FRotX')
8. Field rotation in Y ('FRotY')
9. Field magnification in X ('FMagX')
10. Field magnification in Y ('FMagY')

There is a desire in the semiconductor industry to partially replace metrology by prediction. If the amount of metrology could be reduced, it would decrease production time and therefore increase the yield in a fab. The input parameters for such a prediction can be found in the abundance of measurements performed during alignment and exposure in the lithography scanner.

2.2. Available data

To develop an overlay prediction model, we have access to a dataset containing overlay measurements for just over a thousand processed wafers over the course of three weeks. The wafers have been processed with several different layers, using four different scanners. Overlay measurements can be performed after each layer, and in total there are just over two thousand measurements. For each processing step, scanner data is recorded about alignment and exposure.

Data is generated during alignment of the wafer and during exposure. The same data is available for the bottom layer, and because overlay is measured with respect to the bottom layer this data is also used for prediction. Additionally, the corrections applied to the scanner due to the overlay feedback loop are also considered.

This particular dataset was generated with non-overlapping layers. The fact that layers are non-overlapping means that overlay of each layer is measured with respect to the same bottom layer. The overlay is therefore not expected to be influenced by the previous layer, but rather by the bottom layer.

Wafers are grouped into lots, which are batches of wafers that are entered into the machine together and may follow the same processing path. An illustration of the full process can be found in figure 2.3.

The wafers can be processed with different machines, using different reticles. Inside the machine, the wafers are processed one by one through alignment and exposure. To speed up this process, the machine contains two chucks. While one wafer is exposed on chuck one, the alignment procedure can already be performed on chuck two. The combination of machine ID, chuck ID, reticle ID and the same conditions used for the bottom layer are considered as context data. The bottom layer is relevant because the overlay is defined with respect to the bottom layer.

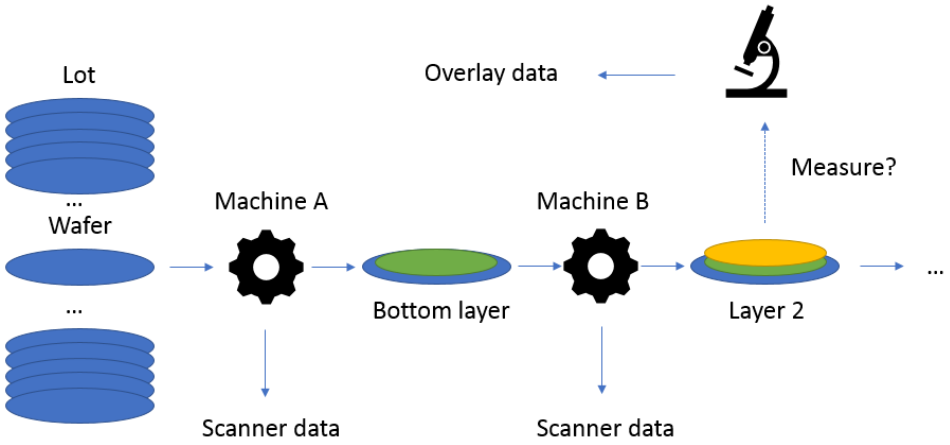

Figure 2.3: Illustration of the process in which scanner data and overlay measurements are generated.

Context parameters contain valuable information, but they are categorical variables. To use these variables in a regression model they are translated to numerical values using one-hot encoding[1], leading to a multitude of binary variables. Arguably, there could be more effective ways to include the context in the regression model, but optimally handling context parameters is outside the scope of this thesis.

2.3. Interactive visualization

Active learning is only one part of a multitude of methods available to the data scientist, which aim to use sparsely available data in an optimal way. It can be considered as a small part in a broader activity to, generally spoken, provide more insight with fewer measurements. Insight in this case is a combination of predictive performance and informativeness of visualizations, which is difficult to quantify. The ultimate goal is to assist an expert in the exploration of a dataset. Adding a measurement is one of the actions the expert could perform to add information to the model. Additionally, he or she could provide knowledge directly to the model or interact directly with a visualization.

The aspect of interaction with an expert, and the aim to provide an 'informative' visualization are not central to this thesis. However, in order to be useful for the broader scope it is important that a regression method and active learning strategy fit into such an interactive system. Therefore, certain design decisions are influenced by the context of interactive visualizations.

Also, interactive visualization is not limited to regression problems such as this overlay prediction setting. The idea for the current project was inspired by an imbalanced classification problem of a rare error occurring in a complex process. Some initial research was done for this problem specifically, which is summarized in appendix A, but it is not immediately relevant for this thesis. For practical reasons, specifically the availability of a suitable dataset, this research path was ended before any active learning was investigated.

[1] One-hot encoding is a transformation from a categorical variable with n categories to n binary variables.

2.4. Research goal and questions

The general goal of this thesis is to design an overlay measurement selection strategy for faster improvement of the visualization and predictive performance in the case of few measurements. The selection strategy should be able to select a wafer based on the data received from the scanner and the context variables. To achieve this goal, a number of research questions need to be answered:

- *Which active learning strategies are available in research, and how can they be adapted to a regression model that enables both prediction and visualization?*

  This question will be answered in chapters three to five, describing first the model for prediction and visualization, then the active learning frameworks available in literature and finally the specific implementations. It is impossible to discuss all known methods and solutions, so the aim is to give an overview of the most common frameworks that are relevant to this specific problem. The implementations serve as examples for how a general framework can be adapted to different models.

  The predictive model is essential to the question of which active learning methods are applicable, but the model itself is not the main interest of this thesis. Therefore it is discussed in chapter three and assumed as the basis for the rest of the thesis. A literature review of active learning frameworks is presented after the predictive model has been established and the active learning methods are adapted to the model. The adaptation leads to a number of challenges, for which different solutions are discussed and implemented to evaluate the effect on performance.

- *Do the active learning strategies lead to an improved predictive performance for this specific dataset?*

  This question is answered in chapter five, where the specific implementations are evaluated through experiments. The performance measure is discussed in chapter three and the result of an experiment is a learning curve, which shows the progression of performance over time. If active learning is beneficial, the learning curve is steeper leading to improved performance at some point. The learning curves provide insight in terms of if and when the strategy works better or worse. It is possible that a strategy works well in some part of the curve, but worse later on, so this is not a yes or no question.

- *Can we expect these results to generalize to a different dataset?*

  If the active learning strategy were to be used in practice, there has to be some confidence that the results in this thesis can be reproduced for different data. To get an idea of generalization, the same experiments are performed on a synthetic dataset. The synthetic data has a Gaussian distribution, but it is designed to have the same covariance matrix as the original data. The conclusions drawn from a synthetic dataset do not necessarily generalize to any other dataset, but it will give some insight in the specifics of the overlay dataset with respect to a Gaussian distribution.

- *Can we expect these results to generalize to a different model?*

  In practice, and especially in this field, labeled data is valuable. It is therefore very possible that the data sampled with one model in mind will also be used to train a different model in the future. Therefore, it is important to know the impact of active sampling on a different model. This question is answered by looking at the impact of model complexity on the active learning results in chapter five, but also by evaluating the selected samples on a different model in chapter six. By increasing the model complexity the active learning method can be evaluated under different conditions, but it is not the same as generalizing to a different model. Another regression model is considered by using the samples selected through the best performing active learning method, and evaluating the performance of a different regression model on these samples.

3

Regression methodology and baseline results

This chapter describes the methods used for preprocessing, prediction and evaluation. These methods can be seen as a framework in which active learning strategies can be developed and tested. Note that the active learning strategy is tightly coupled to the specific regression model and evaluation method, so results achieved in this context can not be easily translated to other regression methodologies.

3.1. Preprocessing

Features are scaled and centered to zero mean and unit variance. Often, only a small part of the data is available during training of the regression. The variances within a subset of the data, used for training, are compared to the rest of the data. If the variance in the data is very large compared to the training set for a specific feature, this feature is temporarily removed from the data. Without this correction, a linear regression is likely to wrongly estimate the contribution of this feature. This can be the case for one-hot encoded context variables, when a specific context is not present in the training set. The result is a set of approximately 300 features.

3.2. Partial Least Squares Regression

Partial Least Squares Regression (PLS) is a supervised feature extraction method for multivariate regression problems, which is especially effective when the number of predictors is large and there is collinearity among the independent variables. This makes it a good candidate for this dataset, where we do expect collinearity and redundant features. The goal of PLS is to extract a number of latent variables from both the predictors (X) and the targets (Y) that have a large predictive power. It combines aspects of PCA[1] and multiple linear regression [1] [29]. The extracted features can be used for regression, but also to generate a supervised visualization of the data as shown in Section 3.4.

Provided a set of training data consisting of n pairs of predictors (X) and targets (Y), the first result of PLS regression is a decomposition of the $(n \times d)$ matrix of independent variables X into p components:

$$X = TW + E_x,$$

where W is a $(p \times d)$ weight matrix containing the components or latent variables and the $(n \times p)$ matrix T contains the scores for each of the instances in X. E_x is a residual that contains the information that could not be encoded in TW, and it decreases as the amount of components p increases. The component matrix W can be used to project any new sample to the PLS space of reduced dimension p. The difference between the PLS decomposition and a PCA decomposition is that components in PCA describe covariance within the predictors, while the components in T describe the covariance between predictors and response variables. PLS can extract as many components as the rank of X, so the dimensionality of the projected space is one parameter that should be optimized empirically.

[1] Principal Component Analysis, an unsupervised linear feature extraction method [24].

Once the latent variables are found, the next step is to predict the targets. The prediction model for Y is:

$$Y = TQ + E_y,$$

where Q contains the linear regression coefficients for the targets in Y and extracted features in W, and E_y is the residual. The residual E_y is minimized in a least squares sense.

The optimal projection W is found through simultaneous decomposition of both X and Y. For a more thorough explanation of the PLS technique, including an implementation through the NIPALS algorithm, see the article by Abdi [1].

A prediction $\hat{Y}$ can now be made for samples X by first projecting the samples to p dimensions using W and then estimating the targets using Q:

$$\hat{Y} = XWQ$$

Figures 3.1 shows the first two extracted features for scanner data (X) predicting overlay targets (Y), coloured by the dominant context which is the machine ID. In this linear projection there is no clear separation of all four machines, though some machine-dependent structure is already visible.

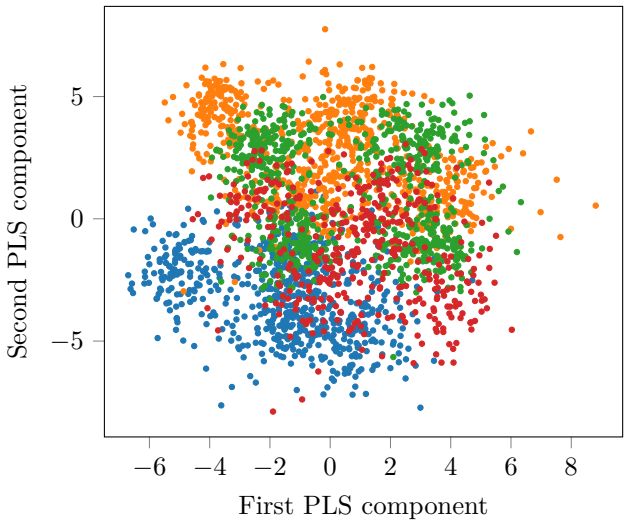

Figure 3.1: PLS projection of only relevant layers with all target variables. Colored by scanner ID's.

3.3. Evaluation

To compare and interpret prediction models, it is important to consider the method and measure of evaluation. To this end, the performance is measured through randomized cross-validation with R^2 as a performance measure.

3.3.1. Cross-validation

Evaluation of the performance of a prediction model should aim to give a reliable estimate of performance for unseen data. It is therefore essential that performance is measured on data that was not used to train the model. Closest to a realistic scenario would be to order the measurements chronologically and assert that all training data should precede the test set. Although this is realistic, it limits the amount of scenarios available for evaluation with a limited dataset. The choice of train and test set can have a large influence on the measured performance, especially in the case of (groups of) outliers. A reliable estimate therefore requires a number of different splits of the data, and letting go of the chronological ordering increases the amount and the diversity of all possible selections, therefore potentially improving the performance estimate. The performance will therefore be evaluated through randomized cross-validation where the data is randomly divided into k folds. In k iterations, each fold is used once as a test set while the other folds are used to train the model, as illustrated in Figure 3.2. The resulting performance is an average across these k folds.

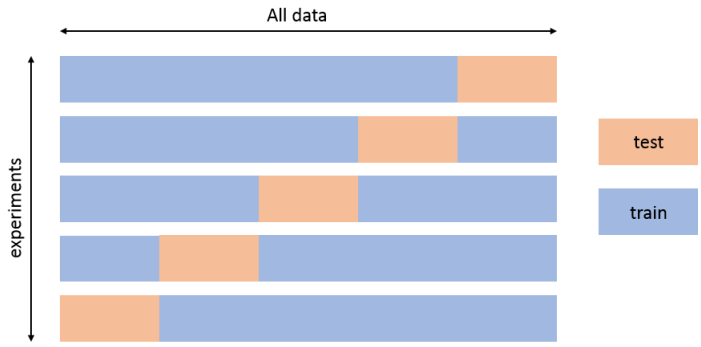

Figure 3.2: Illustration of k-fold cross-validation, with $k = 5$.

3.3.2. R-squared performance

A common way to evaluate performance in a regression model is to report the mean squared error (MSE) of n predicted targets ($\hat{y}_i \in \hat{Y}$, for $i = 1, ..., n$) and the true values ($y_i \in Y$), which is defined as:

$$MSE(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

The R-squared performance is a normalized MSE, with the advantage that it can be easily interpreted and compared to performances in other problems. It is defined as:

$$R^2(Y, \hat{Y}) = 1 - \frac{MSE(Y, \hat{Y})}{MSE(Y, \bar{y})}$$

where $\bar{y}$ is the mean of Y. A perfect prediction yields an R^2 of one, and when R^2 is zero it means that the model performs equally well as a constant prediction of the mean. Anywhere between zero and one the R^2 can be interpreted as the fraction of variance in the results that is explained by the model. Although the performance cannot exceed one, it can become negative for biased models. A negative R^2 means that the prediction is worse than a constant estimate of the mean target value $\bar{y}$, so $\hat{y}_i = \bar{y}$ for $i = 1, ..., n$. Note that $\bar{y}$ is unknown during prediction.

3.3.3. Performance with all data

Table 3.1 lists R^2 performance results for each of the ten overlay parameters, achieved through ten-fold cross-validation. The mean and standard deviation over the ten folds are listed. The PLS regression model allows the choice of dimensionality of the projected space, for which three options were tested: $p = 2, 10$ and 50 dimensions. With these results it is clear that a two-dimensional projection does not capture the full potential of the original high-dimensional data.

	2D		10D		50D	
Target	Mean	Std dev	Mean	Std dev	Mean	Std dev
Tx	0.098	0.057	0.189	0.101	0.230	0.127
Ty	0.150	0.086	0.320	0.071	0.433	0.067
WRotX	0.400	0.035	0.512	0.027	0.545	0.013
WRotY	0.425	0.047	0.600	0.061	0.726	0.024
WExpX	0.081	0.042	0.179	0.042	0.323	0.092
WExpY	0.265	0.056	0.439	0.056	0.680	0.031
FRotX	0.252	0.044	0.312	0.073	0.338	0.092
FRotY	0.248	0.042	0.325	0.069	0.334	0.101
FMagX	0.105	0.043	0.260	0.049	0.337	0.120
FMagY	0.081	0.087	0.254	0.103	0.325	0.108

Table 3.1: Cross-validated R^2 performance results for PLS regression using 2D, 10D and 50D projections.

Although PLS can be performed on all target values together, this would also require a performance measure that takes all targets into account. This may be interesting for the sake of overlay prediction, as collinearity between the targets can be considered, but it complicates the analysis of active learning methods in general. For the sake of simplicity one target is selected to predict and evaluate the performance. Unless stated otherwise, the target is 'WRotY'. For this target the PLS regression achieves the highest performance with all data, offering most potential for active learning strategies.

3.3.4. Learning curves

The effectiveness of an active learning strategy can be expressed through the learning curve, when compared to a baseline of random selection. The learning curve is generated by labeling points and evaluating the performance with the available data so far. The performance is expected to increase with an increasing amount of data, though this is not guaranteed. A typical learning curve starts steep and gradually decreases in slope, eventually approaching a limit. An active learning strategy is considered effective if it generates a steeper learning curve than random selection, as illustrated in Figure 3.3.

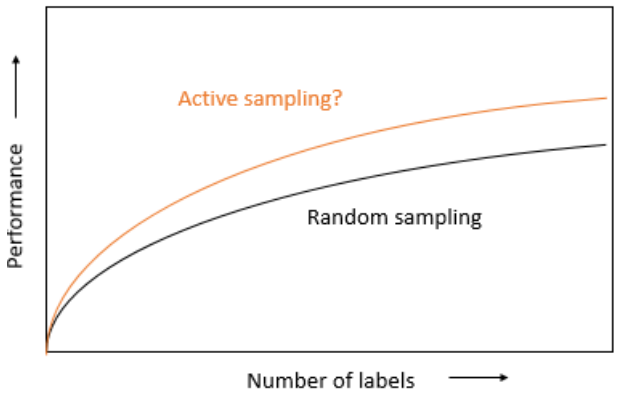

Figure 3.3: Illustration of a typical learning curve, with potential improvement due to active learning.

3.4. Visualization

Visualization has an important role in the broader scope of active learning and visualization. Although looking at visualizations offers no quantifiable performance measure, it can provide important insights. The aim of active sampling is not only to improve performance, but also to improve the visualization in 2D for a human expert. What constitutes a good visualization is subjective in general and dependent on the user and purpose of the visualization. In this thesis we aim only for performance increase and assume that it is directly linked to an improved visualization. Additionally, the visualization is used to visualize the effect of active sampling strategies in Section 3.4.

When PLS makes use of more than two components a t-SNE [18] space can be extracted to visualize the more complex relations in this high-dimensional space. T-SNE is a non-linear feature extraction method that aims to represent the high-dimensional structure of data in a lower-dimensional space and it tends to generate meaningful clusters, making it a powerful technique for visualization. Note, however, that t-SNE may also show artificial structures, especially when no inherent structure is present in the data. For an investigation into the behavior of t-SNE, see the interactive article by Wattenberg et al. [28]. Figure 3.4 contains an overview of visualizations for two and ten PLS components, where the regression was applied to a single target variable ('WRotY').

The visualizations are shown in three variants, which can be used to illustrate three relevant aspects to evaluate the data as projected to 2D. First of all, it is colored by the most relevant context, which is in this case the machine ID. By doing this, a user can recognize the structures shown in the visualization. Secondly, the samples are colored by target value, so that the expert can see how the structure is related to the target that he or she is interested in, and potentially evaluate how well the visualization relates to the target. A third visualization is included that shows the density of samples, as it is difficult to estimate the relative amount of samples in different clusters from the original scatterplot. The target value is well represented in both visualizations and it is clear that the machine context is an important feature for prediction.

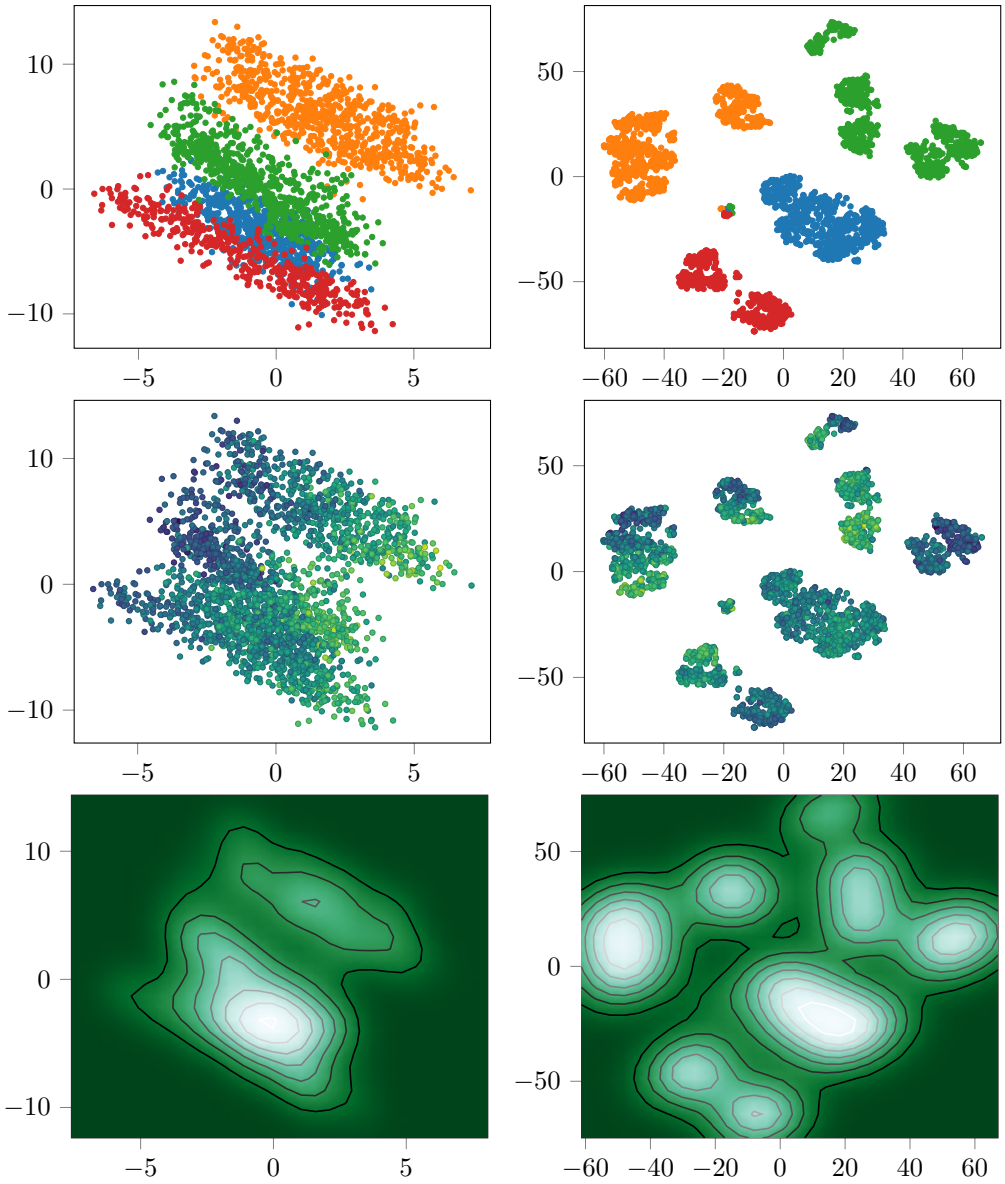

Figure 3.4: Visualizations using PLS Regression. Left: Projection by first two PLS components. Right: Projection by first ten PLS components followed by t-SNE to two dimensions. Top row: Colored by machine ID. Middle row: Colored by target value. Bottom row: density extracted using gaussian kernel density estimation.

4

Active Learning

This chapter contains an introduction of active learning theory and common frameworks for sample selection. Note that this is in no way a definitive overview of methodologies available in literature, but rather a description of frameworks that are relevant to this problem. After shortly discussing the types of active learning in terms of problem definition and a short introduction to bootstrapping, four general frameworks for active learning are discussed: uncertainty sampling, optimizing model impact, optimizing the expectation on unlabelled samples and optimizing distribution. This chapter concludes with a discussion of caveats to active learning.

Although bootstrapping is not strictly related to active learning, it will be used extensively in the active learning strategies described in the following chapters. Therefore, the bootstrapping method is described before discussing actual active learning strategies.

4.1. Definition of Active Learning

Active learning, in a broad sense, is the act of making intelligent decisions while collecting data. In this thesis, an active learning strategy is defined as a strategy for selecting the next sample to measure, to obtain a target value, in order to achieve an increase in performance on the test set that is as large as possible. In the description of these methods, the terms 'measurement', 'label' and 'target value' mean the same thing.

Although there may be varying definitions and modes of active learning across literature, there are generally two types: pool-based and stream-based learning (as defined by Settles [21]). In the former, the full set of unlabelled data is available from the beginning and the algorithm can sample these until all instances are labelled. This research assumes a different type which is stream-based active learning. In this case, the data is presented to the algorithm consecutively, possibly in batches of k samples. It is only allowed to select a certain number of instances from the batch to be labelled, and an entirely new batch is generated for the next iteration.

The choice of experimental method should depend on the intended application, which in this case is the situation in a fab. We assume that data analysis should not interfere with production by delaying wafers and that means that they are available for measurement in a stream-like way. A limited amount of wafers is available at any time and if a wafer is not selected, it will never be available for measurement again. Therefore, in this thesis, stream-based active learning is the more realistic approach and this is modeled using batches of k samples of which only one may be selected.

4.2. Bootstrapping

The term bootstrapping refers to random (re-)sampling with replacement. If a set of n values is sampled n times with replacement, this bootstrapped set will contain, on average, 63% of all unique values in the original set [14]. Bootstrapping has been known to be effective in estimating model performance [14], but also to create ensemble methods for prediction [7]. Many active learning frameworks require some estimate of uncertainty or variation in predictions, and PLS regression as a deterministic method, where the result is a single estimate of the optimal model, has no natural answer to this. Bootstrapping can provide a randomized variation in the model which enable the estimation of uncertainty and variation in prediction. This solution has been demonstrated to be effective for expected model change maximization in Cai et al.[4].

4.3. Uncertainty Sampling

Uncertainty Sampling can be used if the model provides some measure of uncertainty in the prediction, either explicitly or implicitly. Explicit uncertainty can be found in the posterior probability distributions of Bayesian models, while implicit uncertainty could be found in the distance to a decision boundary in classification problems, or the variation between nearest neighbours in a k-nn model. The strategy is to sample those points with large uncertainty, with the aim to improve performance in those areas where the model is least effective.

A slight variation on uncertainty sampling is 'Query by Committee' [23], which selects the samples on which a committee of models has most disagreement. The difference with uncertainty sampling is subtle, and depending on the model one might be more easily derived than the other. The algorithm requires a set of hypotheses (a committee) that are equally likely with the given data.

The original formulation of Query by Committee (QBC) [23] considered a binary classification problem with a committee of consistent and randomly drawn models. Querying a point where models disagree then decreases the version space of hypotheses that are consistent with the data.

In some problem settings with other prediction models, it is impossible to draw multiple consistent models and so the committee of models has different forms in literature. For a Bayesian model, drawing model parameters from their posterior distribution leads to equally likely hypotheses, which are used as a committee in Dagan et al. [6]. If the model offers no such possibility for random generation of different models, bootstrapping offers another solution to generating random and equally likely models [2].

Although all methods mentioned before apply to a classification problem, it may also be applied to regression. In this case the disagreement between models translates to the variance in prediction between models, as suggested by Cohn et al. [5], and QBC sampling is known to reduce prediction variance. If the model is unbiased this works well, but for biased models or noisy target values the benefit is not guaranteed [3].

In chapter five, the QBC method is applied to PLS regression by creating an ensemble of predictors using bootstrapping, but also by randomly excluding features from the analysis. The disagreement between models in the committee is defined as the variance in their predictions.

4.4. Optimizing model impact

To achieve fast improvement of the model, one approach is to elicit maximal impact on the model with one sample. The effect that a sample has on the model is very much dependent on the model shape, but also on the label. Since the latter is not known, deriving a formulation of the impact requires some prediction of the label. Implementation of this framework can be found in the expected gradient length for models trained by gradient descent [22]. Another maximal impact model has been derived for the case of rank learning using an SVM and a boosting method [8].

The common theme in these methods is that the current model is used to achieve some posterior probability for the label in a candidate sample. These probabilities are used as weights to value the impact of labelling the sample on the current model. In the expected model change maximization method [4], which was derived for regression problems, no such posterior probabilities are needed. Instead, an ensemble of bootstrapped predictors is used to estimate the error in a candidate sample, and the local gradient of model parameters is maximized.

A caveat to methods that maximize model change is that model change does not equate to positive model change, let alone a reduced error on the test set. Often, outliers are samples with large impact while they are not representative for the true distribution.

4.4.1. Expected Model Change Maximization

In this thesis, the Expected Model Change Maximization method [4] is implemented. This method is inspired by gradient descent optimization, but instead of optimizing the parameters of a model, this method optimizes the selection of a new point x_+ to add to the model. It is one specific example of an active learning strategy that optimizes model impact. The selection criterion $C(x_+)$ is the local gradient of the loss l with respect to the model parameters θ:

$$C(x_+) = \frac{\delta l(x_+, \theta)}{\delta \theta}.$$

Expected Model Change Maximization is easily derived for linear regression models and will likely sample in an exploratory way in this case. Therefore, it is a promising candidate for this active learning problem.

For linear regression, where the prediction models is $\hat{y} = \theta x$, EMCM has a simple solution. The least squares loss for points $\vec{x} = (x_1, ..., x_n)$ with labels $\vec{y}$ is quadratic and additive:

$$l(\vec{x}, \theta) = \frac{1}{2} \sum_{i=1}^{n} (\theta x_i - y_i)^2.$$

The change in loss due to a newly selected point x_+ therefore does not depend on the current loss:

$$\begin{aligned} l(\vec{x}, x_+, \theta) &= \frac{1}{2} \sum_{i=1}^{n} (\theta x_i - y_i)^2 + \frac{1}{2} (\theta x_+ - y_+)^2, \\ &= l(\vec{x}, \theta) + l(x_+, \theta), \end{aligned}$$

and the EMCM criterion with respect to point x_+ can be simplified to:

$$\frac{\delta l(x_+, \theta)}{\delta \theta} = (\theta x_+ - y_+) \frac{\delta \theta x_+}{\delta \theta} = (\theta x_+ - y_+) x_+.$$

The true value of y_+ is unknown, so this must be estimated in some way. In the original design of EMCM ([4]) the criterion was averaged over predictions made by a bootstrapped ensemble of k predictors $\hat{\Theta} = (\hat{\theta}_1, ..., \hat{\theta}_k)$. These predictors are trained with only a part of the current training set, and relying on their estimate of y_+ could be dangerous when the model is in an early stage of training. Therefore, in chapter five, multiple ways to estimate y_+ are implemented. The final criterion $C(x_+)$ for a bootstrapped model is:

$$C(x_+) = \sum_{j=1}^{k} |(\theta x_+ - \hat{\theta}_j x_+) x_+|.$$

4.5. Optimizing expectation on unlabelled samples

More sophisticated active learning models go one step further by formulating some expectation of future performance based on the pool of unlabelled samples or the test set. The goal is to reduce the error on these samples, of which the label is unknown, so a proxy for the true error is used in the form of an expected risk [20] [32]. Not only do these methods require a prediction of the sample label, but when an expected loss is computed this also requires some prediction of the rest of the labels before and after labeling the candidate samples. This make them computationally expensive, and they also rely heavily on probabilities generated by the current model, which are questionable if there are few labels available. Compared to uncertainty sampling and model impact, the greatest potential benefit is that the true distribution is considered and, if available, knowledge of the test set can be used.

4.5.1. Expected Error Reduction

In this thesis, the optimization with respect to unlabelled samples is implemented based on Expected Error Reduction, which means that the error is optimized directly. It is impossible to know the true error reduction, so these methods rely on estimations of this error. In the method by Roy & McCallum [20], the current error is estimated by assuming a loss function on the current model posteriors for each possible label for each element in a pool of unlabeled samples. When assuming a log loss function, the error E is estimated as:

$$E(D) = \frac{1}{n} \Sigma_{x \in X} \Sigma_{y \in Y} P_D(y|x) \log(P_D(y|x))$$

Where x refers to the elements in an unlabeled pool X of size n, and Y contains all possible labels. The probabilities P_D are estimated using the current model with the set of labeled samples D.

When considering a new sample, it is again considered with each possible label y and a new error $E(D + (x, y))$ is computed. The weighted sum of expected errors is the new expected error and the sample x_{new} that minimizes this error is selected to be measured.

This method assumes a classification problem with pool-based active learning, where the posterior probabilities can be estimated. It is computationally expensive, though depending on the method it could be optimized as discussed in [20]. For the regression methods in this research there is neither a discrete set of possible labels nor an explicit description of posterior probabilities, so implementation of expected error reduction is not straightforward and is discussed in Section 5.5.

4.6. Optimizing distribution

The most important caveat to active learning is that it imposes a sampling bias, meaning that the distribution sampled through active learning is not the same original distribution of the data. Although the bias may improve the model at first, it may skew the distribution over time causing eventual deterioration of the model predictions. This inspires active learning models that aim to optimize the distribution, so that the sampled distribution is a good representative of the true distribution. In many cases random sampling is already a straightforward way to achieve this, but this is not the case if the distribution changes over time. Active learning methods to deal with changing distributions are found in the field of transfer learning or covariate shift [27].

In this research we assume i.i.d. sampling for simplicity, which is enforced by randomly sampling the full dataset. In the actual implementation of an active learning strategy it is not guaranteed that the distribution of data is constant over time, but this future distribution of scanner data is most likely not known at the time of training. Together this is an argument to not consider methods that optimize distribution, for the scope of this thesis.

4.7. Caveats

Again, the sampling bias induced by active sampling is an important caveat to consider. Especially those method that sample where the model is most uncertain, or where samples have a large impact, are likely to sample outliers. Outliers are often poor representations for the underlying model, and unlikely to re-appear in the future, so a predictive model may not benefit from outliers at all.

Active sampling methods introduce a sampling bias that may (or may not) be beneficial to performance. Loog et al. [17] investigated the effect of this bias empirically and concluded that none of the state-of-the-art methods are guaranteed to improve performance for all datasets. Sugiyama [26] showed that specifically for misspecified linear regression models the sampling bias can have a detrimental effect on performance. Apart from performance, it must be noted that any further analysis such as estimation of posterior probabilities would be compromised by a sampling bias [17].

Another caveat to active learning is that the sampling methods are usually derived for a specific model and domain, and the conclusions with respect to performance results are only applicable to those models. In practice the dataset may be analysed with different models as new insights in the field arise. The sampling bias induced by active learning may have been effective for the original model, but no such guarantees exist for the future. This aspect of active sampling is also considered in the research questions of this thesis, and discussed further in chapter six.

Most methods are model-specific and converting them to other methods requires alterations. Combined with a different application domain, it is highly questionable whether similar results can be achieved. In the research questions the focus is on generalization of performance results to different models, and therefore the active learning strategies are evaluated with different model complexities and datasets in chapter five. The effect of sampling bias is shown by looking at visualizations of the sampled data in Section 5.6, but also by evaluating the data with a different model in chapter six.

5

Implementations and results

The frameworks for active selection described in chapter four can all be applied to the method of partial least squares regression, though some alterations may be required. In this chapter the specific implementation of three frameworks - Query by Committee, Expected Model Change and Expected Error - are described. Often, multiple interpretations are possible and each framework has slight variations in the implementation which may lead to vastly different results. The implementations are evaluated using active learning simulations applied to both the original data and synthetic data from a Gaussian model.

All experiments are executed according to the same general setup, which is described first. To put the results in perspective a set of baseline performance curves are discussed including the potential optimal sampling.

The methods and experiments in this thesis were executed using Python 3 and the scikit-learn library [19].

5.1. General experimental setup

Unless stated otherwise, experiments to generate learning curves with different selection strategies are executed in the following way:

1. A single target parameter is selected. Unless stated otherwise this is 'WRotY'.
2. The data is randomly split in five folds of train and test data, where each sample is selected for the test set at least once.
3. From the training set, an initial set of n_{start} samples is randomly selected.
4. The rest of the training set is ordered randomly and split into batches of size n_b. If not otherwise stated, $n_b = 10$.
5. For k consecutive steps, a single sample is selected from a new batch, according to the selection criterion.
6. After each step, a PLS regression model is trained using the initial set and all samples added so far. The R^2 performance is measured using the test set.
7. The learning curve represents the mean performance across the folds. Results in this chapter contain ten iterations, which is two sets of five cross-validation folds.

5.2. Baseline results

The goal of active sample selection is to improve the performance faster with the same amount of samples. This means that the aim is a steeper learning curve than the baseline, which is a curve generated through random selection. Figure 5.1 shows the learning curves generated by PLS regression with 2-, 10- and 50-dimensional projections when applying random selection. For the full amount of labelled samples it is beneficial to increase the model complexity, but in these learning curves it is clear that this is not the case for very few samples. The 50D PLS regression goes through a stage of severe overfitting before finally improving upon the 10D model after 400 samples.

Active learning strategies are expected to be effective when the model has not reached its full potential yet. Figure 5.1 suggests that this is below approximately 600 samples. The region of interest for this thesis is below approximately 200 measurements. For random selection we see that the low-dimensional models go through the steepest part of the learning curve in this region and the performance quickly converges to a

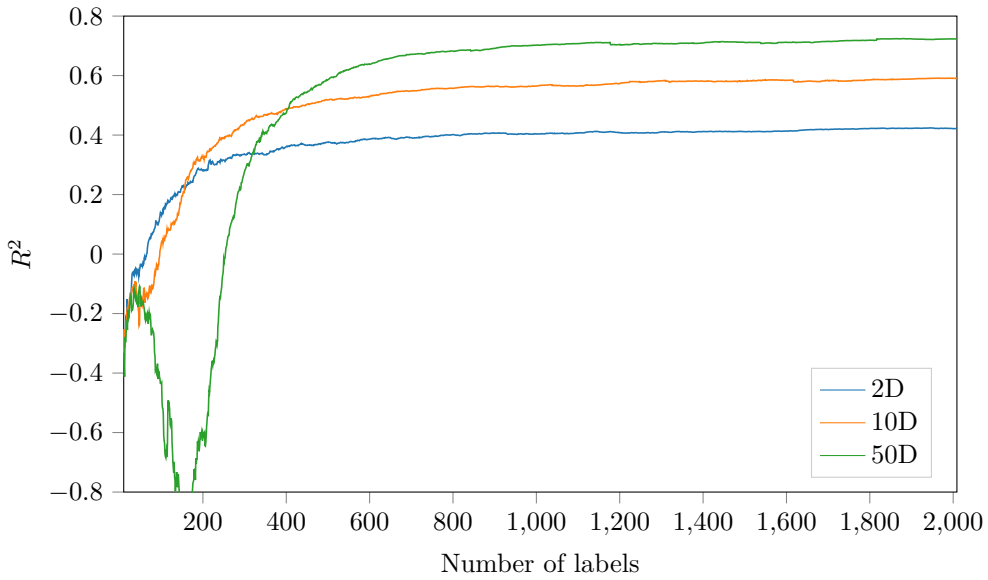

Figure 5.1: Full learning curves for random selection, using PLS regression models with 2D, 10D and 50D projections.

maximum afterwards. This also corresponds to the intended use case of the active sampling strategy, which is to train efficiently with very few available labels.

5.2.1. Optimal selection

To set an upper bound for the benefit of an active learning strategy, figure 5.2 contains the performance for an optimal selection strategy in the region of 10 to 200 samples. This optimal selection was simulated according to the method described in Section 5.1 and optimality is defined as highest R^2 performance on the test set after selection. It is optimal in a greedy sense, so it does not consider future selections. Note that both 2D and 10D optimal performance after 210 samples are higher than the limit of performance with all data. This means that there is also a benefit in leaving out certain measurements.

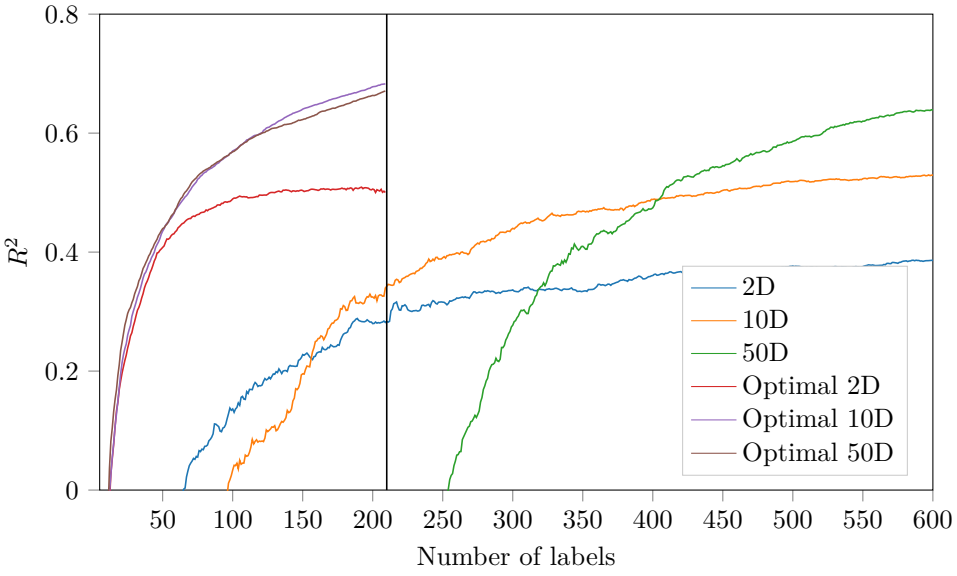

Figure 5.2: Full learning curves for random selection, and optimal results for active learning up to 210 samples. Using PLS regression models with 2D, 10D and 50D projections.

5.2.2. Synthetic Gaussian data

As a reference, a synthetic dataset was generated with exactly the same covariance structure as the original data, but with a Gaussian distribution. This means that each feature in the original data is represented in the synthetic data, and the covariances with other features and the target value is maintained. Figure 5.3 contains the baseline and optimal results for the Gaussian data. Although the model converges slower on this data, the same general behaviour emerges.

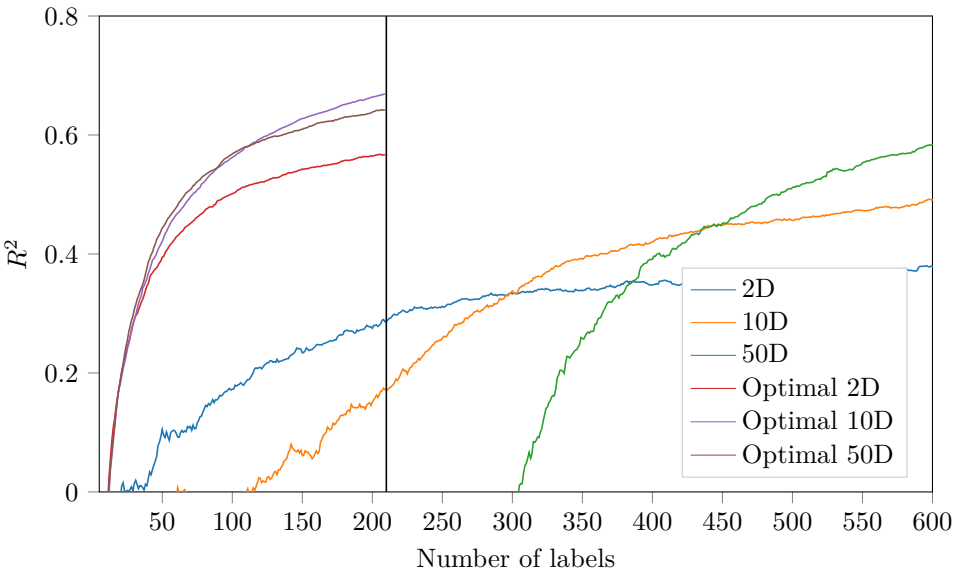

Figure 5.3: Full learning curves for random selection, and optimal results for active learning up to 210 samples. Using PLS regression models with 2D, 10D and 50D projections.

5.3. Query By Committee

The first active learning strategy to be implemented is Query by Committee. As PLS regression is a deterministic model it offers no natural variation or version space for application of the Query by Committee framework, so some randomness needs to be introduced. In this application we follow the method of Abe and Mamitsuka [2] and establish an ensemble of models through bootstrapping. Additionally, a method is tested where the committee is established by re-sampling the features rather than the samples. The two selection criteria are computed as follows:

1. Create an ensemble of five regression models $\hat{f}$ using bootstrapped versions of the training set ('Committee')
   or
   Create an ensemble of five regression models $\hat{f}$ using randomly drawn subsets of the features ('CommitteeFeatures')
2. For each sample x_i in the batch ($i = 1,...,10$), and for each model $\hat{f}_j$ in the ensemble ($j = 1,...,5$), predict the label $\hat{y}_{ij} = \hat{f}_j(x_i)$
3. Compute the variance of predictions y_i for each sample x_i, across the ensemble of predictors. Select the sample with largest variance.

Figure 5.4 contains the learning curves for the active sampling strategies mentioned above, comparing to random sampling. The synthetic data, generated using the same covariance matrix as the original set, shows approximately the same general trend as the original data but with a smaller standard deviation in the performance between iterations. The active sampling strategies show mostly the same performance as the random sampling within one standard deviation.

For the 2D projection there is some discrepancy between datasets, as random sampling performs better for the synthetic data and the 'Committee' method performs better for the original data. Note however that the differences are in the order of one standard deviation. At the end of the curve all methods converge to approximately the same performance. For a more complex PLS model with 10D projection there is also no significant difference between sampling strategies.

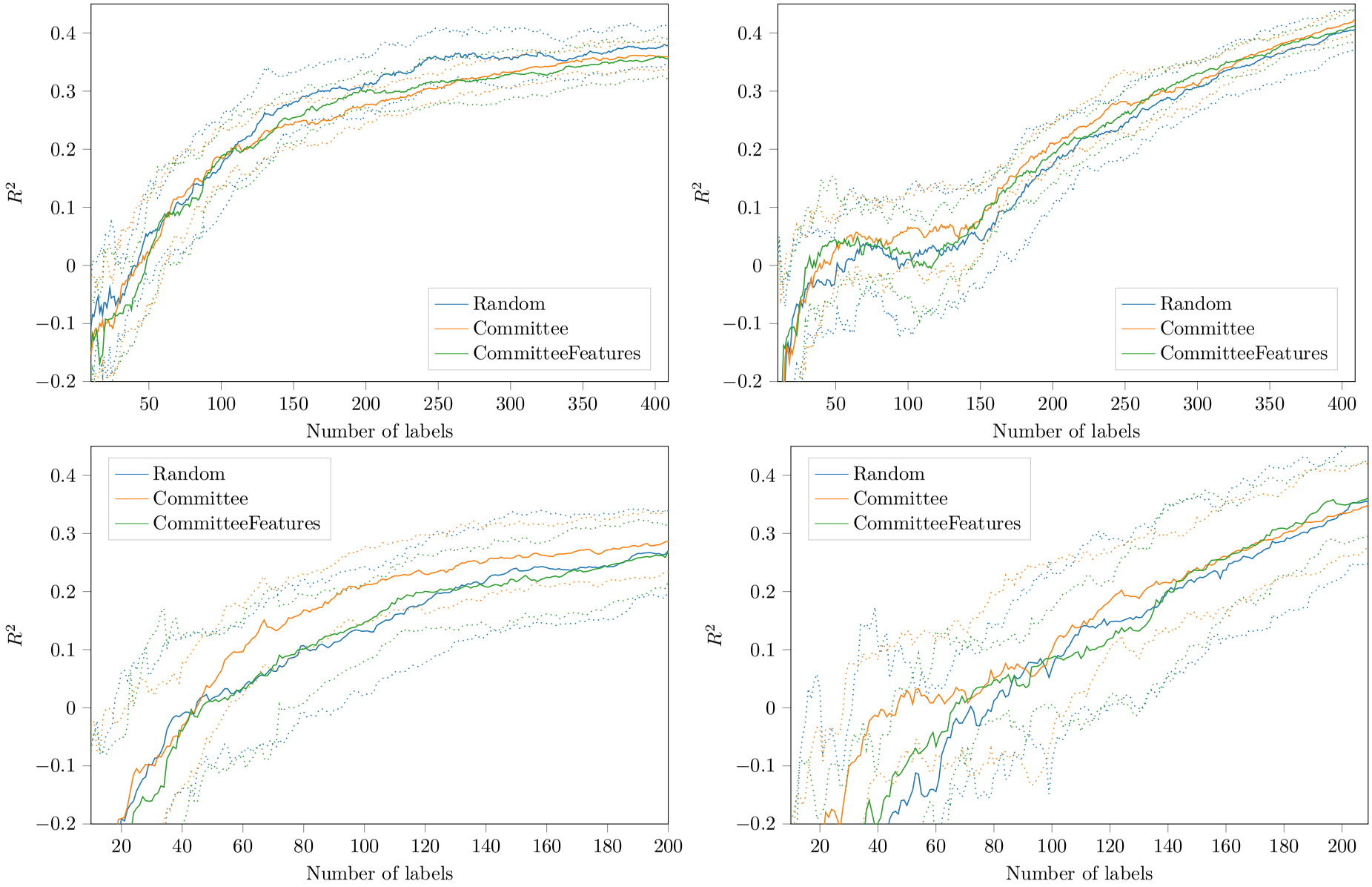

Figure 5.4: Mean learning curves for active sampling strategies and random sampling, over 10 iterations. The dotted lines indicate mean plus/minus standard deviation. Left: 2D PLS projection. Right: 10D PLS projection. Top: Synthetic Gaussian data. Bottom: Original data.

From these results we can not conclude either an improvement or deterioration of performance by the QBC sampling strategy. If any of the two methods provides benefit it would be the committee established through sample boostrapping.

5.4. Expected Model Change Maximization

As a second active learning method, the Expected Model Change Maximization (EMCM) by Cai et al. [4] is implemented. For PLS regression the parameters are contained in the decomposition W of the predictors and the regression coefficients in Q (see section 3.2). The prediction model can be described as:

$$\hat{Y} = XWQ = X\beta_{PLS}.$$

In essence, PLS is therefore a regularized linear model and the expected model change for the combination of interdependent variables W and Q can be expressed in the same way as for the simple linear model:

$$C(x_+) = (\theta x_+ - y_+)x_+.$$

Where θx_+ is the current model prediction, y_+ is some imputed value for the actual label and x_+ represents the independent variables. As x_+ is a vector, it must be replaced by the a vector norm. For PLS, a choice can be made between the original high-dimensional data or the projected value x_+W. If the latter is used, the EMCM criterion measures only model change in terms of the regression step Q. If the original value is used, the criterion combines and simplifies the projection and regression step. The actual result of this simplified model is analysed empirically.

The EMCM criterion is evaluated through experiments according to the method described in 5.1, where different choices for y_+ and x_+ are compared to each other and to the baseline of random sampling. The following criteria are included:

1. The current mean target value is chosen as a prediction of the true y_+. The value for x_+ is the norm of the original high-dimensional vector. ('EMCMMean')
2. A randomly selected ensemble of k previous target values is chosen as a prediction of the true y_+. The value for x_+ is the norm of the original high-dimensional vector. ('EMCMRandom')
3. Bootstrapping is used to generate an ensemble of k target values as proxies of the true y_+. The value for x_+ is the norm of the original high-dimensional vector. ('EMCMBootstrap')
4. Bootstrapping is used to generate an ensemble of k target values is chosen as a proxy of the true y_+. The value for x_+ is the norm of the projected value XW. ('EMCMBootstrapPLS')

5.4.1. Results

Figure 5.5 contains the average learning curves for the methods described above, for both the original data and a synthetic dataset. Again, there is some discrepancy between synthetic and real data. The distance between curves is larger for original data, but so is the standard deviation. For the 2D PLS model random sampling is among the best performing methods, although 'EMCMBootstrapPLS' has a slightly better mean performance - within one standard deviation - when analysing the original data. The EMCM criteria without boostrapping, using mean and random imputation of y_+, perform worse than random selection.

In the case of 10D PLS the results are different, as the random sampling is consistently among the worst performers. The best performing strategy is again 'EMCMBootstrapPLS' and with a significant distance to random sampling for the synthetic data. In this case all EMCM methods appear to learn faster while the randomly sampled model lingers at low R^2 performance, even the models that were clearly performing worse in the 2D setting.

From these results it appears that there is some benefit to the EMCM sampling criteria. However, it is important that the true value of y_+ is estimated properly through bootstrapping, as less careful estimations like the mean or random imputation may lead to worse performance than random sampling. The 'EMCMBootstrapPLS' model provides a performance that is consistently among the best models and often better than random sampling. These results do show that the complexity of the model influences the relative benefit of active learning methods, especially when looking at the performance of the 'EMCMRandom' and 'EMCMMean' models.

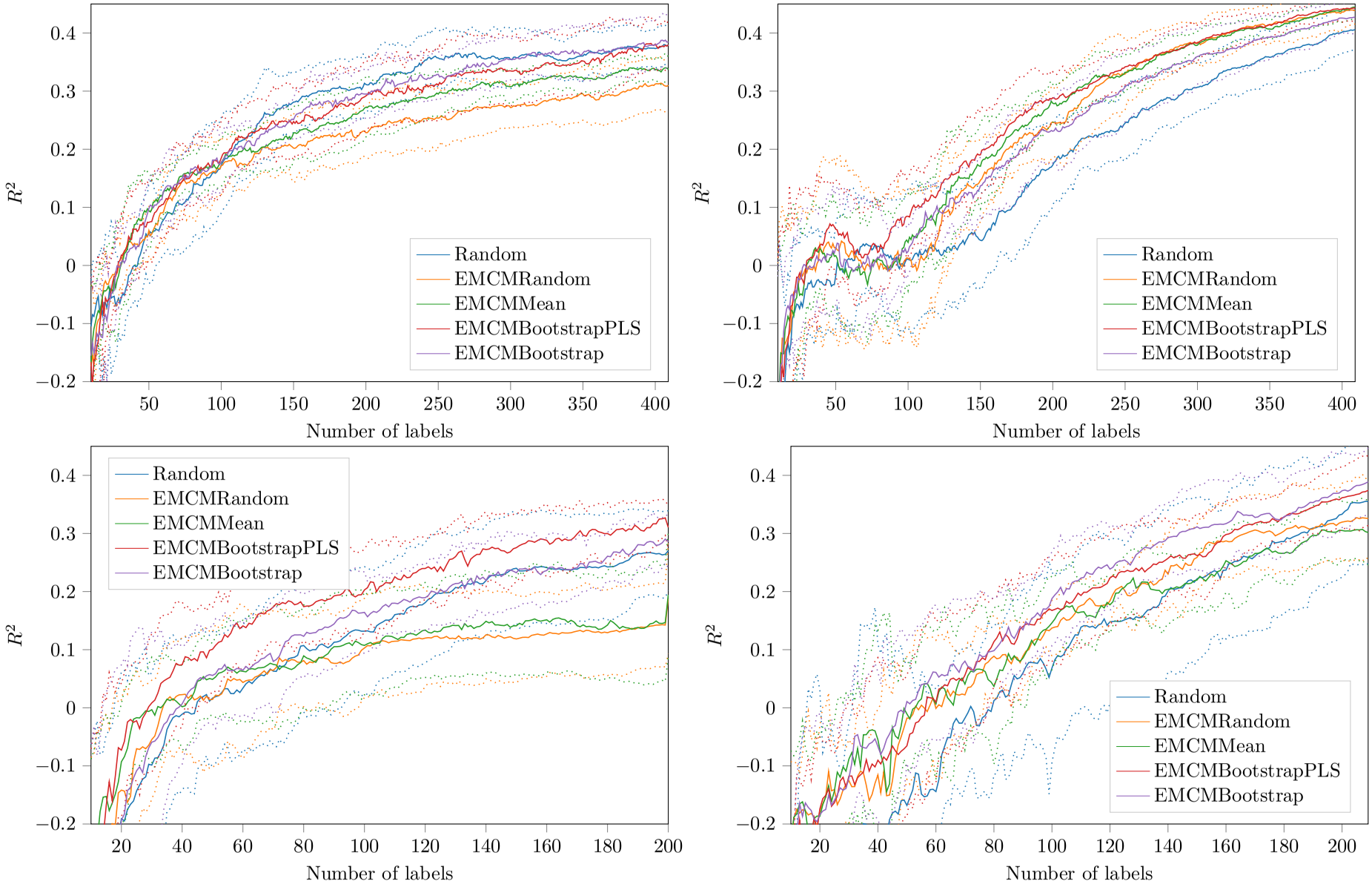

Figure 5.5: Mean learning curves for active sampling strategies and random sampling, over 10 iterations. The dotted lines indicate mean plus/minus standard deviation. Left: 2D PLS projection. Right: 10D PLS projection. Top: Synthetic Gaussian data. Bottom: Original data.

5.5. Expected Error for PLS regression

As the PLS regression model does not have an explicit estimate of posterior probabilities, the algorithm presented in Roy et al. [20] can not be applied without alterations. To translate the intention of error reduction an alternative definition of expected loss is required. The log loss for classification problems can be interpreted as a measure of entropy, or uncertainty in the predictions. If the model is certain of a prediction the loss is small, and it increases as posterior probabilities for different predictions become more similar.

A proxy for expected error can be found in the variance of prediction. Geman et al. [9] pointed out that the error can be decomposed into three factors: noise, bias and variance of prediction. If we assume that the model is unbiased, or at least that we cannot optimize for bias directly, the prediction variance is the only factor of the error that we can optimize based on unlabeled data. The PLS regression model is deterministic by nature and therefore has no natural variance in predictions, but bootstrapping offers a possibility to introduce and measure variance.

5.5.1. Algorithm for Expected Error Reduction

The algorithm for expected error reduction using the PLS regression model is as follows:

1. Consider each unlabeled example, x, in the batch as a candidate for next labeling request
2. Generate an ensemble of possible labels, $\hat{y} = (\hat{y}_1, ..., \hat{y}_k)$, for x, using bootstrapped PLS models and add the pair $(x, \hat{y}_i)$ for $i = 1, ..., k$ to the training set
   (a) re-train the classifier with the enlarged training set, $D + (x, \hat{y}_i)$
   (b) estimate the resulting criterion on the test set
   (c) assign to x the average value for the criterion on the test set for the ensemble of possible labels, $\hat{y}_i$ for $i = 1, ..., k$
3. Select the sample that optimizes the criterion by choosing the min- or maximum depending on the objective

The variation between implementations is found in the criterion that is used as a proxy for expected error. As suggested above, minimizing the variance is expected to improve prediction accuracy. As a comparison, the opposite target is also implemented which is to maximize the variance in prediction. Finally, one criterion is implemented where the covariance between target and PLS components is optimized as this bears a direct relation to the PLS regression model, where the components of projection have a maximal covariance to the target values. The following criteria are used:

1. The minimal mean variance between predictions on the test set, made by a bootstrapped ensemble of predictors ('ExpErrMinVar').
2. The maximal mean variance between predictions on the test set, made by a bootstrapped ensemble of predictors ('ExpErrMaxVar').
3. The maximal covariance between the predicted target value of the test set and the first PLS component ('ExpErrMaxCovar').

5.5.2. Results

Figure 5.6 contains the average learning curves for the selection methods mentioned above. There is no method that clearly performs better than others, as most curves are within one standard deviation of random selection. However, there is one clear method to avoid, which is the 'ExpErrMaxVar' in the case of a 10D PLS projection. The performance for this method remains so low that most of the curve is outside the scope of the figure.

Although these results, again, show no benefit of the expected error framework with respect to random sampling, they do show the negative influence that sampling may have. This negative impact of maximizing the expected variance in prediction ('ExpErrMaxVar') is especially prevalent in the original dataset, while the synthetic data does not suffer at all. This is likely due to the fact that each sample in the synthetic data is generated according to the same model, and no outliers or inconsistent samples exist. The original data clearly does have a number of samples that confuse the regression model and this sampling strategy is very effective in finding those points.

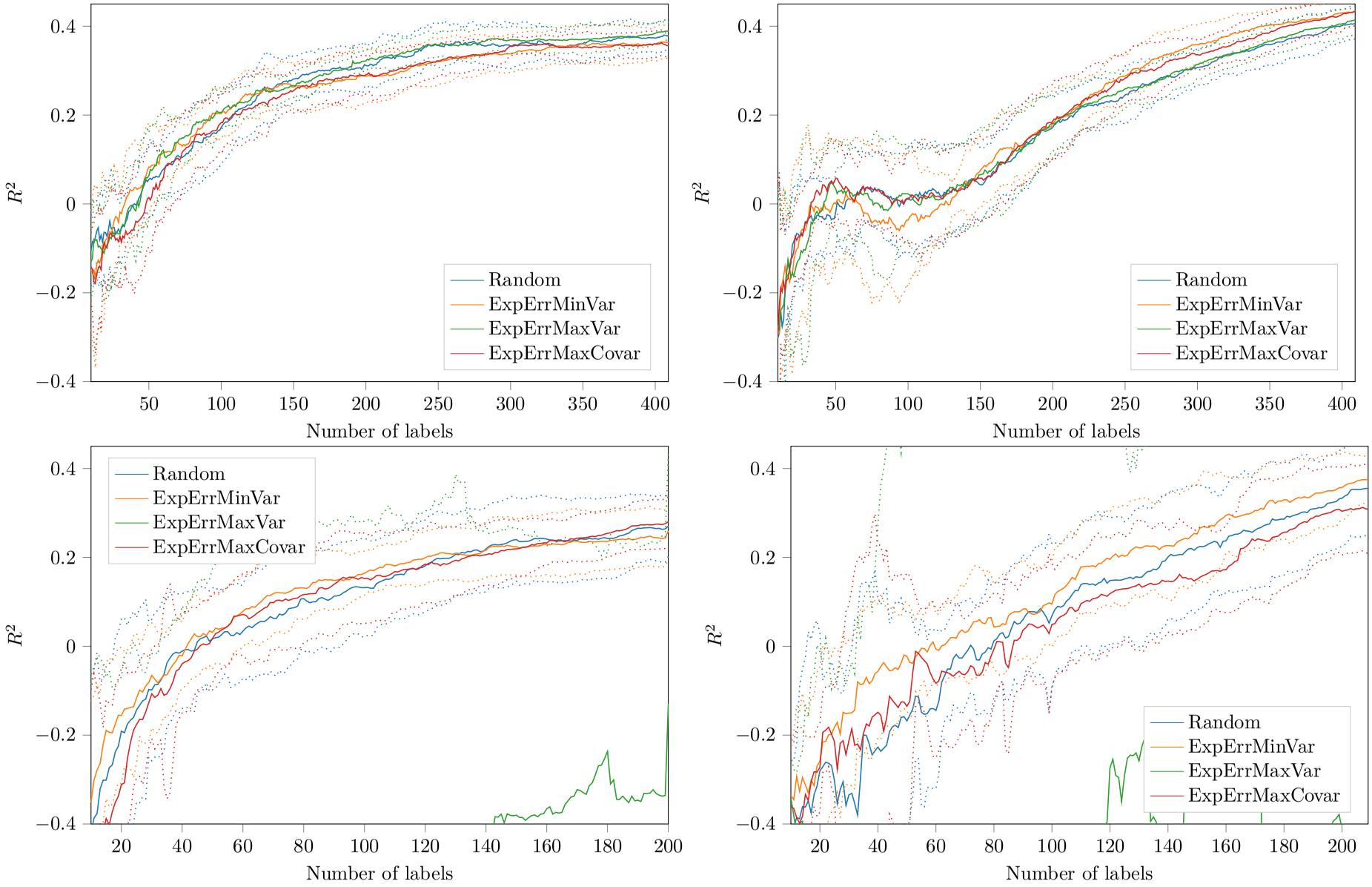

Figure 5.6: Mean learning curves for active sampling strategies and random sampling, over 10 iterations. The dotted lines indicate mean plus/minus standard deviation. Left: 2D PLS projection. Right: 10D PLS projection. Top: Synthetic gaussian data. Bottom: Original data.

5.6. Visualization of selections

To gain insight in the results of the different selection strategies, the selections can be visualized. Figure 5.7 contains a 2D projection of the selections, generated by applying a 10D PLS projection followed by a t-SNE transformation to the full dataset. Only the 200 samples selected by the active learning algorithm over the course of a simulation are shown in the figure, for each of the ten iterations, resulting in a distribution of 2000 samples which may contain duplicates.

Looking at visualizations of the selected samples gives a general idea of the reason that some sampling strategies do not work well. The random sampling serves as the original distribution in this Figure 5.7. The worst performing strategies clearly undersample certain clusters (see 'ExpErrMaxVar' and 'EMCMRandom'). The best performing model ('EMCMBootstrapPLS') seems to sample each cluster evenly, slightly undersampling the high-density regions and selecting more from the low-density regions. One can imagine that this works well, because it provides the model with enough data to estimate the overlay for each cluster. For models that show similar performance to the baseline there is also no clear difference in the visualized distributions.

The same visualizations for synthetic data are shown in Figure 5.8, but here the differences are less obvious. This is not surprising, as the differences in performance are also less clear in the learning curves for the synthetic data. Again, however, the 'EMCMBootstrapPLS' model does not sample any region with high density, but rather seems to create a more constant density across the input space.

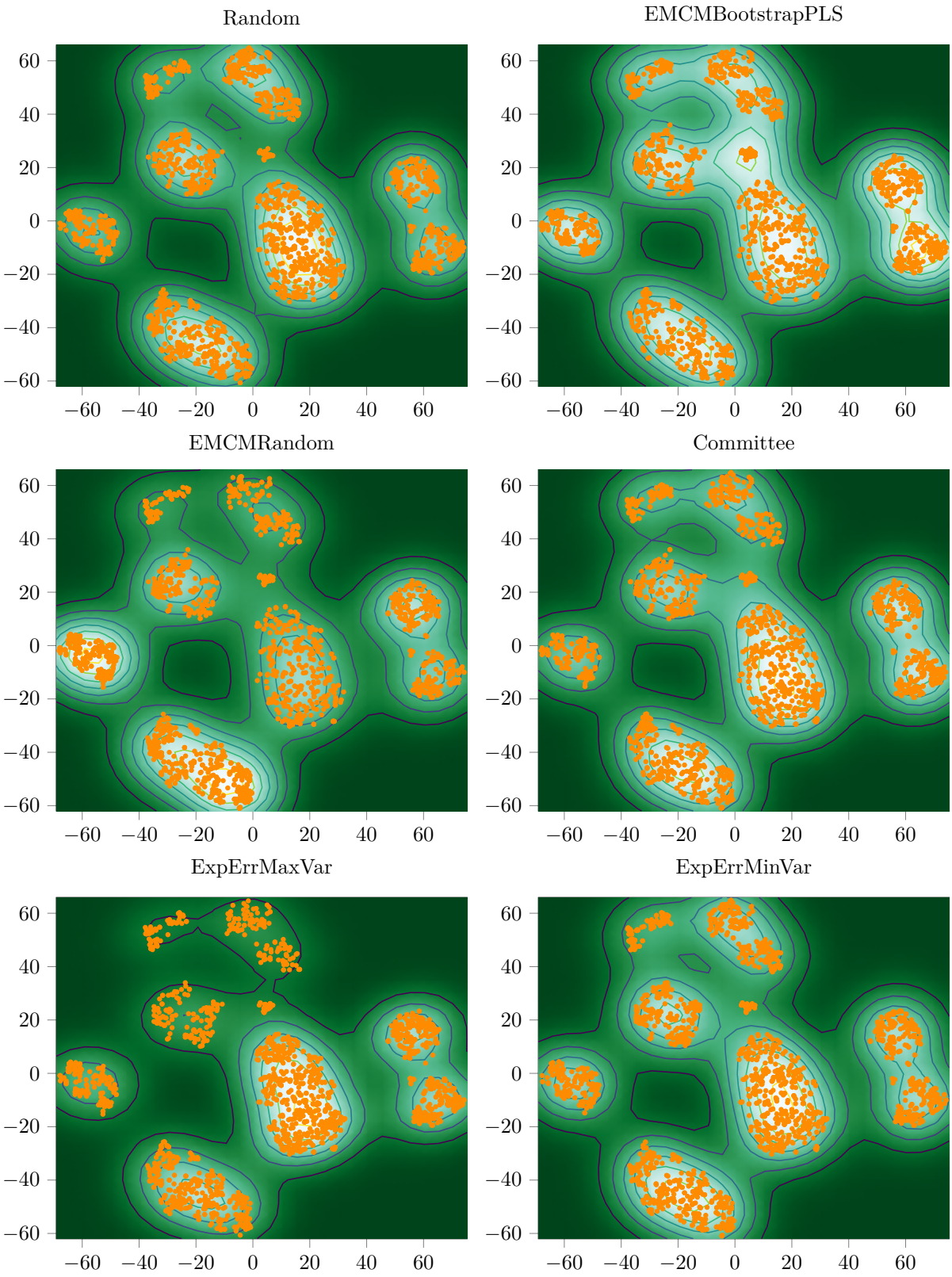

Figure 5.7: Selected points in t-sne space with different active learning strategies. Density using Gaussian KDE.

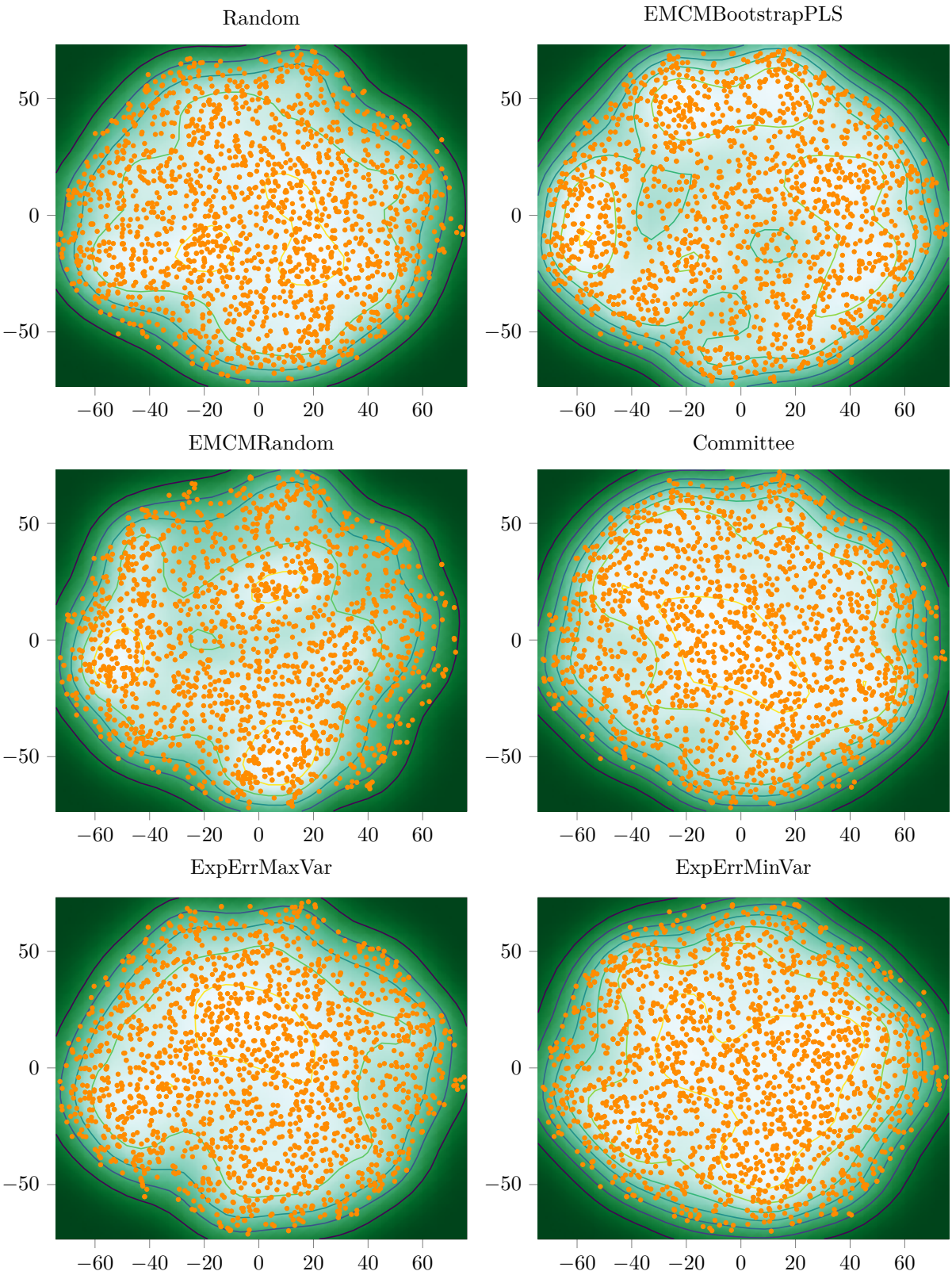

Figure 5.8: Selected points in t-sne space with different active learning strategies. Density using Gaussian KDE.

6

Comparison to probabilistic model

The question to be answered in this chapter is whether the selected samples that would benefit a PLS regression model, would have a similar effect on another model. One hypothesis would be that if a sample is informative to one model, it will also be informative to a different model so the effect on the learning curve should be similar. Another hypothesis, however, is that an active learning method that is tailored to a specific model leads to a specific sampling bias, that is more likely to have a negative effect when analysing the data with a different model. This chapter is an attempt to analyse the impact of actively sampling for a specific model (PLS regression) and using these samples to train a different model.

The model of choice is the probabilistic 'Spike and Slab' (SnS) model, which is also a linear model. An advantage of the SnS model for user interaction is that it estimates the probability that a feature is relevant, which enables further understanding of the problem. This makes it an interesting model in the context of informative visualizations, and therefore relevant in the broader scope of this thesis.

6.1. Spike and slab model

The 'Spike and Slab' (SnS) model [13] assumes a linear regression model where many of the variables are irrelevant to the actual prediction. The name refers to a prior on the model parameters which is composed of a Dirac δ-function at zero ('spike') and a Gaussian centered around zero ('slab'). This means that with a probability p, a variable does not contribute at all to the model. With a probability $1-p$ the contribution is still limited by the Gaussian distribution. The prior expectation for any regression coefficient β_i for $i \in 1,...,d$ is:

$$\beta = \gamma N(0,\sigma) + (1-\gamma)\delta(0),$$

where $N(\mu,\sigma)$ is the normal distribution with mean μ and variance σ and $\delta(0)$ is the Dirac delta function centered at 0. The parameter γ is a Bernoulli random variable depending on the probability p:

$$\gamma = B(p).$$

The parameters p is also learned by the model, but is set with a prior expectation ρ. This prior expectation is combined with evidence provided by the samples, to generate a posterior distribution for β. This estimation is performed through expectation propagation [13]. The computation is mathematically complex and computationally expensive, and the actual implementation is not within the scope of this thesis. The posterior of the model parameter p is a direct estimate of the probability that a variable is relevant, as this is equal to the probability that the regression coefficient is inside the 'slab', so non-zero.

6.2. Comparison of performance

In order to compare performance, the selected samples for simulations with PLS regression are transferred directly to the spike and slab model, including the same test set and initial set. With these exact same datasets, for each step in the simulation, the spike and slab model is fitted and evaluated by R^2 performance. The resulting curves can be compared in Figure 6.1. In this analysis the prior ρ is set to $\rho = 0.5$.

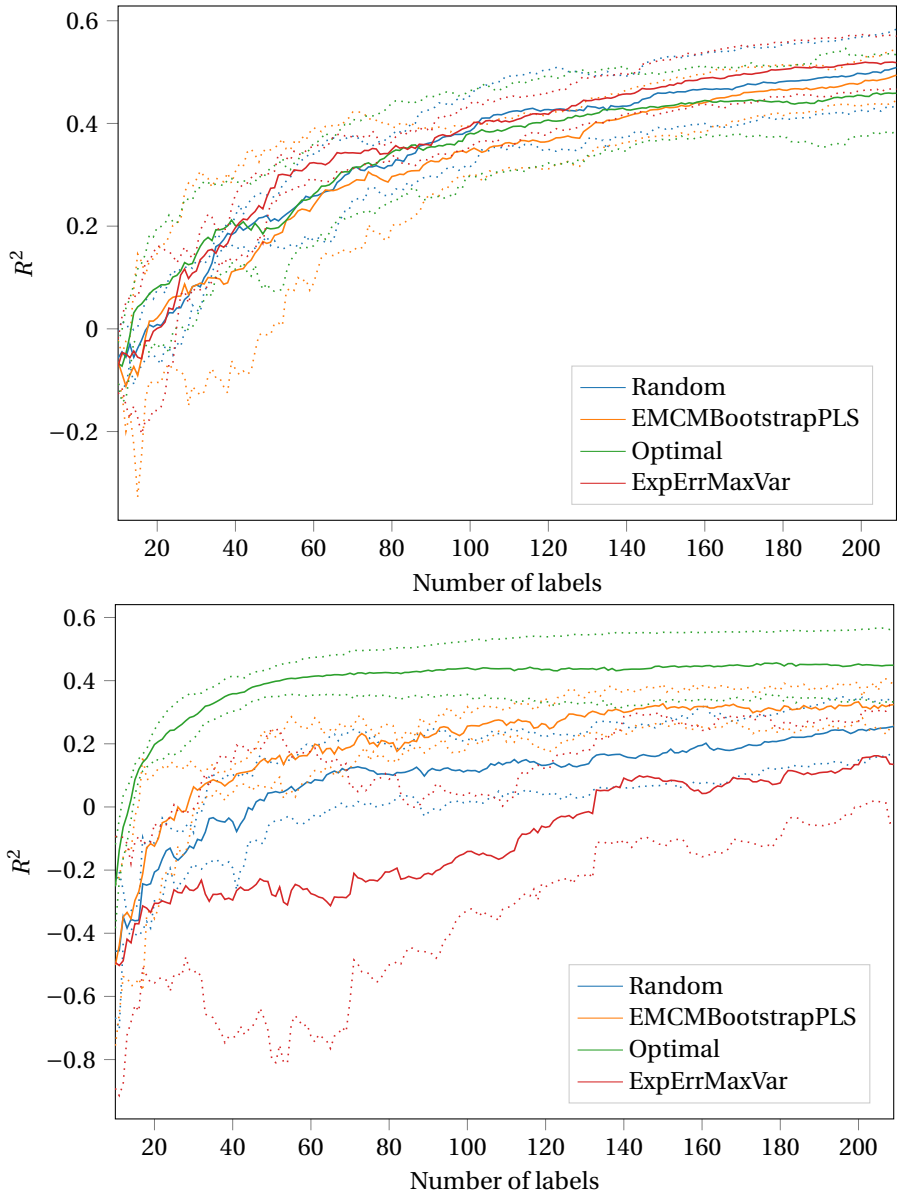

Figure 6.1: Mean performances with same datasets and selections for spike and slab (top) and the original PLS model (bottom).

First of all, the SnS model is more effective than PLS regression with few samples. This means that the regularization that SnS applies by feature selection is more effective than the regularization through feature extraction in PLS. The differences in performance shown in the PLS model, as a result of specific sample selections, are not apparent for the SnS model. In fact, the performance for each model is approximately the same and not significant when considering the standard deviation. The optimal strategy for PLS regression has no benefit for the SnS model, but also the strategy 'ExpErrMaxVar' that has a negative effect on the PLS model does not negatively effect the SnS model.

It is striking that the different samples that influence the PLS model have very little effect on the SnS model, even though they are both linear regressors. At least for this comparison, neither of the hypotheses stated in the beginning of this chapter has been confirmed. These results indicate that the SnS model is very robust to changing sampling distributions.

7

Conclusion and discussion

An empirical evaluation was done for a number of active sampling frameworks, with both the original overlay prediction data and randomly generated data from a Gaussian distribution. Although all methods are rooted in literature they require some interpretation and translation to the PLS regression model or even to a regression problem in general, which is not always straightforward. Many of the tools commonly used in active learning, such as uncertainty and variation in predictions, are not naturally available to a linear regression model like PLS regression. Random re-sampling, or bootstrapping, offers a solution to these questions as it adds a source of variation to the model.

Active sampling is a promising technique and speaks to the imagination of data scientists. Research in this field is as abundant as it is specific, to a specific problem definition or machine learning model, which makes it difficult to draw general conclusions. One conclusion we can draw from the results of this thesis is that active learning is more difficult than it may sound. From a broad set of established sampling techniques only the Expected Model Change Maximization seems to increase performance, in some cases, with respect to random sampling (Figure 5.5). Others are very much similar in both performance and perceivable distribution of samples (see 'committee' results in Figures 5.4 and 5.7) and some are almost guaranteed to lead to deteriorating performance (see 'ExpErrMaxVar' in Figure 5.4).

The EMCM strategy, estimated in the reduced PLS space, is relatively effective as it chooses samples where prediction is extreme in terms of value and sensitivity to model variation. This is where the model is likely to be overfitted and the learning curves also show that the benefit is mainly in the region of low performance (see figure 5.5, top right). The selected samples for the bootstrapped EMCM strategy seem to be more evenly distributed across the input space than the original distribution (see Figure 5.7).

In the comparison between the original data and a Gaussian model with the same covariance matrix, it seems that active sampling has less impact on performance when it stems from a perfect Gaussian distribution. Especially when active sampling has a large negative impact on the performance it affects the original dataset more than the synthetic data (see Figure 5.5). This makes sense, because in the real world samples may be messy, the true model is likely to be non-linear and sampling the wrong points may confuse a linear model. When the underlying model is truly linear then the fitted model will likely not deviate too much as a result of the sampling strategy.

When fitting a different model, in this case the probabilistic and sparse Spike and Slab model, to the data sampled by active learning on the PLS model, there seems to be no benefit or loss to the performance. Even an optimal sampling on the PLS model does not improve the SnS model much (Figure 6.1). Note however that these are both linear models and the feature selection makes it a relatively robust model. These conclusions do not extend to other models, especially if they are non-linear.

One benefit of the PLS model, with respect to other regression models such as the Spike and Slab model, is in the visualization, as the combination of a PLS projection and t-SNE mapping generates visualizations that convey the underlying structure of the data in a way that relates to the target parameter. The t-SNE mapping also clearly separates the relevant contexts, in this case the different scanners, and therefore relates to a structure that an expert could recognize (see Figure 3.4). This would be a useful feature in the broader context of this thesis, where the goal is also to provide informative visualizations to an expert.

In general, this thesis does not show much benefit from active learning. This could be due to the nature of the linear regression model. If the underlying model is truly linear, and the expected noise in each sample is the same, then each sample contains approximately the same information. It makes sense to choose samples that show some variation in features, but random sampling is likely to achieve this already. This in contrast to, for example, a linear classifier where samples close to the decision boundary are clearly more informative than samples in an area where one class is already dominant.

So for a linear regression model random sampling is already an intelligent choice. If the regression model would be non-linear, such as a random forest regression or a locally linear regression, an active learning strategy could be tailored directly to the structure of the regression and use unlabeled samples to see where most benefit can be achieved.

7.1. Future research

With the conclusion and discussion in mind, this thesis leaves a number of problems for future research. The PLS model specifically does not seem to benefit much from active learning, and the likely cause is that the problem is too simple. Adding complexities to the problem might also introduces ways in which active learning can be beneficial. One way to introduce complexity is to consider a non-linear model and active learning strategies tailored to the specific workings of that model.

Also, this thesis considers a simplified version of the general problem of overlay prediction, in the sense that the samples are drawn i.i.d. from a known dataset. The data therefore contains no drift or sudden changes, no sampling bias with respect to the test set or even the introduction of new machines to the fab. These are only some of the challenges that a data scientist could face when implementing active sampling in practice. For the simplified problem no clear benefit can be seen from the active learning methods considered in this thesis, but if the distribution of train and test set are not equal there could be some benefit to active learning, as applied in the field of transfer learning.

A third way to adapt the problem is to consider a different performance measure. The R^2 performance assigns the same importance to each sample, but in practice it might be more interesting to find large overlay values. If this is represented in the performance measure, an active learning strategy can be tailored to predict these more interesting results. One way to adapt the problem is to consider a classification problem like in Appendix A.

With respect to the active learning methods evaluated in this thesis, the best performing active learning strategy, Expected Model Change, is rooted in the idea that new samples achieve maximal impact. However from the visualization it seems that this strategy is especially effective in sampling the input space evenly, even though that is not its explicit goal. It would be interesting to see why these things are apparently related and whether a different sampling strategy based on exploration of the full input space achieves similar results.

7.2. Recommendations for ASML

In this thesis a single use case, that of overlay prediction with few measurements, is considered and the problem was simplified by assuming that samples were independent and identically distributed across initial training set, batches and test set. In practice the i.i.d. assumption might be broken by, for example, the introduction of a new machine during training, machine events like the replacement of parts and adaptation of the production process. The sampling scheme could be adapted to deal with such events and that would be a specific use case of active learning, where it might be more beneficial than in the simplified experiments of this thesis. The user would likely be able to identify when the distribution of samples is likely to change, due to specific events, and user input could be used to adapt the sampling scheme. To develop an active sampling strategy that meets customer needs and is applicable in practice, ASML would have to collaborate closely with a customer fab.

Also, in the context of the fab, it is important to note that overlay measurements are used in overlay control through feedback. Choosing different measurements therefore would not only impact prediction, but also correction and therefore future overlay values. Before implementing any active sampling the impact on overlay control must be studied. Additionally, the sampling strategy could work in close collaboration with the overlay control by aiming for detection of slow drifts or outliers. Again, to develop this one would need input from the fab where these control loops are implemented, to find out which specific use cases are interesting and what the practical demands are.

Bibliography

[1] Hervé Abdi. Partial least squares regression and projection on latent structure regression (pls regression). *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):97–106, 2010.

[2] Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 1–9, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL `http://dl.acm.org/citation.cfm?id=645527.657478`.

[3] Robert Burbidge, Jem J Rowland, and Ross D King. Active learning for regression based on query by committee. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 209–218. Springer, 2007.

[4] Wenbin Cai, Ya Zhang, and Jun Zhou. Maximizing expected model change for active learning in regression. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 51–60. IEEE, 2013.

[5] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

[6] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995.

[7] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-45014-6.

[8] Pinar Donmez and Jaime G Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *Proceedings of the 25th international conference on Machine learning*, pages 248–255. ACM, 2008.

[9] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[10] Manuel Giollo, Auguste Lam, Dimitra Gkorou, Xing Lan Liu, and Richard van Haren. Machine learning for fab automated diagnostics. page 31, 06 2017.

[11] Dimitra Gkorou, Alexander Ypma, George Tsirogiannis, Manuel Giollo, Dag Sonntag, Geert Vinken, Richard Van Haren, Robert Jan Van Wijk, Jelle Nije, and Tom Hoogenboom. Towards big data visualization for monitoring and diagnostics of high volume semiconductor manufacturing. 05 2017.

[12] Patrick J Grother. Nist special database 19 handprinted forms and characters database. *National Institute of Standards and Technology*, 1995.

[13] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, and Pierre Dupont. Generalized spike-and-slab priors for bayesian group feature selection using expectation propagation. *The Journal of Machine Learning Research*, 14(1):1891–1945, 2013.

[14] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[15] Auguste Lam, Francois Pasqualini, Jean de Caunes, and Maxime Gatefait. Overlay breakdown methodology on immersion scanner. *Proc.SPIE*, 7638:7638 – 7638 – 12, 2010. doi: 10.1117/12.849245. URL `https://doi.org/10.1117/12.849245`.

[16] Auguste Lam, Alexander Ypma, Maxime Gatefait, David Deckers, Arne Koopman, Richard van Haren, and Jan Beltman. Pattern recognition and data mining techniques to identify factors in wafer processing and control determining overlay error. In *Metrology, Inspection, and Process Control for Microlithography XXIX*, volume 9424, page 94241L. International Society for Optics and Photonics, 2015.

[17] Marco Loog and Yazhou Yang. An empirical investigation into the inconsistency of sequential active learning. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 210–215. IEEE, 2016.

[18] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

[21] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[22] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[23] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 287–294, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130417. URL `http://doi.acm.org/10.1145/130385.130417`.

[24] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[25] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 905–912, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143958. URL `http://doi.acm.org/10.1145/1143844.1143958`.

[26] Masashi Sugiyama. Active learning for misspecified models. In *Advances in neural information processing systems*, pages 1305–1312, 2006.

[27] Devis Tuia, E Pasolli, and William J Emery. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9):2232–2242, 2011.

[28] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016. doi: 10.23915/distill.00002. URL `http://distill.pub/2016/misread-tsne`.

[29] H. Wold. *Partial Least Squares*. American Cancer Society, 2006. ISBN 9780471667193. doi: 10.1002/0471667196.ess1914.pub2. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/0471667196.ess1914.pub2`.

[30] Lining Zhang, Lipo Wang, and Weisi Lin. Generalized biased discriminant analysis for content-based image retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(1):282–290, 2012.

[31] Xiang Sean Zhou and Thomas S Huang. Small sample learning during multimedia retrieval using biasmap. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[32] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.

A

Error Classification using Biased Discriminant Analysis

The main body of this thesis considers a prediction of the overlay in magnitude, resulting in a regression problem. However, initially, the active learning activity was aimed at predicting errors in a binary way. These binary errors could be related to a specific phenomenon or could simply be an error that exceeds a threshold value. In any case, the problem at hand in this chapter is a binary classification problem, and specifically the data is assumed to be imbalanced towards the non-error ('positive') class and errors are sparse.

Classification for highly imbalanced datasets is a difficult problem in itself, so before active learning is applied this problem requires an effective method. As visualization needs to be an integral part of the method, and the problem is high-dimensional with few labels, the first goal of the research is to find an effective feature extraction method to project the data to two dimensions.

In this appendix Biased Discriminant Analysis is used as a feature extraction method, and it is extended to a semi-supervised method by incorporating unlabeled data. This extended method is evaluated with respect to the tuning of parameters and general performance for an increasing number of labels.

A.1. Biased Discriminant Analysis

Biased discriminant analysis (BDA) was first introduced in the context of relevance feedback (RF) for content-based image retrieval [31]. In this setting, there is a positive class that shares some common theme, and a negative class consisting of images that do not contain this theme. It is a $(1+k)$-class problem, where k is an unknown number of classes grouped under one label.

Like Fisher's Discriminant Analysis (FDA), BDA finds a linear projection of the high-dimensional features space that is supervised and separates two classes of instances by looking at the intra- and inter-class scatter matrices. BDA is unique in the sense that it does not attempt to describe the negative class, but only distances the negative instances from the positive class. The assumption is that the negative instances do not have a common theme beyond the fact that they are not positive.

For BDA and for the rest of this analysis, two scatter matrices are considered. First, the between scatter S_b is the scatter between the two classes $(+,-)$ with number of samples n_+ and n_-:

$$S_b = \frac{\alpha}{n_+ n_-} \sum_{y_i \neq y_j} (x_i - x_j)(x_i - x_j)^T,$$

where y_i and y_j are class labels and x_i and x_j are the feature vectors. Furthermore, the within scatter matrix of the positive class, which is the dominant class, is considered:

$$S_{w+} = \frac{\beta}{n_+^2} \sum_{y_i = y_j = +} (x_i - x_j)(x_i - x_j)^T.$$

The optimal BDA projection T is then achieved by minimizing S_{w+} and maximizing S_b, formulated as a the following optimization:

$$T_{OPT} = \arg\max_T \left[\frac{tr(T^\top S_b T)}{tr(T^\top S_{w+} T)} \right],$$
$$s.t. T^\top T = I,$$

where I is the identity matrix. In contrast to the FDA projection, which can only be used to extract a one-dimensional projection, the BDA projection has an effective dimension of $\min(n_+, n_-)$ [31].

A.1.1. Generalized BDA

Like Fisher's Discriminant Analysis (FDA), BDA works under the strong assumption of gaussian distribution, albeit only for the positive class. When the number of positive samples is low compared to the dimensionality, which is often the case in image retrieval, BDA also suffers from a singular positive within-scatter matrix so that it requires regularization. Zhang et al. [30] extended the original BDA method to deal with both of these drawbacks. First of all, to avoid the singular problem, the optimization criterion was changed to a difference rather than a fraction of between and positive within scatter.

Secondly, a localized approach was adopted to allow for non-linearity of the sample distribution. This adaptation is in line with the concept of local FDA introduced by Sugiyama [25]. It considers only the local scatter matrix where locality is defined in the high-dimensional space, in a nearest-neighbor sense. Each sample has k nearest neighbors and in the scatter matrices, only those distances are included that are between nearest neighbors.

Finally a regularization term of locality preserving projection (LPP) was included that aims to retain local structure in general, which is a local scatter matrix including unlabeled instances.

A.1.2. Application to system failures

The RF problem is similar to the problem setting described in this research. It assumes a user in the loop who gives feedback in the form of labels, and there is an expected class imbalance towards positive examples. However, the assumption of a $(l+k)$-problem, where the negative class is highly diverse, is not necessarily true for this analysis.

A.2. Method

Both the original inverse formulation of BDA and the adaptation to a difference as suggested by Zhang et al. [30] have been evaluated. In order to make use of unlabeled data, a term is included that maximizes scatter within the unlabeled data. This is analogous to the unsupervised PCA projection.

A.2.1. Difference model

Here, S_b is the scatter between the two classes, S_{w+} is the scatter within the positive class and S_u is the scatter within the unlabeled data. The objective is to maximize S_b, minimize S_{w+} and S_u can be considered a regularization term.

$$T_{OPT} = \arg\max_T [tr(T^\top B T)]$$
$$s.t. T^\top T = I$$

$$B = \alpha S_b - \beta S_{w+} + \gamma S_u \tag{A.1}$$
$$= \frac{\alpha}{n_+ n_-} \sum_{y_i \neq y_j} (x_i - x_j)(x_i - x_j)^T - \frac{\beta}{n_+^2} \sum_{y_i = y_j = +} (x_i - x_j)(x_i - x_j)^T$$
$$+ \frac{\gamma}{n_u^2} \sum_{y_i = y_j = 0} (x_i - x_j)(x_i - x_j)^T$$

The parameters α, β and γ can be tuned to define the type of mapping being learned. For experimental purposes, each parameter can be set to zero. The solution is found by solving the eigenvalue problem

$$B\psi = \lambda\psi$$

where the eigenvectors ψ corresponding to the largest eigenvalues form T_{OPT}.

A.2.2. Inverse model

The inverse model is formulated with only two free parameters α and γ, as scaling S_{w+} would result in a total scaling factor. As the method suffers from a singular S_{w+}, a regularization term needs to be added of ϵI, which is set to $\epsilon = 0.001$ in all experiments.

$$T_{OPT} = \arg\max_{T}[tr(T^\top BT)(T^\top CT)^{-1}]$$

$$B = \alpha S_b + \gamma S_u \tag{A.2}$$

$$C = \beta S_{w+} + \epsilon I \tag{A.3}$$

Due to the regularization term, each of the parameters α, β and γ can be tuned and set to zero. The solution T_{OPT} is computed by solving the eigenvalue problem, analogous to the difference model.

A.3. Datasets

To analyse methods related to these settings in a controlled manner, the well known NIST handwritten digits dataset [12] was adapted to mimic the scenario of an imbalanced binary dataset with a complex structure.

This dataset consist of 8x8 pixel images of digits 0-9. It was adapted in the following way:

Two digits ('2' and '3') are sampled 250 times each to form the positive class and the rest of the digits make up the negative class with 10 samples each. Of these samples, a fraction p is labeled and the rest is provided as unlabeled examples.

A.4. Evaluation

The goal of this research is to design and evaluate a visualization, not a classifier. However, to quantify the effectiveness of a visualization, the assumption is made that the effectiveness of the visualization is directly related to the performance of a classifier trained in the 2D space. The classifier that is used must be simple, but it need not be linear. For the evaluation the parzen classifier is used, which is based on the local sample density. The amount of labels available to the projection should not influence the performance of the classifier, so the same (larger) number of labels is always available to the Parzen classifier. The classifier performance is measured by the area under the ROC, as this is a robust performance measure for problems with class imbalance. The complete experiment and evaluation is done in the following way:

1. Select a fraction p of the original dataset.
2. Compute the optimal 2D projection T using the selected data.
3. Project the full dataset using T.
4. Split the projected data in two equal sets: training and test.
5. Use the training set to train a parzen classifier.
6. Use the test set to evaluate the classifier, measuring the area under the ROC.

A.5. Parameter optimization

Both methods have free parameters to set the relative importance of the three scatter matrices S_{w+}, S_b and S_u. As literature offers no guidance in how to set these parameters, they will be tuned experimentally in this section.

A.5.1. Difference model parameters

To evaluate the influence of each term in (A.1), the projection was evaluated over all combinations of parameters for values from 0 to 100 on a logarithmic scale. The results are shown in figure A.1. From these figures it is clear that S_b (weighted by α) is essential for good performance. Removing either S_{w+} or the unlabeled data has little effect on the performance. Best results are achieved when $\beta = 10\alpha$, though it quickly drops when β becomes even larger. When the unlabeled data receives a larger weight than the labeled terms, performance drops towards the minimal value of 0.5.

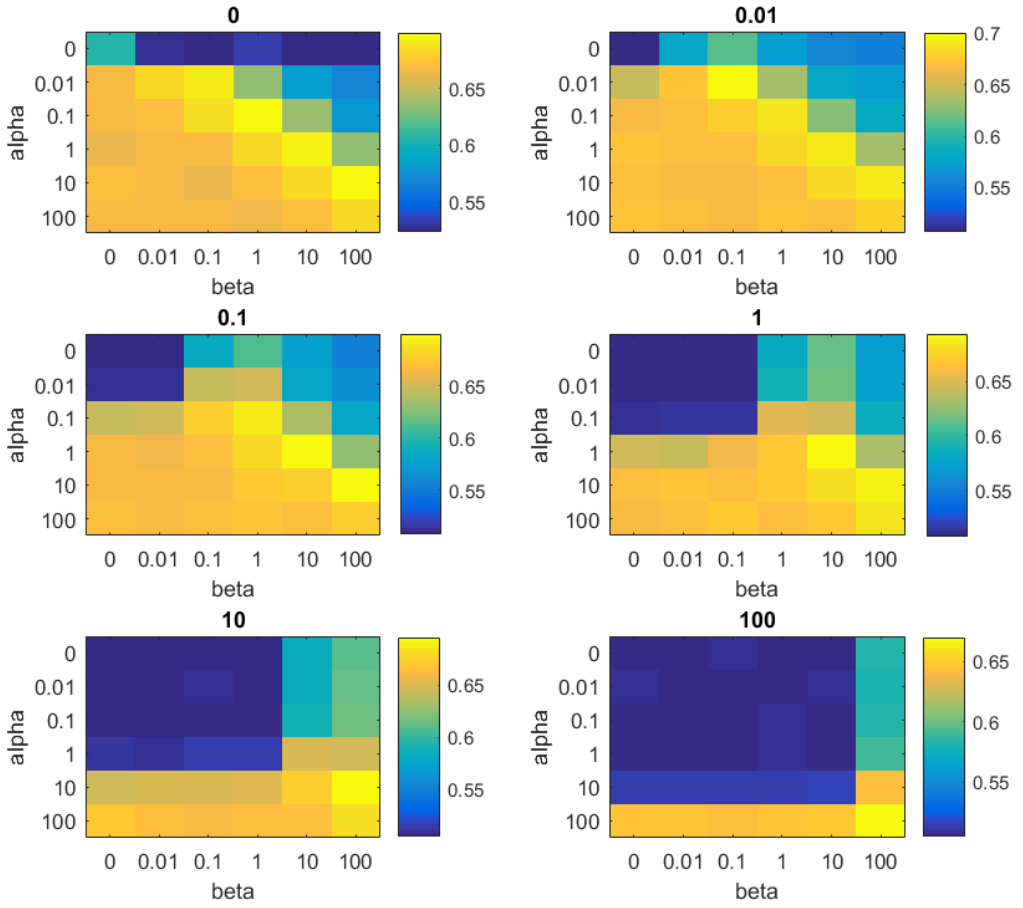

Figure A.1: Performance as surfaces in (α, β)-space, for different values of γ (0 - 100). Performance averaged over 30 samples of 580 images, with a factor $p = 0.1$ labeled. Images '2' and '3' make up the positive class (500 samples) and the rest is negative (80 samples).

A.5.2. Inverse model parameters

In the same way, the parameters α and γ in A.2 were evaluated (figure A.2). Here we can see again that the performance drops to nearly 0.5 when unlabeled data is dominating. As long as $\alpha > \gamma$, the performance is quite constant and similar to the performance of the difference model. Removing unlabeled data completely has no significant effect on the performance.

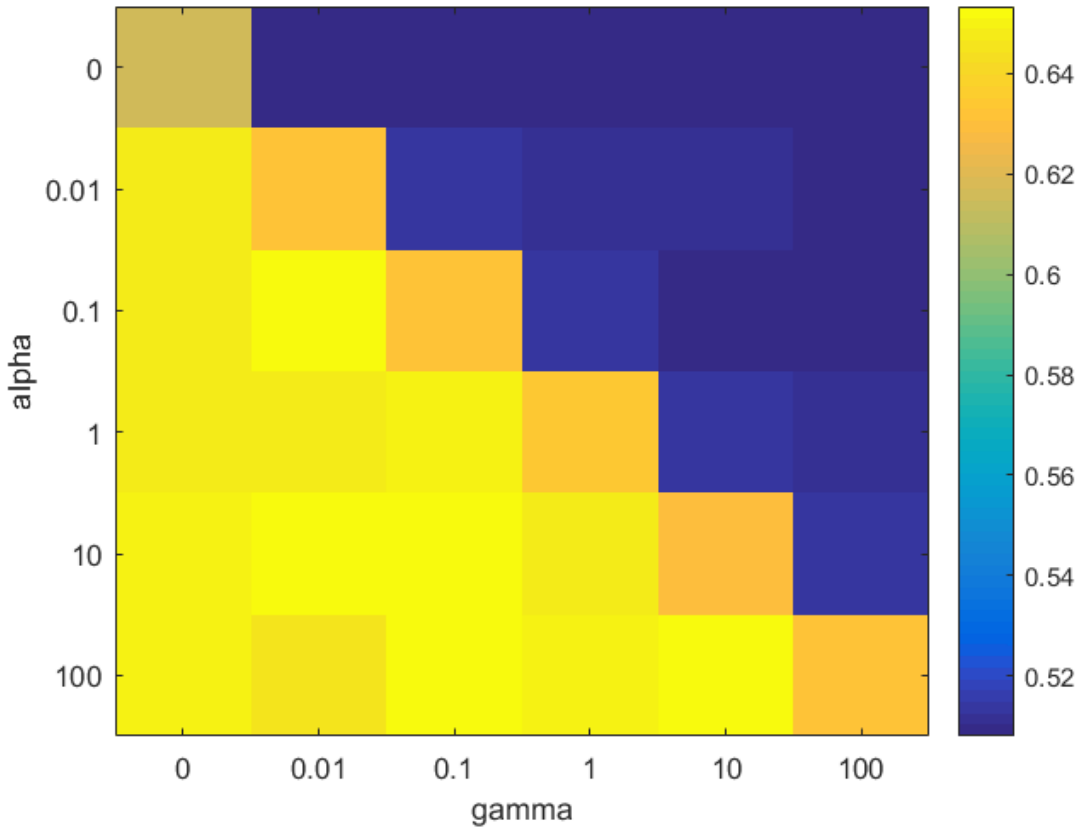

Figure A.2: Performance as surfaces in (γ, α)-space. Performance averaged over 30 samples of 580 images, with a factor $p = 0.1$ labeled. Digits '2' and '3' make up the positive class (500 samples) and the remaining negative (80 samples).

A.5.3. Negative samples model

As experiments show that the between-class scatter is a much more essential term than the within-class scatter of the positive class, the influence of negative within-class scatter was investigated as well, by maximizing:

$$B = \alpha S_b + +\beta S_{w-} + \gamma S_u \tag{A.4}$$

This time, β and γ were varied to negative values as well as positive values. The results are shown in figure A.3. Note that this method achieves the highest performance so far with an AUC of more than 0.75. Also, the performance for $\alpha = \beta = \gamma = 0$, which corresponds to the first two dimensions of the original data, is already 0.61. The highest performance is consistently achieved with $\alpha = -\beta = -\gamma > 0$. β may be even smaller than $-\alpha$ to still achieve good results, while γ can be anywhere between $-\alpha$ and 0.

The effect of S_{w-} is striking since it is a covariance term that consists of only 8 samples in this experiment. Merely the addition of S_b is apparently enough to prevent overfitting.

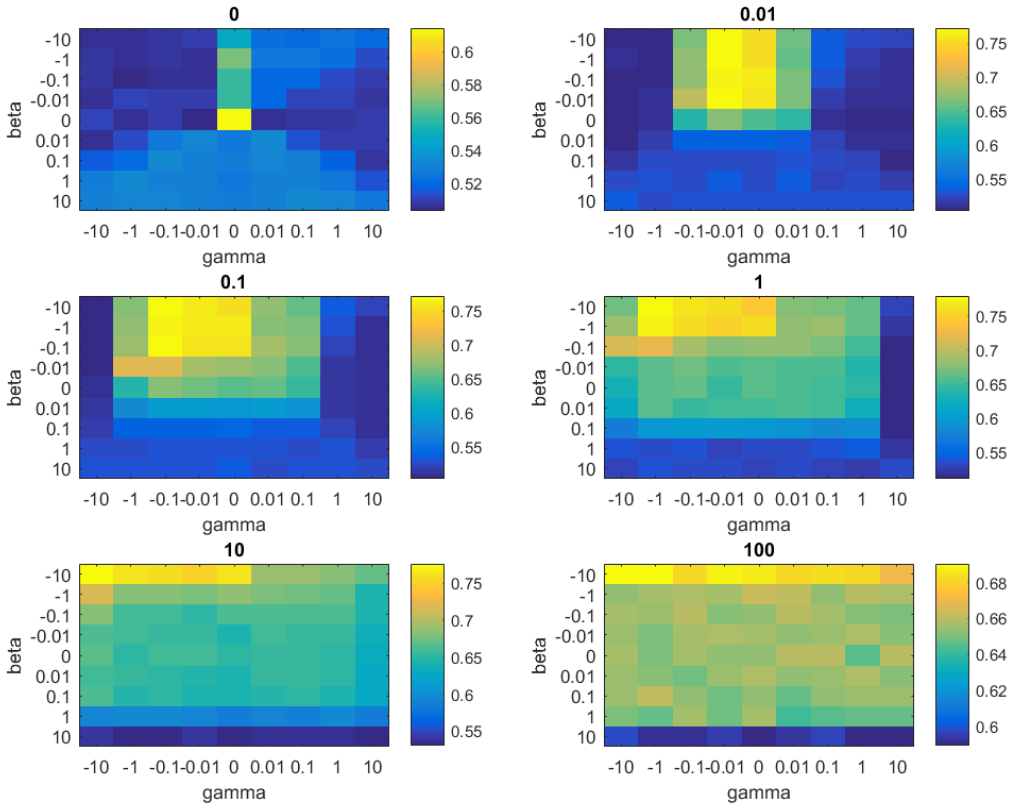

Figure A.3: Performance as surfaces in (γ, β)-space. Performance averaged over 30 samples of 580 images, with a factor $p = 0.1$ labeled. Digits '2' and '3' make up the positive class (500 samples) and the remaining are negative (80 samples).

A.6. Comparison of models

Each of these models were evaluated with $p = 0.1$. Although this is a realistic scenario, with 58 labels, it is essential for the active learning problem to evaluate the methods with different values of p. Taking the parameters with near-optimal performance from previous experiments, figure A.4 shows the performance versus amount of labels on a total of 580 samples. The figure includes the performance of PCA as a reference.

The model that minimizes the negative within-scatter actually decreases in average performance when the amount of labels increases. It also has a very large standard deviation, suggesting that it may profit occasionally from overfitting when there are few negative samples. This theory is supported by the fact that it only has a single positive eigenvalue at all times. The second eigenvalue can even become negative when more than 150 samples are available. With two negative terms, this is not surprising.

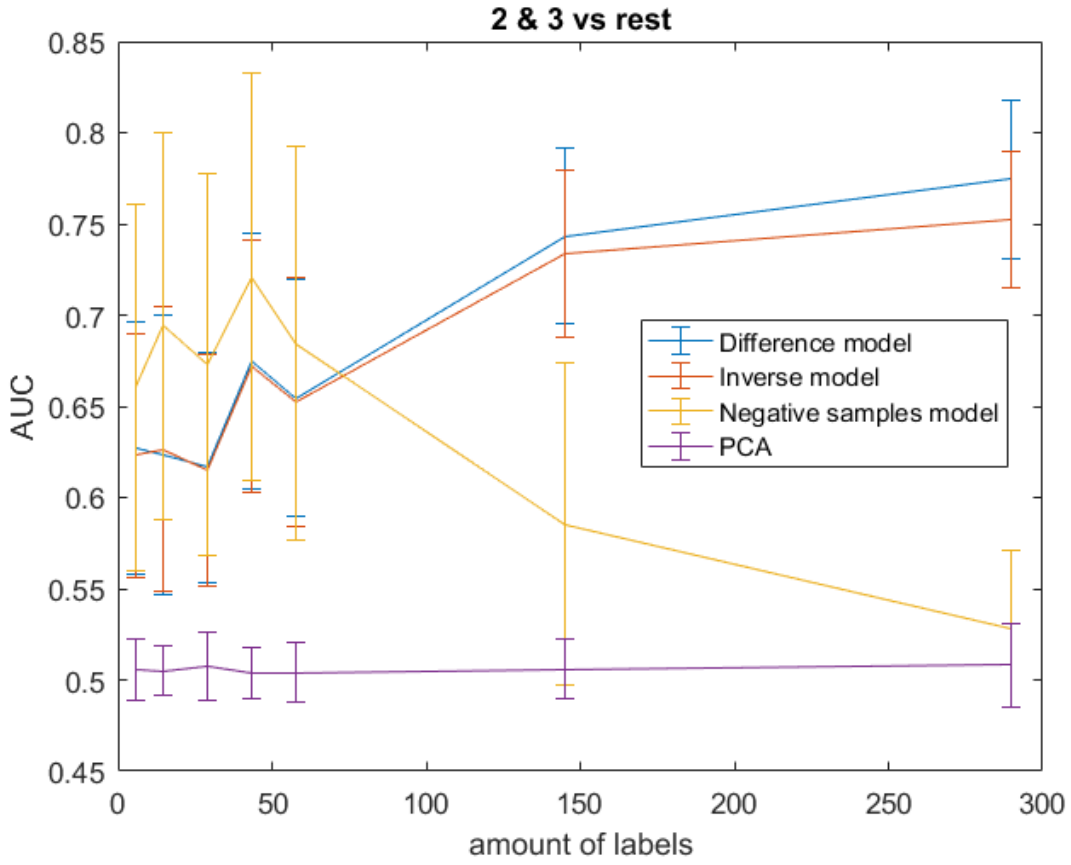

Figure A.4: Performance for increasing amount of labels on 580 instances, for different models. Mean and standard deviation over 100 randomly sampled iterations.

A.7. Conclusion and discussion

With the goal of visualizing a high-dimensional and imbalanced dataset, different variations on Biased Discriminant Analysis are evaluated. With the addition of a PCA term for unlabeled samples, these methods have free parameters to set the relative importance of the terms. An empirical evaluation gives some insight in the effects of setting these parameters for the quality of the projection, measured by means of the performance of a classifier in the 2D space.

When looking at performance in general, there is not much difference between the difference and inverse formulation of the extended BDA model. The supervised nature of the model makes it more effective than PCA in this setting. No comparison with the closely related Fisher's Discriminant Analysis was made, though it must be noted that the FDA projection will only yield one effective dimension, which makes it less suitable for visualization.

No data is currently available to evaluate the visualization for the intended problem, related to rare problems in semi-conductor manufacturing. Originally, BDA was designed for a problem where the negative class is much more diverse than the positive class. In the experiments with the NIST digits this diversity was introduced by re-sampling the original classes, but the situation may be completely different when another more representative dataset is used.