# Visualization of Regression Models using Discriminative Dimensionality Reduction

Alexander Schulz and Barbara Hammer

Bielefeld University - CITEC centre of excellence, Germany
{aschulz|bhammer}@techfak.uni-bielefeld.de

Preprint of the publication [14], as provided by the authors.

**Abstract**

Although regression models offer a standard tool in machine learning, there exist barely possibilities to inspect a trained model which go beyond plotting the prediction against single features. In this contribution, we propose a general framework to visualize a trained regression model together with the training data in two dimensions. For this purpose, we rely on modern nonlinear dimensionality reduction (DR) techniques. In addition, we argue that discriminative DR techniques are particularly useful for the visualization of regression models since they can guide the projection to be more sensitive for those aspects in the data which are important for prediction. Given a data set, our framework can be utilized to visually inspect any trained regression model.

## 1 Introduction

The increasing complexity of data as concerns their dimensionality, size and form constitutes a major challenge on the task of automated data analyses. In many scenarios, it is not possible to formalize an analysis task in advance and, hence, interactive data analysis is required. In this scenario, humans interactively specify learning goals and the appropriate tools [21, 9, 16, 12] in order to interpret heterogeneous and high-dimensional data sets. In this context, data visualization and model interpretability become increasingly important. A trained regression model is not judged by its prediction error only, rather, other questions come into focus, such as: what are particularly difficult regions in the data space for the model, which instances seem to be noisy to the model or which regions of the data space are too sparsely represented by data?

Data visualization is a particular useful tool, since it presents relations among many data points in a well comprehensible way for humans. This field constitutes a well-investigated research topic with many different proposed visualization techniques in the machine learning context. Besides classical methods such as linear mappings computed by principal component analysis or linear

discriminant analysis and nonlinear extensions such as the self-organizing map (SOM) or generative topographic mapping (GTM), a variety of (often nonparametric) dimensionality reduction (DR) techniques has been proposed in the last decade, such as t-distributed stochastic neighbor embedding (t-SNE), neighborhood retrieval visualizer (NeRV), or maximum variance unfolding (MVU), see e.g. the articles [18, 10, 7, 17, 20] for overviews on DR techniques. These approaches are often utilized to visualize data in two dimensions. However, they cannot be directly applied to additionally project the prediction function of a trained regression model. Such a visualization could provide further insights into the regression problem: is the model particularly complex in certain regions of the data space, is it too simplistic in others, how are noisy regions and outliers treated, how does the model extrapolate, and so on.

Besides approaches to judge the quality of trained regression models with quantitative estimates [4], there exists only little work which aims to visualize the regression function itself. For the special case of Decision Trees, a direct inspection is possible through the special tree structure of the model. However, these models can get unclear with increasing size and data complexity. More general approaches such as Breheny and Burchett [2] try to analyze the relationship between the target and a single explanatory variable by visualizing the predictions of the model for different values of this variable while keeping the others fix. However, this approach treats the explenatory variables independently (or a small subset simultaneously) and thus cannot find information that is present in many dependent features.

Our proposed approach, conversely, aims to visualize the whole data set together with the model in one plot, such treating all features simultaneously. Our contribution is based on ideas from a similar approach which was designed recently in our group to visualize classification methods [13]. We adapt these ideas such that they are applicable to the visualization of regression models.

Given a trained regression model, we identify typical user tasks which can be addressed with our framework. These include the questions:

1. How complex is the learned function? Does it overfit/underfit some regions?

2. Is the data multi-modal, i.e. are clusters present in the data and how does the regression model deal with those? What is the prediction for the regions in-between the clusters?

3. Are specific aspects of the selected model visible (such as local linear functions) and are these suited for the data at hand.

4. Are there potential outliers in the data and how does the model treat these?

We will exemplarily show how these questions can be addressed in the experiments section.

The remaining of the paper is structured as follows: First, we discuss popular dimensionality reduction techniques with certain properties which are important

for our proposed framework. Section 3 presents our main contribution, the general framework to visualize regression models. Subsequently, we present the experiments where we exemplarily address the user tasks and argue that discriminative DR is particularly suited for this purpose. Finally, section 5 gives a short discussion.

# 2 Dimensionality Reduction

Dimensionality reduction (DR) mappings try to find low-dimensional embeddings $\pi(\mathbf{x}) = \boldsymbol{\xi} \in \Xi = \mathbb{R}^2$ for given high-dimensional data points $\mathbf{x} \in X = \mathbb{R}^d$ while preserving as much information as possible. The formalization of the latter goal, however, yields many different approaches [18, 10, 7, 17, 20].

Having such a dimensionality reduction mapping $\pi$, some approaches also provide an inverse mapping $\pi^{-1} : \mathbb{R}^2 \mapsto \mathbb{R}^d$. This is in particular the case for parametric DR techniques. Since we will need such a mapping, we will discuss how to compute it if it is not provided by the DR method in subsection 2.2.

First, we give a short description of popular DR approaches which we utilize.

- The goal of *Multidimensional scaling (MDS)* is to embed the data such that the distances in $X$ and in $\Xi$ agree. If these distances are Euclidean, MDS is equivalent to PCA. However, other metrics can be integrated directly.
- One popular nonlinear alternative is the parametric *generative topographic mapping (GTM)* [1]. Essentially, GTM relies on data being generated by a constraint mixture of Gaussians. The centers of the Gaussians are generated by a smooth mapping from regular lattice positions in a two-dimensional latent space which can be used for data visualization. GTM is optimized by a maximization of the data log likelihood function. The GTM provides a smooth mapping of the data to its low-dimensional projection $\pi(\mathbf{x})$ and vice versa $\pi^{-1}(\boldsymbol{\xi})$. Different metrics can be integrated [8].
- *T-distributed stochastic neighbor embedding (t-SNE)* [17] is a nonparametric approach and defines local neighborhoods in a probabilistic sense by using Gaussians based on pairwise distances in the feature space and student-t distributions induced by euclidean distances in the projection space. Training takes place by a minimization of the error in between these distributions as measured by the Kullback Leibler divergence.

## 2.1 Discriminative dimensionality reduction with the Fisher metric

Dimensionality reduction in general is an ill-defined problem. This is particularly critical if the data is intrinsically high-dimensional and hence not embeddable in two dimensions. Then, the DR methods have to make compromises

and omit information. This choice which information to omit is often arbitrary or even depends on random aspects.

Hence, the class of discriminative DR approaches has been proposed[1]. These methods suggest to use auxiliary information to guide the DR method. These can be data labels or the values of the target variable, $y$ in our case.

A particular successful and general approach is to use the Fisher information metric as a basis for the DR techniques. The general idea is to define a Riemannian manifold which takes the auxiliary information of the data into account. This modified metric can then be plugged into any DR technique which relies on distances only. This idea has been applied for classification problems very successfully [11, 13], and very recently for the case of regression problems [15].

The distance from a point $\mathbf{x}$ to $\mathbf{x}'$ on the Riemannian manifold can be computed by finding the shortest path

$$d_M(\mathbf{x}, \mathbf{x}') = \inf_P \int_0^1 \sqrt{P'(t)^\top \mathbf{J}(P(t))P'(t)}dt \tag{1}$$

where the infimum is over all differentiable paths $P : [0, 1] \to X$ with $P(0) = \mathbf{x}$ and $P(1) = \mathbf{x}'$. Here, local distances are defined using the Fisher information matrix

$$\mathbf{J}(\mathbf{x}) = \mathbb{E}_{p(y|\mathbf{x})} \left\{ \left( \frac{\partial}{\partial \mathbf{x}} \log p(y|\mathbf{x}) \right) \left( \frac{\partial}{\partial \mathbf{x}} \log p(y|\mathbf{x}) \right)^\top \right\} \tag{2}$$

at the position $\mathbf{x}$.

In [15], a Gaussian Process is used to estimate $p(y|\mathbf{x})$. Further, approximations for the path integrals are investigated in [11]. A good compromise between performance and quality is to restrict arbitrary paths on the Riemannian manifold to a straight line and to approximate the integral (1) by T piecewise constant terms. More formally, assume $\mathbf{x}_t = \mathbf{x} + (t-1)/T \cdot (\mathbf{x}' - \mathbf{x})$. Then the distance on the manifold $d_M$ can be approximated by

$$d_T(\mathbf{x}, \mathbf{x}') = \sum_{t=1}^{T} \sqrt{(\mathbf{x}_t - \mathbf{x}_{t+1})^\top J(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{x}_{t+1})}. \tag{3}$$

## 2.2 Inverse dimensionality reduction

We have seen in a previous part of this section, that parametric DR methods often provide an approximate inverse DR mapping $\pi^{-1}$, which is not the case for nonparametric methods.

In this subsection, we repeat the ideas from [13] to define such a mapping for an arbitrary DR approach.

We assume that points $\mathbf{x}_i \in X$ and projections $\pi(\mathbf{x}_i) = \boldsymbol{\xi}_i \in \Xi$ are available. For an inverse projection, we assume the following functional form

$$\pi^{-1} : \Xi \to X, \boldsymbol{\xi} \mapsto \frac{\sum_j \boldsymbol{\beta}_j k_j(\boldsymbol{\xi}, \boldsymbol{\xi}_j)}{\sum_l k_l(\boldsymbol{\xi}, \boldsymbol{\xi}_l)} \tag{4}$$

[1] We use the terms *discriminative* and *supervised* as synonyms, in the following.

where $\boldsymbol{\beta}_j \in X$ are parameters of the mapping and $k_j(\boldsymbol{\xi}, \boldsymbol{\xi}_j) = \exp(-0.5\|\boldsymbol{\xi} - \boldsymbol{\xi}_j\|^2/(\sigma_j^{\boldsymbol{\xi}})^2)$ constitutes a Gaussian kernel with bandwidth determined by $\sigma_j^{\boldsymbol{\xi}}$. Summation is over a random subset $\Xi'$ of the given data projections $\boldsymbol{\xi}_i = \pi(\mathbf{x}_i)$, or over codebooks resulting from a previously run vector quantization on the $\boldsymbol{\xi}_i$.

Formalizing a valid cost function to optimize the parameters of $\pi^{-1}$ constitutes a challenge, if the intrinsic data dimensionality is larger then 2. In this case, the inverse position of a given projection $\boldsymbol{\xi}$ can be ambiguous. In order to emphasize those directions in the data space that are relevant for the target variable we utilize the distance as measured with the Fisher metric in the cost function:

$$E = \sum_i \left( d_1 \left( \mathbf{x}_i, \pi^{-1}(\boldsymbol{\xi}_i) \right)^2 \right) = \sum_i \left( \mathbf{x}_i - \pi^{-1}(\boldsymbol{\xi}_i) \right)^\top \mathbf{J}(\mathbf{x}_i) \left( \mathbf{x}_i - \pi^{-1}(\boldsymbol{\xi}_i) \right) \quad (5)$$

We utilize the distance $d_T$ with $T = 1$ in order to save computational time. This local approximation works usually well since in the course of optimization the points $\mathbf{x}_i$ and $\pi^{-1}(\boldsymbol{\xi}_i)$ will get close to each other. Minimization of these costs with respect to the parameters $\boldsymbol{\beta}_j$ takes place by gradient descent.

# 3 General Framework for Visualizing Regression Models

In this section, we are in the position to put the pieces together towards a general framework for the visualization of regression models. We assume the following scenario: a data set including points $\mathbf{x}_i \in X$ is given. Every data point is accompanied with a target value $y_i \in \mathbb{R}$. In addition, a regression model $f : X \to \mathbb{R}$ has been trained on the given training set, such as a support vector machine for regression (SVR). A visualization of the given data set and the regression model would offer the possibility to visually inspect the prediction result and to address user tasks as formulated in section 1. We propose a general framework how to visualize a regression model and a given data set.

## 3.1 Naive approach

Assuming a nonlinear dimensionality reduction method is given, a naive approach to visualize a regression model could be like follows:

- Sample the full data space $X$ by points $\mathbf{z}_i$.
- Project these points nonlinearly to two-dimensional points $\pi(\mathbf{z}_i)$ using some nonlinear dimensionality reduction technique.
- Display the data points $\pi(\mathbf{x}_i)$ and the contours induced by the sampled function $(\pi(\mathbf{z}_i), f(\mathbf{z}_i))$, the latter approximating the prediction of the model.

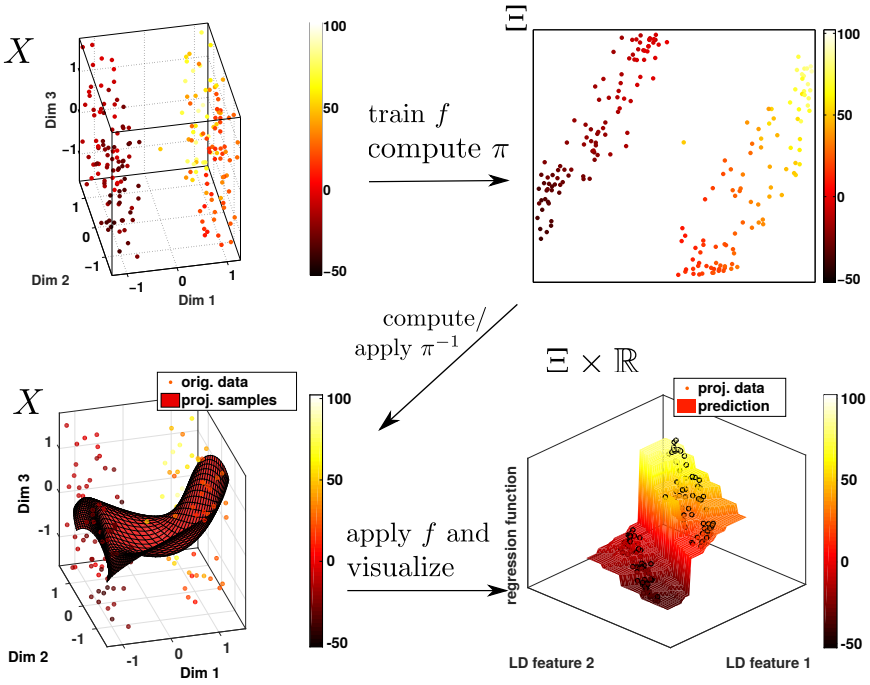


Figure 1: Illustration of our proposed approach to visualize a regression model (in this case a Decision Tree ).

This simple method, however, fails unless $X$ is low-dimensional because of two reasons:

- Sampling $X$ sufficiently requires an exponential number of points, hence it is infeasible for high-dimensional $X$.
- It is impossible to map a full high-dimensional data set faithfully to low dimensions, hence topological distortions are unavoidable when projecting the prediction function.

The problem is that this procedure tries to visualize the function in the full data space $X$. It would be sufficient to visualize only those parts of the function which are relevant for the given training data and the given prediction function.

## 3.2 Our proposed approach

Hence, we propose to sample in the projection plane instead of the original data manifold, and we propose to use a discriminative DR technique to make the problem of data projection well-posed in the sense that discriminative methods define clearly what structure preservation means (see [15] for a discussion on this). Together with the techniques presented in the last section, this leads to the following feasible procedure for the visualization of regression models:

- Project the data $\mathbf{x}_i$ using a nonlinear discriminative DR technique leading to points $\pi(\mathbf{x}_i) \in \Xi$.
- If not provided by the selected DR technique, compute an inverse mapping $\pi^{-1}$ by optimization of equation (5).
- Utilize the mapping $f \circ \pi^{-1}(\mathbf{z})$ to visualize the regression function on any position $\mathbf{z} \in \Xi$ in the low-dimensional space.

In order to execute the last step, we sample the projection space $\Xi$ in a regular grid leading to points $\{\mathbf{z}_i\}_{i=1}^n$. Finally, we visualize the pairs $(\mathbf{z}_i, f \circ \pi^{-1}(\mathbf{z}_i))$ as contours of a two-dimensional plot, or plotting $f \circ \pi^{-1}(\mathbf{z}_i)$ over the third axis as done in Fig. 1. An illustration of this approach is depicted in Fig. 1 with a three-dimensional data set, where dimension 3 is irrelevant for prediction (see section 4 for further details of this data set). The Fisher GTM has been utilized to obtain $\pi$ and $\pi^{-1}$ in this example.

Unlike the naive approach, sampling takes place in $\mathbb{R}^2$ only and, thus, is feasible. Further, only those parts of the space $X$ are considered along which the regression function changes. Such, directions irrelevant for the target mapping are neglected.

### 3.3 Evaluation measure

In order to evaluate the quality of the obtained visualization of the regression model, we propose to utilize the Pearson correlation of $f(\mathbf{x})$ and $f \circ \pi^{-1} \circ \pi(\mathbf{x})$:

$$\frac{\mathbb{E}\left\{\left(f(\mathbf{x}) - \mathbb{E}\left(f(\mathbf{x})\right)\right) \cdot \left(f \circ \pi^{-1} \circ \pi(\mathbf{x}) - \mathbb{E}\left(f \circ \pi^{-1} \circ \pi(\mathbf{x})\right)\right)\right\}}{\sqrt{\mathbb{E}\left\{\left(f(\mathbf{x}) - \mathbb{E}\left(f(\mathbf{x})\right)\right)^2\right\}} \cdot \sqrt{\mathbb{E}\left\{\left(f \circ \pi^{-1} \circ \pi(\mathbf{x}) - \mathbb{E}\left(f \circ \pi^{-1} \circ \pi(\mathbf{x})\right)\right)^2\right\}}} \quad (6)$$

This criterion does not measure in how far $\pi^{-1}$ is the exact inverse of $\pi$. Obtaining such an inverse mapping would be impossible for the most data sets. Instead, equation 6 evaluates the precision of $\pi^{-1}$ with respect to $f$, i.e. errors along directions where $f$ doesn't chage are not accounted as such. This way, only directions in the data space are considered which are relevant for the prediction.

This procedure estimates the quality of the visualization of the regression model at the positions of the data points. Other regions are not evaluated with this approach. We prefer the Pearson correlation over the normalized MSE, because the former is always normalized between -1 and 1.

For the computation, we utilize only those points $\mathbf{x}$ which were not utilized to train the mapping $\pi^{-1}$. Further, we approximate $f \circ \pi^{-1} \circ \pi(\mathbf{x})$ by the prediction value of the closest sampled point $\mathbf{z}'$ of $\pi(\mathbf{x})$, simply because we have already computed $f \circ \pi^{-1}$ for these points.

## 4 Experiments

In this section we demonstrate our proposed approach with artificial and real life data sets. We employ the popular Support Vector Machine for regression

and the Decision Tree scheme as the models that we interpret. Furthermore, since we do not assume any particular property of the regression model, any regression scheme could be visualized in the same way. A description of the models follows.

- The Support Vector Machine for regression (SVR) [19] employs a linear function $f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}$ in the feature space for prediction. Errors are penalized linearly, where small errors, i.e. predictions lying in an $\epsilon$-tube around the target, are not penalized. Since the whole approach can be formulated using scalar products of the data only, kernels can be employed. In the experiments, we utilize the Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\beta \|\mathbf{x}_i - \mathbf{x}_i\|^2\right)$. We use the implementation provided by the libsvm [5].

- Decision Trees (DecTree) [3] for regression partition the data space X, where the prediction value in each partition is the mean of the points lying in the according partition. Splits are optimized such that the mean squared error is minimized. We utilize the Matlab implementation here.

In the following, we demonstrate how the user tasks described in section 1 can be tackled with of our proposed approach, what effects the choice of the employed dimensionality reduction can have and we apply our presented approach to a real world data set. In the following, we briefly characterize the utilized data sets.

- *Data set1* is depicted in Fig. 2 (left) and consist of three two-dimensional clusters positioned above each other. One of these clusters (the bottom one) has additional noise in the third dimension. The prediction function is again encoded in the color and is a squared function of dimension three.

- *Data set2* consists of two three-dimensional clusters with an outlier in-between these two clusters. Fig. 2 (right) depicts this set, where the color indicates the target variable of the regression task which is a linear function for the left cluster and a squared function for the right one. In both cases, the target function depends only on the first two dimensions.

- The *diabetes* [6] data set describes 442 patients by the 10 features age, sex, body mass index, blood pressure and 6 blood serum measurements. The target variable is a measure of the progression of the diabetes disease one year after feature acquisition.

## 4.1 Effect of the selected dimensionality reduction technique

One key ingredient in our proposed approach is the DR. However, since any DR technique can be applied, we discuss in this section effects of the selected methods. For this purpose, we train a SVR model on data set1 and visualize it with different techniques.

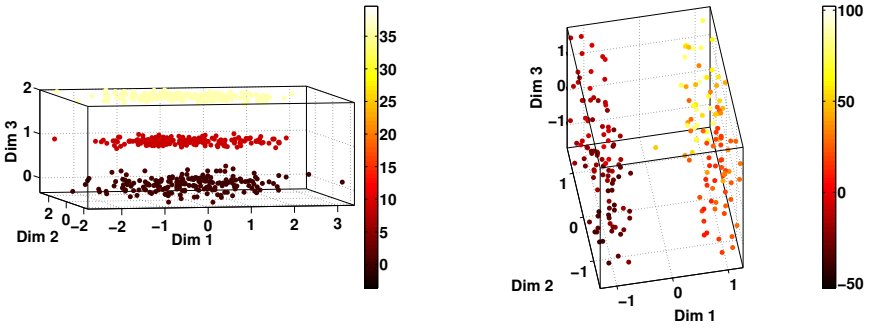


Figure 2: Two toy data sets: data set1 (left) and data set2 (right).

The most common visualization approach is PCA. However, the latter is driven only by the variance of the data and neglects other structure. Hence, using PCA for data set1 yields to overlapping clusters and hence to a bad visualization of the underlying regression model: the accordance as evaluated by (6) is 0.21, i.e. the visualized model has only a small correlation to the original one.

In a scenario where the structure of the data is not known, more powerful nonlinear DR methods can be necessary. We investigate here the two methods GTM as a generative model and t-SNE as a neighborhood embedding.

Applying our regression model visualization approach to the trained SVR using the GTM yields a visualization with a quality of 0.95 (as summed up in Table 1). The visualized model is depicted in the top left corner of Fig. 3. Although, the accordance of the visualized prediction model with the original one is high, the visualization tears the cluster structure apart. So, more powerful methods for DR can increase the visualization quality. However, there still might be undesired effects, especially if the approaches act in an unsupervised way. An other option, besides choosing more powerful methods, is to utilize supervised ones. This can be done, as discussed earlier with the use of a supervised metric.

To demonstrate the effect of such a supervised visualization, we apply our regression model visualization approach using Fisher MDS, Fisher GTM and Fisher t-SNE. Applying these techniques, we obtain three different visualizations of the same regression model. We evaluate them and obtain a quality of 0.99 for each visualization (summed up in Table 1). The visualizations (in Fig. 3) of Fisher MDS (top right) and Fisher GTM (bottom left) agree largely, while the Fisher GTM based visualization shows the shape of the squared polynomial target function without any distortions. In the Fisher t-SNE projection, the squared prediction for the single clusters can be observed, but it is not so clear as in the Fisher GTM mapping. One reason for this is that t-SNE often tears clusters apart since it has a high focus on local neighborhood preservation.

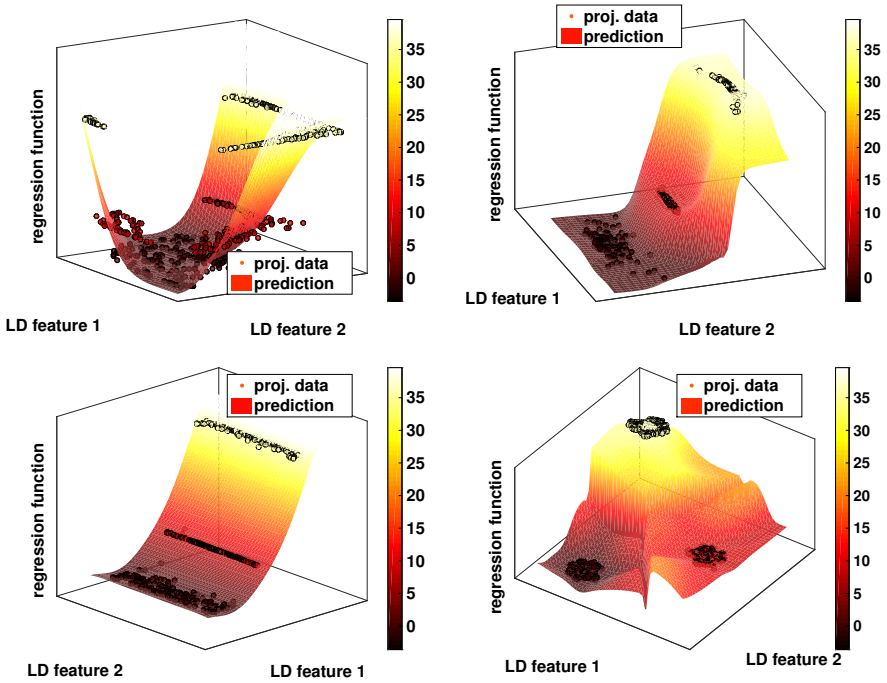


Figure 3: Four different visualizations of the same regression model. These are based on (from top left to bottom right): GTM, Fisher MDS, Fisher GTM, Fisher t-SNE.

## 4.2 Illustration of potential user tasks

We utilize data set2 to illustrate the identified user tasks. For this purpose, we train a SVR and a Decision Tree using this data set. For the DR, we employ the supervised technique Fisher GTM - an unsupervised approach would try to embed this intrinsically three-dimensional data set in two dimensions and, hence, might result in an embedding not well suited to visualize the target function.

Using these ingredients, we can visualize the two regression models with our proposed approach. Employing the numerical evaluation scheme in 3 implies a quality of 0.99 for both visualizations, as measured by the Pearson correlation. I.e. the regression model is shown accurately at least at the positions of the data. The evaluation results for all experiments are summed up in Table 1.

The resulting visualized models are shown in Fig. 4. The left plot depicts the SVR and the right one the Decision Tree. In both cases, the first two coordinate axes encode the two-dimensional embedding space of the data. The target variable is encoded both by the third axis and by the coloring. The surface depicts the prediction of the respective regression model.

We exemplarily address the user tasks for these visualizations. Considering user task 1, the complexity of the prediction functions can be observed directly

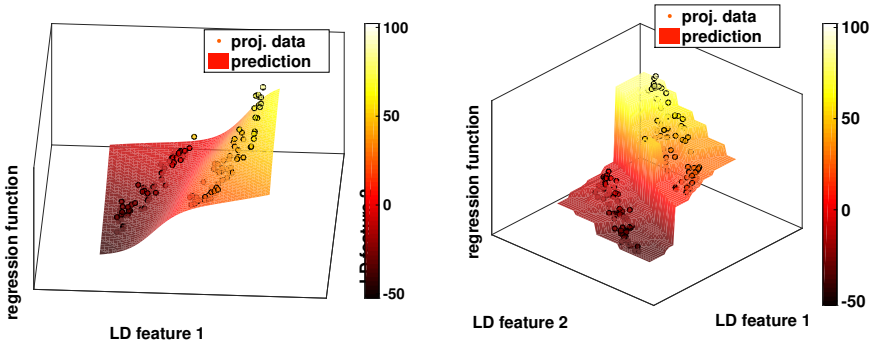


Figure 4: A Fisher GTM induced visualization of the SVR (left) and Decision Tree (right) with data set1. The continuous surfaces depict the prediction of the regression models.

in the visualizations: the SVR instance shows a smooth predictive function while the Decision Tree instance is very complex. This is particularly the case for the cluster with the squared function: the trained SVR might be considered underfitted, here. Dealing with user task 2, the user can observe that the complexities of the target functions are quite different in the two present clusters. The user could prefer to train two independent local models on these clusters, for instance. The extrapolation between these clusters is smooth in the left image but very steep in the right one, which might lead to bad predictions if future data are expected to lie also between the clusters. In the right visualization, the piecewise constant regions are good visible which is typical for Decision Tree models (user task 3). Considering user task 4, the visualizations directly imply how both models treat the outlier point: the SVR ignores it and the Decision Tree overfits it. Having this insight, the user can judge which model handles the data point of interest better, depending on his estimation of the regularity of this point.

## 4.3 Applying the proposed framework to real world data

For the diabetes data set, we train the SVR model by splitting the data set multiple times randomly in a training and a test set in order to estimate a good parameter value for the kernel of the SVR.

Table 1: Visualization qualities for the regression models, as measured by the Pearson correlation.

| | PCA | GTM | Fisher MDS | Fisher GTM | Fisher t-SNE |
|---|---|---|---|---|---|
| data set1, | 0.21 | 0.95 | 0.99 | 0.99 | 0.99 |
| data set2, SVR | – | – | 0.99 | 0.99 | 0.99 |
| data set2, DecTree | – | – | 0.99 | 0.99 | 0.99 |
| diabetes | – | – | – | 0.94 | 0.92 |

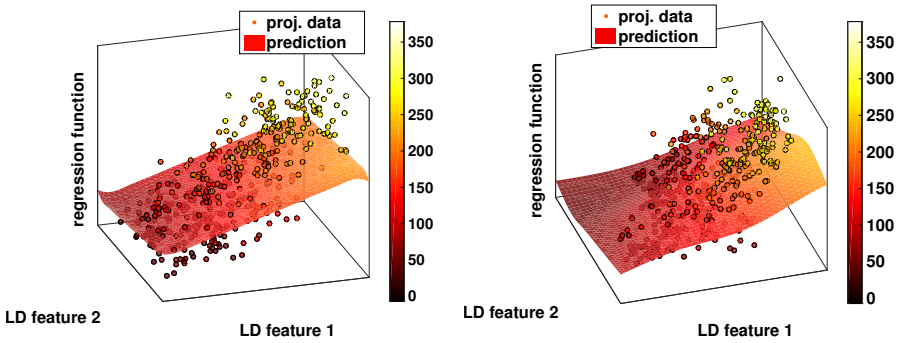


Figure 5: A Fisher GTM (left) and a Fisher t-SNE (right) visualization of a SVR model trained on the diabetes data set.

In the previous subsections we have argued that discriminative nonlinear DR methods are best suited for the visualization of regression models. Hence, we apply two such methods, i.e. Fisher GTM and Fisher t-SNE to the SVR model trained on the diabetes data set.

The evaluation based on (6) yields a quality value of 0.94 for the visualization based on Fisher GTM and a value of 0.92 for the Fisher t-SNE induced visualization. Both are shown in Fig. 5.

Interestingly, both visualizations agree in that sense that they show an almost linear prediction function. We have validated this by training a linear model and have obtained a similar error on the test data.

# 5 Discussion

We have proposed a general framework to visualize a given trained regression model together with the training data. This allows the user to inspect various properties of the trained model.

While the approach is in that sense general, that it allows to use any DR and any regression technique, only the Support Vector Regression and Decision Tree approaches have been utilized so far. Further work will demonstrate this framework on other regression models as well as on more real life data sets.

# References

[1] C. M. Bishop, M. Svensén, and C. K. Williams. Gtm: The generative topographic mapping. *Neural computation*, 10(1):215–234, 1998.

[2] P. Breheny and W. Burchett. Visualization of regression models using visreg, 2013.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

[4] A. C. Cameron and F. A. G. Windmeijer. An r-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77(2):329–342, Apr. 1997.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[7] A. Gisbrecht and B. Hammer. Data visualization by nonlinear dimensionality reduction. *WIREs Data Mining and Knowledge Discovery*, 2014.

[8] A. Gisbrecht, B. Mokbel, and B. Hammer. Relational generative topographic mapping. *Neurocomputing*, 74(9):1359–1371, 2011.

[9] T. W. House. Big data research and development initiative, 2012.

[10] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

[11] J. Peltonen, A. Klami, and S. Kaski. Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, 17:1087–1100, 2004.

[12] P. E. Rauber, R. R. O. d. Silva, S. Feringa, M. E. Celebi, A. X. FalcÃ£o, and A. C. Telea. Interactive Image Feature Selection Aided by Dimensionality Reduction. In E. Bertini and J. C. Roberts, editors, *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2015.

[13] A. Schulz, A. Gisbrecht, and B. Hammer. Using discriminative dimensionality reduction to visualize classifiers. *Neural Processing Letters*, pages 1–28, 2014.

[14] A. Schulz and B. Hammer. *Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015, Proceedings, Part II*, chapter Visualization of Regression Models Using Discriminative Dimensionality Reduction, pages 437–449. Springer International Publishing, Cham, 2015.

[15] A. Schulz and B. Hammer. Discriminative dimensionality reduction for regression problems using the fisher metric. In *Accepted in IJCNN 2015*, 2015.

[16] S. J. Simoff, M. H. Böhlen, and A. Mazeika, editors. *Visual Data Mining - Theory, Techniques and Tools for Visual Analytics*, volume 4404 of *LNCS*. Springer, 2008.

[17] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[18] L. van der Maaten, E. Postma, and H. van den Herik. Dimensionality reduction: A comparative review. Technical report, Tilburg University Technical Report, TiCC-TR 2009-005, 2009.

[19] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[20] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *JMLR-10*, 11:451–490, 2010.

[21] M. Ward, G. Grinstein, and D. A. Keim. *Interactive Data Visualization: Foundations, Techniques, and Application*. A. K. Peters, Ltd, 2010.