The utility of clustering for real estate valuation

Immobilienbewertung mithilfe clustering-gestützer Marktsegmentierung

Kathrin Thenhausen

Universitätsbachelorarbeit
zur Erlangung des akademischen Grades

Bachelor of Science
(B. Sc.)

im Studiengang
IT Systems Engineering
eingereicht am 28. Juni 2022 am
Fachgebiet Algorithm Engineering der
Digital-Engineering-Fakultät
der Universität Potsdam

Gutachter	Prof. Dr. Tobias Friedrich
Betreuer	Vanja Doskoč
	Dr. Sarel Cohen
	Aishwarya Radhakrishnan

Abstract

Given the high amount of data in present times, the accuracy of machine learning algorithms suffers from a rather heterogeneous nature of data. Clustering the samples is one way to obtain more homogeneity within the groups. There is evidence that training models on each of the clusters can improve their accuracy. In this thesis we study cluster-based prediction models for real estate valuation, utilizing the segmentation into submarkets. Therefore, we compare the effects of models trained on similarity-based groups. While clustering does not improve the results for deep neural networks and a case-based reasoning approach optimized via evolutionary algorithms, it has positive effects on simpler algorithms like linear regression models.

When the subsets get too small, we experience significant performance decreases of the single cluster models. Hence, transfer learning is a promising technique to deal with too small groups and benefit from leveraging knowledge from larger domains. The proposed architecture transfers a clustering and optionally its corresponding models from a larger source domain into a smaller target domain.

Zusammenfassung

Mit der derzeit rasant wachsenden Datenmenge nimmt auch die Heterogenität innerhalb der Daten zu, mit der sich Algorithmen des maschinellen Lernens auseinandersetzen müssen. Clustering beschreibt eine Technik, die ähnlichkeitsbasiert Gruppen bildet, deren Daten eine geringere Varianz aufweisen. Der Nutzen des Trainings von einzelnen Modellen auf diesen Clustern, anstelle des ganzen Datensatzes, wurde bereits gezeigt. In dieser Arbeit stellen wir solche cluster-basierten Algorithmen im Rahmen der Immobilienbewertung vor und untersuchen den Effekt der Immobilienmarktsegmentierung auf verschiedene Vorhersagemodelle. Während tiefe neurale Netze und ein von evolutionärem Lernen unterstützer fallbasierter Vorhersagealgorithmus die Zerlegung des Datenbestands weniger gut verarbeiten können, zeigen wir einen positiven Einfluss des Clusterings auf ein simpleres lineares Regressionsmodell. Auf die erstgenannten Ansätze wirken sich vor allem zu kleine Segmente negativ aus.

Transfer Learning ist eine vielversprechende Technik mit geringen Datenbeständen umzugehen, indem Wissen von einer größeren Quelldomäne auf die Zieldomäne übertragen wird. Wir präsentieren hierzu eine Architektur die ein Clustering sowie die entsprechenden Modelle transferriert.

Acknowledgments

First I would like to thank my team for the time of the bachelor's project. Lukas, for your helpfulness and acceptance to share your desk with cooking utensils. Paula, for bringing empathy, sourdough and yoga-based coworking into the daily work. Niklas, for your balance, your programming skills that prevented a lot of mental breakdowns, and your ginger drinks as well. Thanks to Ben who supported me pretty much during the writing of this thesis and his proofreading.

Vanja, Aishwarya and Sarel advised me scientifically, thanks for the great feedback I have got and your interesting ideas.

Contents

1 Introduction

Recent advances in machine learning combined with the explosive growth of big data technologies enables artificial intelligence becoming a paramount part of our daily life. Especially regarding the rapidly evolving real estate market, we wish the decisions any algorithm takes to be as accurate as possible. Though, as the amount of data is increasing fast, models struggle with the treatment of very heterogeneous data.

Breaking the data into smaller, homogeneous subsets, in terms of machine learning referred to as *clustering*, reduces the variance the model has to care about and for that reason may improve the predictions. The central idea of using clustering for prediction tasks is the training of independent models on each beforehand identified cluster and their combination to a more precise prediction model. Hence, we differ the steps of building a *grouping* and forecasting the *target variable*.

For the former one, we consider the *K-Means*-clustering, splitting real estate data into submarkets that are of high importance for valuation [Hay06; WS12]. Secondly, we build *deep neural networks (DNNs)* for each of the calculated clusters, combining them with different ensemble methods as proposed by Trivedi et al. [TPH11].

Training models, especially DNNs, to solve complex problems requires a lot of data. So, the information gathering process is a big topic in the context of deep learning. While there are huge amounts of data available in general, some markets contain fewer data points and splitting reduces the volume once more. The concept of transfer learning overcomes this issue by inferring knowledge from a source domain into a target domain. Classical use cases build a model and re-train it with new data. As we assume house markets to be distributed quite similar, we can use the understanding of a larger real estate source domain for solving tasks in a smaller real estate target domain.

So we build groupings on the target domain upon the knowledge of clustering the source domain and predict prices on the cluster of the target domain. Further, we transfer the pre-trained models on each of the source data clusters to the corresponding groups of the target data and observe positive effects on price prediction tasks regarding the usage of linear regression models. To evaluate the effects of clustering on different machine learning techniques, we compare two models: a prediction model using DNNs that are pretrained on clusters of the source domain as well as a cluster-based variant of a cased-based reasoning approach

optimized via evolutionary algorithms by Angrick et al.[Ang+21]. For the latter one, we restrict the so-called neighbor selection, decisive for price prediction of one sample, to choose only properties within the same cluster.

2 Related work

As the heterogeneous nature of real estate data is a well-known problem, the approach of splitting the market into smaller submarkets gains importance [Mal+18]. The agreement that house markets are naturally segmented into clusters which should be part of the valuation, leads to a high number of studies regarding the segmentation process.

While mainly relying on administrative and geographical defined boundaries does not yield notable prediction improvements [FGM00], Bourassa et al. [Bou+99] successfully proposed a two-step procedure. They extract the relevant factors and dimensions of an Australian real estate data set with *principal component analysis* (PCA) and use it for cluster analysis afterwards. Hedonic equations on K-Means-constructed submarkets give significantly better results than the overall market equation. Despite the critique on K-Means for the necessity of handing in the number of clusters *a priori* [BWV06], Shi et al.'s application of a fuzzy K-Means-clustering, that is a form of clustering in which each data point can belong to several clusters, yields improvements in the subsequent prediction [Shi+15]. In those prediction tasks the usage of multiple regression analysis (MRA) is common regarding real estate, as it provides explainability. In contrary, artificial networks often suffer from their black box character. Shi et al. therefore compare MRA and an adaptive neuro-fuzzy inference system, proposed by Guan et al. [GZL08], that adds interpretability to networks. Other clusterings of real estate include neural networks [KHH02] and geostatistical approaches [Hay06; WS12]. Chen et al. [Che+07] compare a number of clusterings, among others a K-Means-clustering, two-steps clusterings gathering hyperparameters before, and a clustering by high school districts or expert defined submarkets. Their study shows the value of including a priori knowledge, like given in expert or geographical based clusters. In general, the high number of studies regarding house market segmentation shows the importance of those submarkets for valuation, but also the lack of an ideal model.

For the prediction task we consider the work of Trivedi et al. [TPH11], training separate models on each of the clusters, which is suggested by Straszheim [Str74] as well. Further studies on the step of predicting upon the clusters by Yu et al. [Yu+20] and Zhang [Zha03] focus on linear regression and statistical methods like the clusters targets average. We, in comparison, train DNNs on each of the clusters,

combining them with different ensemble methods, like proposed by Trivedi et al. While they relied on simple ensembling using the average of the prediction models results, we will go further and evaluate the mean and median values and the performance of decision trees.

Zurada et al. [ZLG11] mention the small sample size of a high number of studies, that enables transfer learning as a solution to overcome data shortage. Bozinovski and Fulgosi first addressed this approach of inferring knowledge between areas as an explicit area of machine learning in 1976 [BF76]. Since then the number of studies regarding this topic increased fast, but to the best of our knowledge there is no comparable approach of combining transfer learning with clustering-based regression models yet. Segmentation-based predictions in the area of transfer learning are mostly used in the context of classification, e.g. transferring the labels of a classification on the source domain to a clustering of unlabeled target data [Ach+12; Yan+09]. The unsupervised clustering here acts as a grouping to which the source labels can be assigned. We instead transfer the clusterer between the data sets and optionally the corresponding models.

3 Preliminaries

3.1 Deep Neural Networks

Deep learning is an often used approach in machine learning. A deep neural network, inspired by the biological brain structure, is an artificial network with multiple *hidden layers* on which densely connected *neurons* are placed. Each neuron takes the *weighted* output of the neurons in the previous layer, applying an *activation function* σ and adding a *bias* $b \in \mathbb{R}^d$. Therefore, the output $Y \in \mathbb{R}^h$ of the i-th layer, taking $x \in \mathbb{R}^d$ as input and multiplying it with the corresponding weights $W \in \mathbb{R}^{h \times d}$ results in the equation

$$Y = \sigma(W \cdot x + b).$$

The training of a neural network involves two steps: *Forward propagation* refers to passing the data from the input to the output layer applying the transformations of the neurons to it and evaluating the final predictions with a *loss function*. In the *back propagation* phase weights and biases are updated backwards for each layer using the gradient descent. Those steps are iterated for the number of epochs to train the DNN, before we can obtain predictions for new data.

3.2 Clustering

Clustering is an unsupervised machine learning task of automatically discovering groups in data. Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ denote a finite set of data points. The objective of any clustering algorithm is to find $k \in \{1, \ldots, n\}$ clusters, $S = \{S_i \mid 1 \leq i \leq k\}$ with $S_i \subseteq X$ and $\bigcup_{i=1}^{k} S_i = X$ such that the similarity within a cluster is maximized. The criteria for similarity measurement depends on the used clustering method.

3.2.1 Centroid-based clustering

The data is organized in clusters S_j, each being represented by a central vector, the so-called centroid c_j. Let d_E denote the most commonly used squared Euclidean

distance and $\mathbb{1}_{S_j}(x_i)$ an indicator function with

$$\mathbb{1}_{S_j}(x_i) = \begin{cases} 1 & \text{if } x_i \in S_j, \\ 0 & \text{else.} \end{cases}$$

K-Means, the most known representative of centroid based clusterings, minimizes the cost function CF given by the within-cluster sum squared error with

$$CF(X, S) = \sum_{j=1}^{k} \sum_{i=1}^{n} \mathbb{1}_{S_j}(x_i) \cdot d_E(x_i, c_j).$$

Algorithm 1: *Llyod's K-Means algorithm*

Data: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, cluster number k
Result: S, Set of k clusters

```
Initialize k centroids c_i randomly
while changes in clustering S do
    for j ← 0 to k do
        S_j = {}
    for i ← 1 to n do
        Assign x_i to cluster S_j with min_{1≤j≤k} d_E(x_i, c_j)
    for i ← 1 to k do
        c_i = 1/|S_i| Σ_{x_i ∈ S_i} x_i
```

While there are several implementations, we focus on Lloyd's iterative algorithm in Algorithm 1. This algorithm alternates between assigning data points to randomly initialized centroids and recomputing the representative of each cluster until a stable state is reached. As K-Means has an average complexity of $O(k \cdot n \cdot T)$ with n being the number of samples and T the number of iterations, it is a very fast clustering method and therefore widely-used.

3.2.2 Hierarchical clustering

In comparison to this centroid-based clustering any hierarchical clustering seeks to build a hierarchy of clusters in an agglomerative or divisive way. Agglomerative clustering, a so called "bottom-up" approach, starts with treating each observation as an own cluster, pairwise merging most similar pairs with up-going hierarchy.

This iterative process ends when all clusters are merged together, resulting in a so-called dendrogram, which shows the hierarchical relationship between clusters. In the divisive "top-down" approach all observations are seen as one cluster and recursively split, so that less similar data points are distributed along different clusters. Therefore, a distance metric needs to define similarity between clusters e.g. the above used Euclidian distance. As this is supposed to operate with two data points only, the *linkage criterion* decides, which points are chosen to calculate the similarity of two clusters. For example the distance can be computed between the most similar points (*single-linkage*), the least similar ones (*complete-linkage*) or the centers of each cluster (*average-linkage*). Standard algorithms come with a time complexity of $O(n^3)$ and a memory amount of $\Omega(n^2)$.

3.3 Transfer Learning

Transfer learning seeks to improve the performance of learners on a target domain through the transfer of knowledge from a related source domain. In the style of Zhuang et al. [Zhu+20] we denote a domain by $\mathcal{D} = \{\mathcal{X}, P(X)\}$ with $\mathcal{X}$ being the feature space and $P(X)$ the marginal distribution where $X = \{x_1, \dots x_n\}, x_i \in \mathcal{X}$. Here $\mathcal{X}$ is the set of all possible values for the chosen set of features, while X denotes an instance set. For a given domain $\mathcal{D}$, a task $\mathcal{T}$ is defined as a two-element tuple of the label space $\mathcal{Y}$ and a predictive function $f(\cdot)$. f is learned from the feature vector/label pairs $(x_i, y_i), x_i \in \mathcal{X}, y_i \in \mathcal{Y}$. For a source domain $\mathcal{D}_S$, its corresponding source task $\mathcal{T}_S$ and a target domain $\mathcal{D}_T$ with task $\mathcal{T}_T$, transfer learning aims to improve the target predictive function $f_T(\cdot)$, that means to learn the target conditional probability $P(Y_S, X_T)$. There are several typical scenarios that involve transfer learning: Domains differ either in their feature space, meaning $X_T \neq X_S$ or their marginal distribution $P_T(X) \neq P(X_S)$.

Throughout this thesis we will study the house price in different prefectures of Japan, operating with the same feature spaces $X_T = X_S$, referred to as homogenous learning, and a different conditional probability distribution of the source and target tasks, that is $P(Y_t, X_T) \neq P(Y_S, X_S)$.

3.4 Evaluation metrics

For the purpose of evaluating our clusterings and predictions, we introduce a number of standard evaluation metrics. As clustering belongs to the unsupervised learning tasks, the non-labelled samples make it a complex task to perform and

evaluate. Though we can measure the performance of clusterings by the performance of predictions as well, we use a number of methods for the hyperparameter selection e.g. the number k of clusters. The silhouette score, an internal validation metric as it relies on information in the data only, gives the similarity of an object to its own cluster (cohesion) compared to other clusters (separation). For a data point $x_i \in S_j \subset X$ let

$$a(x_i) = \frac{1}{|S_j| + 1} \sum_{x_j \in S_j} d_E(x_i, x_j)$$

be the average intra-cluster distance of x_i and

$$b(x_i) = \min_{j \neq k} \frac{1}{|S_k|} \sum_{x_k \in S_k} d_E(x_i, x_k)$$

be the average distance of x_i to the closest neighboring cluster. The *silhouette value* of x_i is now defined as

$$sil(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}$$

with a range of $[-1, 1]$. We observe that the smaller $a(x_i)$ is compared to $b(x_i)$, meaning the distance to its own cluster being smaller than to the closest neighboring cluster, the closer $sil(x_i)$ is to 1. The silhouette value of a whole cluster S_j is given by the average value of $sil(x_i)$ with $x_i \in S_j$. The *silhouette score*, used later for hyperparameter determination, is defined by Rousseeuw [Rou87] as the mean of $sil(x_i)$ for $x_i \in X$. Values nearby 1 state better connectedness within the cluster and separation to other clusters and therefore indicate better clusterings.

Another heuristic for choosing the number of clusters is given by the elbow method. The inertia of a clustering, defined as the sum of squared distance of samples to their closest cluster center, is wished to be as small as possible. As it decreases with the rising number of clusters k, the information contained in the clusters decreases as well. Over-fitting is more likely to appear. Considering the threshold between a smaller difference of data points in their cluster and an appropriate number of clusters, that does not split natural groups, we pick the value of k at the elbow point on the inertia graph.

Setting the hyperparameters using the cluster related validation metrics, we evaluate the utility of clusterings for predictions with a number of standard error measures for supervised learning. For n data points let y be the vector of observed values on the target variable and $\hat{y}$ the predicted values by some model. The mean

squared error (MSE) is defined as

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

and the mean average error (MAE) is given by

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|.$$

As both the MSE and the MAE depend on the scale of data, we will use the mean absolute percentage error (MAPE)

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i}$$

to obtain comparable results. It offers the following advantages: First it is a relative measure, providing the errors in terms of percentages, it is quite understandable and therefore widely-used. More importantly, the results are not influenced by the magnitude of the data. Hence, the outcome is comparable, which is quite important as we seek to do transfer learning on different parts of the data.

4 Cluster-based prediction models

In everyday life we automatically group real estate according to their qualities, e.g. object type, facilities or location. For instance, we experience single family homes in rural areas to be much different from rental apartments in urban areas but less distant from a semi-detached house. Thinking about expensive properties we would probably consider big buildings or those that are located in upmarket areas rather than small and less well located ones. If we had to assign prices, objects with analog characteristics would be in a similar price range. So we would expect the variance of the price along similarity based formed groupings to be smaller than over the whole set of real estates.

The objective of our real estate valuation is to train a model on predicting the properties prices as precise as possible. Hence, we will explore how predicting on previously clustered data enhances the accuracy.

Using location data

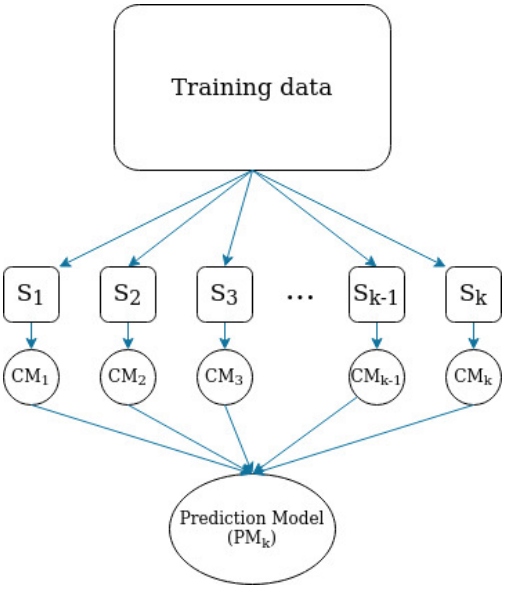
As Kryvobokov [Kry07] stated out, location data is of high importance in real estate valuation. DNNs usually struggle with connecting nearby properties due to their character of rather learning direct relationships between a properties attribute and its value. For that reason we provide the precalculated average price of the neighborhood as input. This derives from the case-based reasoning evolutionary algorithm (CBR+EA) by Angrick et al. [Ang+21] described in Section 5.1.

4.1 Cluster-based deep neural networks

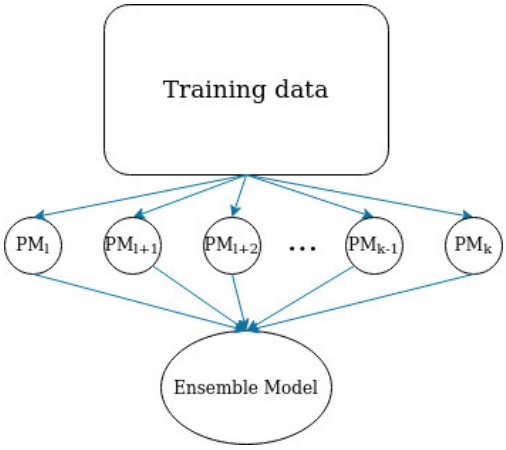
While there are many ideas how clustering might help in classification tasks, there is also some evidence for the power of using this unsupervised technique in regression problems [TPH11], like we face in this thesis. Therefore, we train prediction models on each group of a beforehand created clustering. Since the data within each cluster is more similar, we expect the valuations of each model to be more precise.

Our architecture is inspired by Trivedi et al. [TPH15] and visualized in Figure 4.1 (a): Clusters are formed by a simple clusterer e.g. K-Means, that we fit to

(a) A single prediction model.



(b) An ensemble prediction model.



the training set. Afterwards we train models on each of the clusters S_i, which they refer as cluster model CM_i $(1 \leq i \leq k)$.

In comparison to them, we will train a DNN instead of a linear regression model which is more suitable for the real estate market as shown by Shinde et al. [Shi+19]. It adds the advantage of automated feature-engineering as the weights are assigned by the neural network, that compared to a linear regression model, learns to adjust them. Further, the model is able to process non-linearity features by the use of different activation functions. So we expect the results of our networks to be more accurate, like shown by Lee et al. [Lee+17].

The final prediction model PM_k out of the separate cluster models can then be evaluated by assigning every test sample to a cluster and estimating with the corresponding cluster model. Trivedi et al. [TPH15] built an ensemble prediction model (EM), combining the prediction models PM_k for a number of $k \in \mathbb{N}$ by simple averaging Figure 4.1 (b). We will go further, using a number of different aggregation methods and comparing the effect of mathematical methods e.g. mean or median and training a simple model to select the best prediction.

For clustering we will use K-Means, introduced in Section 3.2, successfully applied to real estate by Dong et al. [Don+15], Bourassa et al. [BHP02] and Chen et al. [Che+07] to real estate. The *a-priori* determination of the cluster number k is often criticized, as it asks for the commitment to a manually chosen number of segments. We deal with this problem by ensembling models build with different ks, so we do

not require a fixing of this hyperparameters. K-Means is easy to implement and adapts to the use of different distance metrics in future work. Further, the variables of the algorithm can be set using the training data only. Afterwards they can be applied to the whole data set. In comparison to hierarchical clustering that involves building memory-intensive tree structures, K-Means scales well with large data sets.

5 Cluster-based transfer learning

5.1 Case-based reasoning

In Chapter 4 the prediction model was given by a DNN. As we are interested in the effects of clustering on the performance of different models, we introduce the case-based reasoning approach optimized via evolutionary algorithms CBR+EA that Angrick et al. [Ang+21] successfully applied valuation tasks. It gains knowledge about a new property by the consideration of similar properties, following a similarity function s that gives us the similarity of two cases from a set of cases C. It maps each pair of elements to a non-negative extended real number where the larger numbers denote higher similarity.

The final predictions are then made by an average prediction function $\text{pred}\colon C \to \mathbb{R}$ with C being a set of possible cases, $s\colon C \times C \to \overline{\mathbb{R}}_{\geq 0}$ the similarity function and $DB \subset C$ the set of cases whose value is known. A new case x is predicted by

$$\text{pred}(x) = \frac{\sum_{y \in DB} s(x,y) \cdot \text{value}(y)}{\sum_{y \in DB} s(x,y)}.$$

For more details on the similarity approach see the work by Galboa et al. [GLS11].

Larger data sets require a *pre-selection* as price determination of each property involves the computation of s for all other objects, which is not efficient anymore. Therefore, the samples within a fixed geographical radius are *pre-selected* to limit the data objects available for each new property. This is done by a *radial pre-selection* within a given geographical radius $r \in \mathbb{R}_{\leq 0}$ or via the selection of *K-Nearest-Neighbour*. We utilize the clustering for the pre-selecting process by only comparing objects in the same cluster. Clusters are mainly formed using K-Means, like described in Section 3.2.

The similarity function of CBR+EA, namely the inverse of a weighted quasi-norm, is found using an evolutionary algorithm. More details on the whole algorithm are given by Angrick et al. [Ang+21] and the work of Bals [Bal21].

5.2 Transfer learning

In everyday life we experience many advantages by inferring knowledge from one task into learning other tasks. Knowing how to play classic piano may help us to learn other music instruments, having a background in math and statistics enables easier access to other scientific areas. Also machine learning benefits from the reuse of a pre-trained model for a new problem, marked as transfer learning. Conventional algorithms are usually trained on specific tasks and have to be rebuilt completely on any other data distributions. Models supposed to solve complex tasks, need a high amount of training data. As the data gathering process is a difficult and expensive one, transfer learning provides a solution by going beyond specific tasks and domains and transferring knowledge from pre-built models. Let $\mathcal{D}_S = \{\mathcal{X}_S, P(X_S)\}$ denote the larger source domain, $\mathcal{D}_T = \{\mathcal{X}_T, P(X_T)\}$ the target domain and m the market price that we want to predict. The feature space is approximately the same, so we operate in the area of homogenous learning, introduced in Section 3.3.

5.3 Cluster-based transfer learning

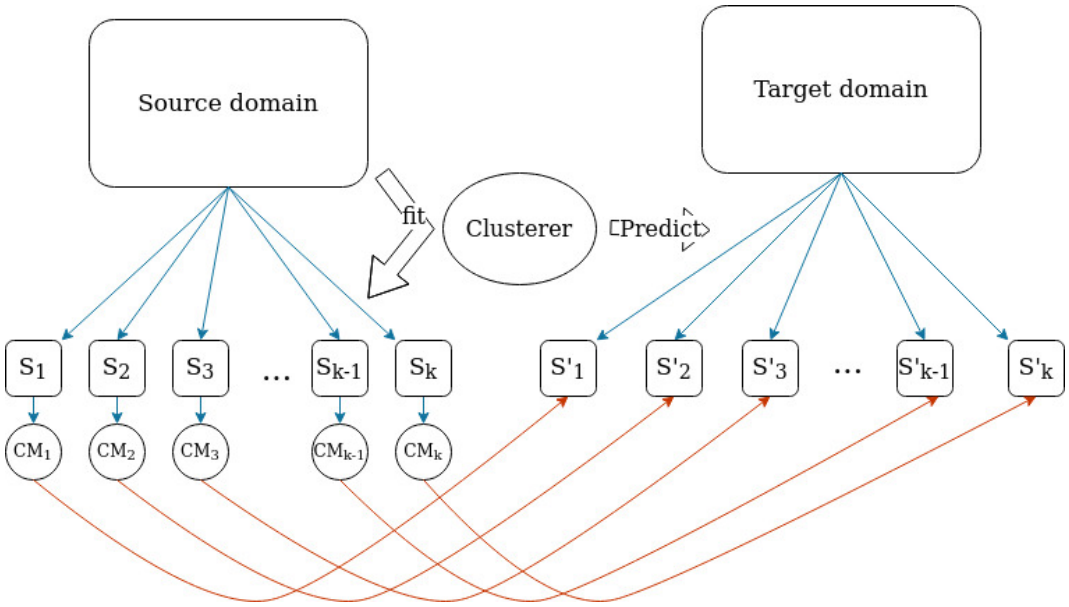
5.3.1 Clustering

We first introduce the explainability tree [Mos+20] that predicts cluster labels in an understandable way. The algorithm, namely Iterative Mistake Minimization (IMM), uses a variation of standard decision trees. An internal K-Means-clusterer computes reference cluster labels that shall be predicted by the decision tree. At each node the tree algorithm decides for the best split, minimizing the number of mistakes, where a mistake occurs when a threshold separates a point from its cluster according to the reference clustering. The decisions of the tree give us an explanation for the construction of the clustering.

We are comparing two different transfer approaches of clusterings. The algorithm to form the groups is marked as *clusterer* and k denotes the number of clusters.

For the first one, we build clusters on $\mathcal{D}_S$ using the target m as an additional input feature, that is handed in the construction of a tree using the IMM algorithm. While the original purpose was the explainability of a clustering, we use the formed tree to predict the labels of the target domain. Therefore, we modify the algorithm to hand in the previously predicted labels l_S from the $\mathcal{D}_S$ with the knowledge of m and fit the tree to make splits only knowing the remaining feature set, that is

Figure 5.1: Our cluster-based transfer model for DNNs.



without the knowledge of the target variable of D_T. Executing the formed decision tree upon the target domain $\mathcal{D}_T$ gives us a transferred clustering.

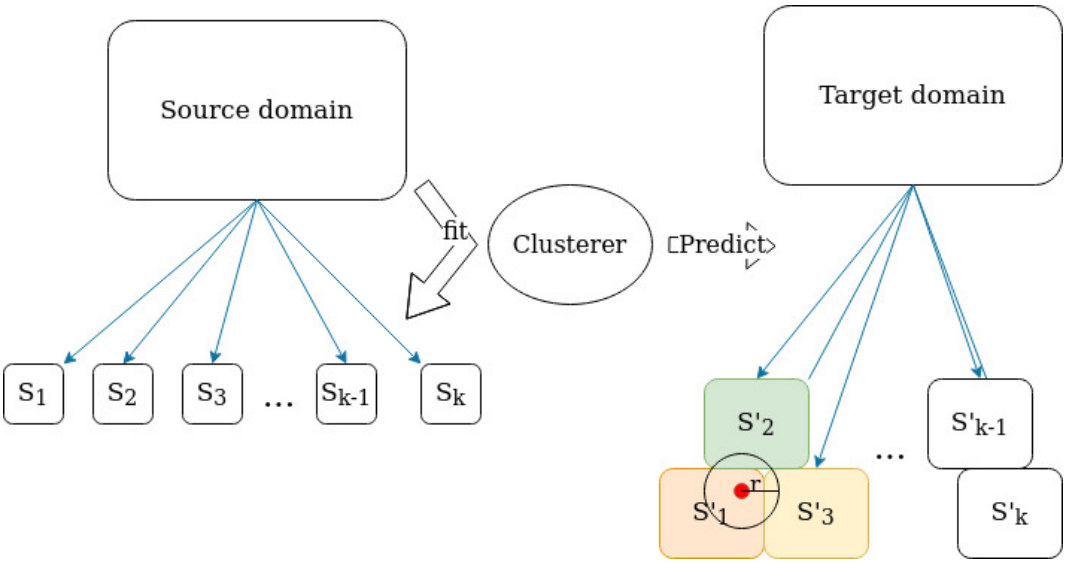
Secondly, we use a simple K-Means-instance fitted on the source domain to predict on the target domain. For this we use Lloyd's algorithm described in the Algorithm 1 and compute appropriate groups upon the source data, obtaining a set of cluster centroids. We can then assign each sample x of the target data the cluster number $i \in \{1, \ldots, k\}$ minimizing the distance of x to the clusters' centroid. As the used attributes are the same, we do not benefit from the house prices of the source domain, but from the datasets larger size only.

5.3.2 Model architecture

The architecture of our cluster-based transfer model that uses **DNNs** is visualized in Figure 5.1. We first build clusters $S_1 \ldots S_k$ on $\mathcal{D}_S$ and train DNN models on each of it. We then transfer the clusterer to the target domain to form clusters $S'_1, \ldots, S'_k$ like described in Section 5.3.1. The orange lines in the graphic indicate the final predictions on $\mathcal{D}_T$ using the cluster models from the source domain. We investigate the effect of retraining those models to the target domain in contrast to train completely new models for each target cluster S'_i.

Regarding the **CBR+EA** we do not use models from the source data set, but

Figure 5.2: Our cluster-based transfer model for CBR+EAs.



instead only transfer the clustering. So a new evolutionary algorithm is trained on the transferred clustering of the target data set. As visualized in Figure 5.2, the neighborhood restriction of a data point is limited not only on the radius $r \in \mathbb{R}_{\leq 0}$ but on the cluster membership as well. Therefore, the price of the red marked example data point is computed using similar properties from the intersection of cluster S'_1 and its radial surrounding.

6 Experiments

We now ask if clustering has positive effects in the presented regression settings. Therefore, we experimentally evaluate our approaches and compare them with known methods on a real-world data set.

6.1 Hypotheses

Due to heterogeneity in real estate data, we construct similarity-based groupings (see Chapter 4) and expect better prediction results by models that are fitted to more specific submarkets. Hence, we want to study the following hypotheses.

▶ **Hypothesis 6.1.** Training a DNN on each group of a beforehand built clustering and averaging the results according to the cluster size to construct the presented prediction model behaves better than a DNN trained on the whole data set. Adding different ensemble methods to combine models with different clustering hyperparameters leads to further improvements. ◀

As models typically benefit from a larger data set, we will validate the proposed methods (Chapter 5) that infer knowledge from a larger source domain.

▶ **Hypothesis 6.2.** Combining the approach of transferring clusterings from a larger source domain into a target domain with the CBR+EA algorithm for price prediction provides performance advantages. ◀

6.2 Set-Up

We evaluate the proposed approaches on a large real estate data set and describe the material and its preparation for usage in the following section.

6.2.1 The Data Set

The data set is given by the "LIFULL HOME'S Data Set" [LIF19], a large real estate data set from Japan that is available to computer science researchers worldwide through the Japanese National Institute of Informatics. After removing outliers

Table 6.1: Acceptable values for filtering outliers. Entries outside the ranges are removed.

Attribute	Lower Bound	Upper Bound
plane location - latitude	−1 140 843,77	1 414 596,17
plane location - longitude	−401 614,57	2 618 746,47
living area	20	2 000
construction year	1 500	2 025
market price	0	300 000 000

(see Table 6.1), that is data with unrealistic prices or properties outside of Japan, we end up with 723 115 data points. Our preprocessing includes an ordinal encoding of the object type and the prefecture, as well as scaling the attributes to the interval $[0, 1]$ with the minimum corresponding to 0 and the maximum corresponding to 1. By this, the effects of different scaled attributes are minimized. As cluster methods usually require filled columns without *nan*-values we impute the data using statistical imputation methods.

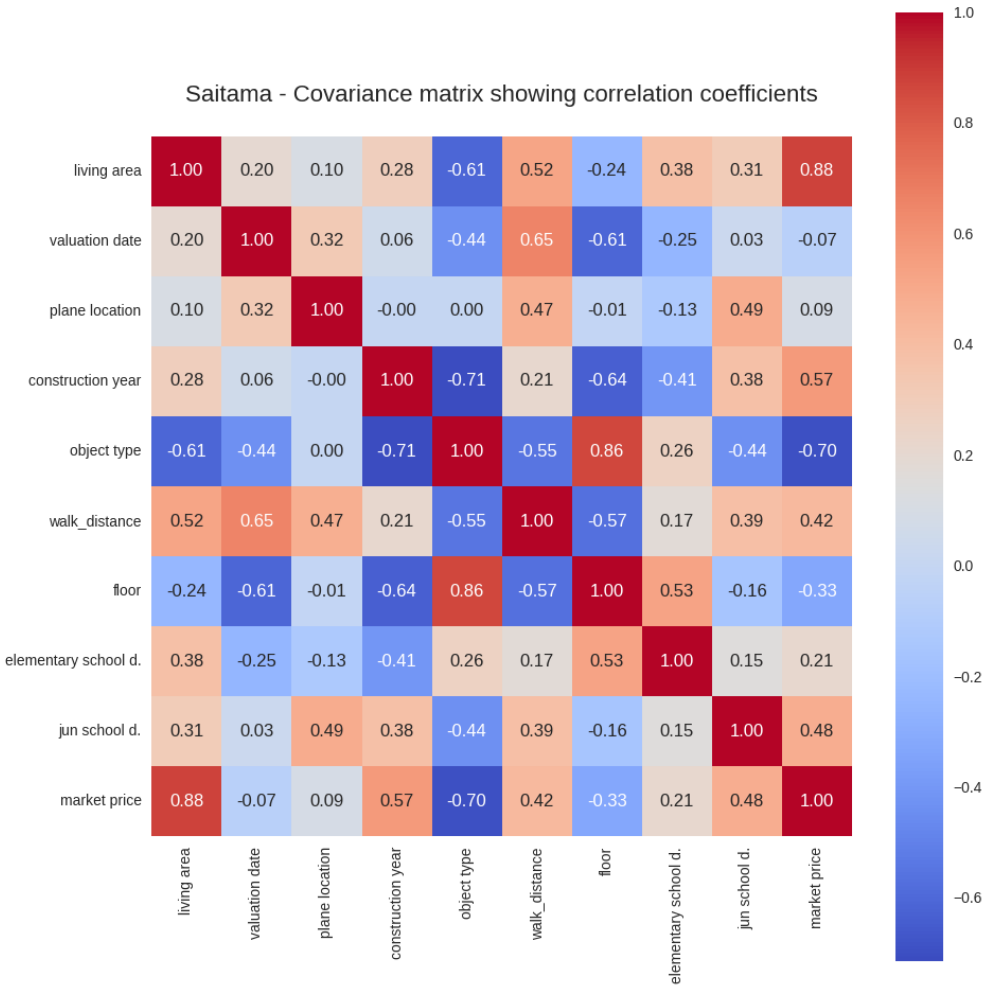
6.2.2 Data Exploration

We construct correlation matrices (see Figure 6.1 for the prefecture of Saitama and Figure A.1 for whole Japan) for measuring the relationships between attributes. For the variables $x, y \in \mathbb{R}^n$ with $n \in \mathbb{N}$ as the number of entries, the cell $\sigma(x, y)$ represents the covariance. Let $\bar{x}$ and $\bar{y}$ be the mean value of the data for the attribute x and y, respectively. Then the covariance is calculated by $\sigma(x, y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$. While values closer 0 indicate no linear correlation, the further away the correlation coefficient is from zero, the stronger the relationship between the two variables. The matrix gives us a high importance of *living area, construction year, object type* and the *distance to an elementary school* for the price prediction.

6.2.3 Evaluation Methodology

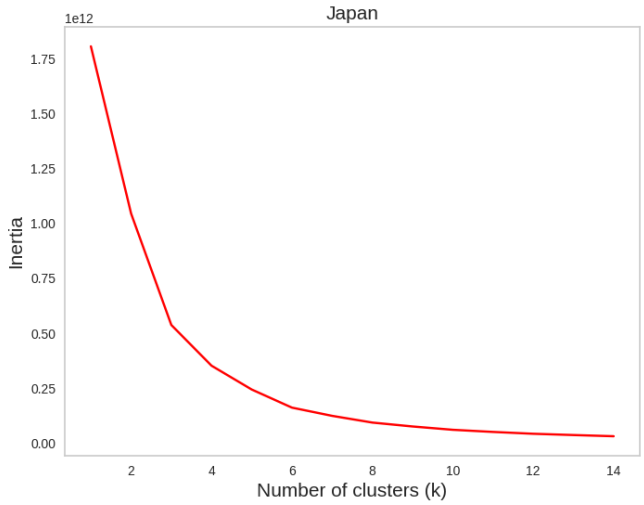
Commonly, the data is separated into a *train* and a *test set* using a random split with a ratio of 80% to 20%. As it is not only intuitive but also a usual practice of real estate price determination to forecast future prices knowing previous data only, we treat the properties that were collected before 2017-03-01 as known and use the remaining samples for the test set. This results roughly in the common 80/20

Figure 6.1: Covariance matrix for Saitama.



ratio and mirrors the usual procedure the best, preventing time leakage efficiently. For the ensemble model Figure 4.1 (b) we introduce a third, fixed validation data set to train our decision tree on the predictions of different models upon this set This is obtained by a random 75/25 split of the train data, taking the latter part as validation set. In the end we get the common 60/20/20 split into training data, validation data and test data.

Figure 6.2: The inertia of different clusterings on whole Japan.



6.3 Methods

We differ the phases of building similarity-based groupings and constructing the prediction models (see Chapter 4 and Chapter 5). First, we will give general details on the clustering, second, outline the implementation of the prediction models.

6.3.1 Clustering

To get suitable values for the cluster number we use the elbow method as well as the silhouette score explained in Section 3.4, resulting in an optimal choice of $k = 5$ (see Figure 6.2).

The feature columns, beforehand already limited to the ones with at least 50% of the samples being defined, are determined with the knowledge of the covariance matrices (see Section 6.2.2). These results are confirmed by a *principal component analysis*, a technique for dimensionality reduction. A large set of variables is transformed into a smaller one preserving as much of the data's variation as possible. The idea behind this technique is the computation of new variables, *principal components*, from the eigenvectors of a covariance matrix, such that most information is contained in the first components. For more details we refer the reader to [BS14]. We reduce the feature space to a size of two and get the decisive columns for the

new feature construction by interpreting the explained ratio. Further, we fit a *decision tree* to the data set and evaluate its splits on the attributes. Splits on a higher level indicate a higher feature importance. We then come up with a feature set of *construction year*, the *living area*, the *prefecture* as well as the *object type*, removing the *prefecture*-attribute when clustering on regional parts of the data set only.

The clusterer is fitted using the previously explained train-test-split. While this works well for centroid-based clusterings like K-Means, hierarchical clusterings do not support the separation of fitting and label-predicting phase due to their nature of building tree-like data structures, called dendrograms. Additionally confronted with the required memory size of up to $\Omega(n^2)$, we take a sample from the train data, obtain labels using hierarchical clustering and predict the cluster membership of the remaining data set, including the test data, with the simple classifier *K-Nearest Neighbors*.

6.3.2 Cluster-based prediction models

The data set is split into the same training and test set as outlined in Section 6.2.3. Starting with $k = 4$, which is supposed to give the best clustering according to the silhouette score, we build clusterings up to $k = 12$ and run following procedure on each of them: A DNN, developed by Angrick et al. [Ang+21] and trained on the whole model, serves as the base of our computations. We refer to it as base model. For each cluster we train a new model, with the same architecture and retrain the base model as well. In case of some clusters being too small to efficiently train a new DNN, we introduce a *threshold* of 5% of the whole data set, chosen by experience. Then we decide dynamically to use the base model with retrained new layers, when the cluster size is lower. For evaluation, we weight the results of the cluster models with the corresponding test size of the cluster and obtain metrics for the whole prediction model PM_k. The combination of different prediction models PM_k is done via the computation of mean and median values and the construction of a decision tree that was fitted to the validation data. To evaluate the effect of clustering separated from DNNs we run linear regression models as well.

6.3.3 Cluster-based transfer models

With respect to the transfer approach in Hypothesis 6.2 we choose the Saitama data set as source domain since it offers the highest number of samples. Tokyo will mainly serve as the target domain because of its high variance of the market price and the comparatively high number of samples. Nevertheless, we repeat the

Table 6.2: Statistics for the most relevant prefectures.

prefecture	value count	mean price	variance price
Saitama	189253	29841346.3	$6.6 * 10^{18}$
Tokyo	134008	57517908.2	$6.8 * 10^{18}$
Kanagawa	120371	36828987.8	$3.2 * 10^{14}$
Chiba	76070	26604919.1	$1.3 * 10^{14}$
Osaka	42116	36293415.5	$2.5 * 10^{18}$
Hyōgo	28549	27777198	$2.6 * 10^{15}$
Aichi	26619	28624500.6	$2.32 * 10^{14}$
Miyagi	13405	27286827.8	$7.3 * 10^{13}$
Nara	12290	23575916.5	$3.9 * 10^{14}$
Kyoto	10938	27637885.3	$4.8 * 10^{16}$
Ibaraki	10231	21273825.7	$1.2 * 10^{14}$

experiments on the smaller data sets of Osaka and Kyoto, that also provide a high variance of the market size, pictured in Table 6.2.

The first approach of transferring a clustering, outlined in Section 5.2, requires the fitting of a K-Means-instance on the Saitama data set and prediction of the labels on our target data sets. The second one, building an explainable tree under inclusion of the target variable from the source domain, struggles with different initialized K-Means-instances. The K-Means on the source data is aware of the price-attribute, while the K-Means-instance being in charge of the prediction on the target data set is not allowed to see the prices. This requires the construction of an appropriate matching. Thirdly, the clustering is built on the target domain only, serving as reference.

For each of the clusterings we run the CBR+EA first, secondly DNNs on the target data set and thirdly DNNs transferred from the source data set. Details on the training of the DNN are given in Section 6.3.2, for the CBR+EA see Angrick et al. [Ang+21].

6.4 Results

This section provides the results necessary to validate the hypotheses in Section 6.1 As the outcomes vary with marginal differences, we carry the experiments out several times and show the median of the runs.

Table 6.3: Statistics of the clustering with $k = 4$, obtaining an overall MAPE of 13.6 percentage points. The school distance contains the distance to junior and elementary schools, given in meter. Prices are given in dollars. For a train size greater than 5% the model is rebuilt completely, otherwise pretrained from the base model. The used value is marked in bold. The last column contains the metrics of applying the base model to the data directly, without any adaptions due to clustering.

No.	∅ urbanity score	σ urbanity score	∅ school dist.	σ school dist	∅ year of construct.	σ year of construct.	∅ living area	σ living area
S_1	2,15	0,9	1096/765	642/442	2 015	2,7	98	13
S_2	2,07	0,7	1218/839	725/501	2 012	9,6	104	36
S_3	2,14	0,7	1223/810	762/505	2 012	9,5	101	30
S_4	2,44	1,7	1312/909	917/621	1 984	11,5	114	77
all	2,15	0,9	1154/795	700/478	2 012	9,7	102	33

No. price	∅ market price	train size	test size	MAPE pre	MAPE new	MAPE base
S_1	$3.4 \cdot 10^7$	211 643	85833	17.4	**11.1**	11
S_2	$2.7 \cdot 10^7$	76496	30180	14.8	**14.7**	13.9
S_3	$3.3 \cdot 10^7$	102618	48460	15.3	**13.7**	13.5
S_4	$2.5 \cdot 10^7$	20913	9748	**30.5**	36.6	35.5
all	$3.2 \cdot 10^7$					13.6

6.4.1 Clustering on DNN

Hypothesis 6.1: DNN outperforms other models. As we are interested in the effects of the clustering, we have a closer look into one of the clusterings first. For practical purposes, we consider the clustering with $k = 4$, shown in Table 6.3.

The table presents the features of the properties for the created clusters. The first three clusters appear similar in building-inherent features, like living area and construction year, but differ in location-based attributes, that are the school distances and the urbanity score. So we notice that without handing in any locational column beside the prefectures the clusterer already seems to support geographical differences like how populated regions are. Properties in cluster 1 come with nearer school distances resulting in higher prices. Hence, we suppose those properties being located in more urban or central areas, which is supported by the comparatively smaller living area. Further the urbanity score of buildings in cluster 2 is lower than in cluster 1, which might be an explanation for the lower market value. As it comes with similar attribute values as cluster 3 and differs in the urbanity score only, the lower market value of cluster 2 ($2.7 \cdot 10^7$ compared to $3.3 \cdot 10^7$) shows the negative effect less urbanity on market price. Cluster 4 contains rather old and

large properties with an average construction year of 1984 and a size of 114 square meters. Further, computing the standard deviations of construction year and living area, we obtain significantly higher values for cluster 4. Standard deviation in construction year is 11 years (2, 9, 9 in clusters 1, 2, 3, respectively) and 78 for living area (13, 37, 78). This indicates the role of cluster 4 as "rest cluster", including the samples that do not fit in any other group and explains the worse prediction results. In summary, we observe differences in the mean values of the clusters, that indicate some groupings according to market segments. As the standard deviations within the clusters are not significantly lower than in the whole data set, the clustering does not represent a clean segmentation into submarkets though, but suffers from a high number of buildings not assigned properly.

In Table 6.4 we show the results for the cluster-based predictions. Firstly, we observe the low relation between the number of clusters and the goodness of results regarding the DNNs. The best MAPE regarding the DNNs was achieved by a split into 6, 9 and 10 clusters, with a MAPE of 13.4 percentage points which is nearly the same as obtained by the reference model. For significantly higher $k's$, the results are even worse with MAPEs of 14.5 ($k = 30$, $k = 100$) and 15.8 percentage points ($k = 50$). As pointed out combining the predictions of several methods is most promising, giving us a MAPE of 13.3 when averaging the results obtained by all prediction models or taking the median. Building a decision tree that learned the predictions of all cluster models with regard to the market price on an additional validation set performs best with a MAPE of 12.3. The experiments with random groups show that clustering provides a value for DNNs though. For $k = 30$, we obtain a MAPE of 14.5 percentage points for similarity-based groups and 15.0 for random groups, $k = 50$ gives 14.8 vs. 51.1, $k = 100$ results in 14.5 vs. 15.6 percentage points. Again, we notice that predictions get worse for a higher number of clusters. Hence, we can infer that the DNN in fact benefits from less heterogeneity in the data, but can not handle a smaller number of data. The heterogeneity is not that bad that it compensates the split of the data.

In a number of additional experiments we follow Trivedi et al. [TPH15] and build linear regression models on each of the clusters. Those simpler models are supposed to handle a small amount of data. By this we want to determine whether the noted bad performance is caused by the clustering or the usage of DNNs. We observe significantly improvements on the MAPE, the higher the cluster number is (see Figure A.2 for the visualization of results). While starting with $k = 4$ gives us a MAPE of 18.5 equal to the reference linear regression model, we obtain a MAPE of 15.7 percentage points for $k = 30$, 15.5 for $k = 50$ and 15.3 for $k = 100$. In comparison to DNNs we note the high impact of minimized heterogeneity. For smaller and more homogenous clusters we obtain better predictions. To exclude

Table 6.4: Error measures for a different hyperparameter selection. We denote by k the cluster number in our clusterings.

		DNN		Linear Regression	
		MAPE (in %)	**MAE** (in €/m^2)	**MAPE** (in %)	**MAE** (in €/m^2)
Reference		13.6	46121	18.6	55445
Clustering	$k = 4$	13.6	42124	18.5	58301
	$k = 5$	13.6	41538	18.5	54598
	$k = 6$	13.4	41968	18.1	53746
	$k = 7$	13.5	41542	18.1	53651
	$k = 8$	13.8	41389	18.1	53123
	$k = 9$	13.4	42139	18.0	52829
	$k = 10$	13.4	41840	17.9	52461
	$k = 11$	13.5	42502	17.9	52468
	$k = 12$	13.5	42093	17.9	52468
	$k = 30$	14.5	45859	15.7	48343
	$k = 50$	14.8	47114	15.5	47980
	$k = 100$	14.5	46828	15.3	47204
Random Split	$k = 30$	15.0	48120	18.6	55557
	$k = 50$	15.1	49324	18.7	55578
	$k = 100$	15.6	50050	18.7	55710
Ensembles	mean	13.3	41666	15.3	47492
	median	13.3	41402	15.3	47455
	decision tree	12.3	38412	15	43876

the possibility of performance improvements due to the split into subsets only, we evaluate a random, not similarity-based clustering as well, which results in worse errors and therefore confirms our approach. For $k = 30$ and the choice of linear regression models we obtain a MAPE of 18.6 percentage points, which is similar to the reference, higher cluster numbers perform worse. So we can infer the utility of clustering on real estate for linear regression algorithms. Building ensembles with the predictions obtained by combined linear regression models from $k = 4$ to $k = 100$ provides similar results as a high number of k. Taking the mean or the median of the predictions gives a MAPE of 15.3, constructing a decision tree with the maximum depth of 8 results in a MAPE of 15.

Table 6.5: Error measures for experiments regarding the effects of clustering on the CBR+EA. The data set includes full Japan

	MAPE (in %)	**MAE** (in €/m^2)
Reference	12.1	35500
Clustering		
K-Means, k=7	12.1	36128
K-Means, k=7, small clusters combined	12.05	36053
K-Means, k=100	13.5	38900
Agglomerative clustering, k=7, linkage=ward	14.5	41400
PCA-based clustering	12.3	36000

We can summarize that DNN is not the preferred model for our clustering approach. While linear regression models can handle a smaller number of data, those networks typically require more data. This leads us to the transfer approach presented in the thesis.

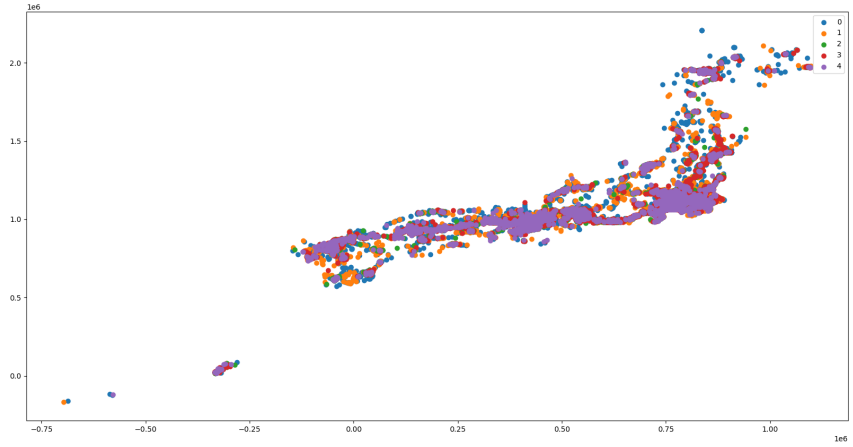
6.4.2 Transfer of Clustering

Hypothesis 6.2: Transferring clusterings outperforms classical cluster buildings. First we evaluate the impact of clustering on the CBR+EA. Running it on unclustered data (whole Japan) gave us a reference of 12.1 percentage points, while clustering with K-Means into seven groups leads to a MAPE varying from 12.05 to 12.2 percentage points. Other cluster methods performed worse.

Note that the clustering approach does not provide significant improvements to the application of a case-based-reasoning evolutionary algorithm without clustering. A reason for this can be found in the mode of operation of this algorithm. The first phase, in which we include the clustering, is about the pre-selection of similar samples. While the former pre-selection only considered the geographical position, clustering brings in additional limitations based on attributes that is not influenced by location information as we see in Figure 6.3. The results indicate that further restrictions on the sample selections by clustering are not helpful, though.

In spite of the non-improving character of clustering on the applied algorithm, we now discuss the additional transfer. The experiments were carried out on the prefectures Tokyo, Osaka and Kyoto represented in the labeled columns.

Figure 6.3: Geographical distribution of the clusters for $k = 5$.



CBR+EA Considering the **EA**-labeled rows of Table 6.6, we note that none of the applied approaches outperformed the algorithm without limitations due to cluster or transform. The first line, fitting a K-Means-instance to the Saitama data set and predicting the labels on the target data set, gives us the worse results, with a MAPE degradation of 0.1 percentage points in Tokyo, 0.7 in Osaka and 1.3 in Kyoto. Secondly, regarding the transfer with the explainable tree that includes the knowledge of the prices in the target domain and therefore promised high potential for improvement, we do not experience better values either. We notice a debasement on the prefectures to the reference model without clustering (10.6 vs 10.5 in Tokyo, 14.8 vs.14.3 in Osaka, 18.9 vs. 18.5 in Kyoto). In comparison, clustering without any transfer in the respectively third line holds similar values to the applied methods with a small improvement of 0.1 percentage points to the reference model on Tokyo, and a degradation of 0.5 percentage points in Osaka and 0.1 in Kyoto. Hence, we note that transfer does not hold positive effects on the evolutionary algorithm. The reason for this may be found in the differences between the prefectures. We notice that the transfer performs better for Tokyo, that is located next to Saitama, than for Osaka and Kyoto. Therefore, we expect Tokyo's and Saitama's living statistics to be more similar to each other [Org]. As Osaka and Kyoto come with 42116 and 10938 data points only, while Tokyo offers 134008 samples, it is reasonable that the limitations due to an additional split into

Table 6.6: Error measures comparing different clustering configurations for the evolutionary method with known similarity functions in Tokyo, Osaka and Kyoto. The cluster number is set to 5. First we transfer the clustering only (a), second we transfer corresponding models as well (b).

		Tokyo		Osaka		Kyoto	
		MAPE (in %)	**MAE** (in €/m^2)	**MAPE** (in %)	**MAE** (in €/m^2)	**MAPE** (in %)	**MAE** (in €/m^2)
Reference	*EA*	10.5	46500	14.3	33400	18.5	43900
	DNN	12.5	47340	20.5	45832	20.8	46215
a)							
EA	Transfer K-Means	10.6	47700	15	35300	19.8	46600
	Transfer - *IMM*	10.9	47500	14.8	34700	18.9	45000
	K-Means	10.4	46300	14.8	34300	18.6	43400
DNN	Transfer K-Means	12.5	53916	20.1	48410	19.7	46638
	Transfer - *IMM*	12.9	55542	20.6	50546	20	46368
	K-Means	12.5	54328	21.3	52732	20.1	50107
b)							
DNN	Transfer K-Means	12.4	57003	21.5	59388	21.2	52427
	Transfer - *IMM*	13.6	62184	22.3	55917	21.8	52672
	K-Means	12.9	58698	23.5	59388	21.5	52672

clusters are not bearable for the evolutionary algorithm. This again argues against the functionality of the transfer approach on this algorithm.

DNN Before we fully reject our hypothesis, we evaluate the transfer approach with DNNs, shown in Table 6.6. First we want to compare the approach of models on transferred clusterings to non-transferred clusterings, that is the comparison of the first and third line in the DNN blocks of the table. We notice that transferring a K-Means-instance results in better values than fitting a K-Means to the target data only. For Tokyo, we obtain improvements of 0.5 percentage points (12.4 to 12.9) with transferring the models as well part (b) and steady values for the transfer of the clustering only (12.5). For Osaka the decrease in the MAPE is 1.2 (20.1 to 21.3) in part (a) and 2.0 (21.5 to 23.5) in part (b) and in Kyoto it is 0.4 (19.7 to 20.1) and 0.3 (21.2 to 21.5). Hence, we can hold the positive effect of transferring a clustering from a source domain to a target domain to building a clustering on the

target domain only when using DNNs for prediction afterwards. A reason for the differences to CBR+EA can be found in the non-similarity based character of DNNs. Regarding the transfer of clusters with the explainable tree (IMM) that integrates source domain knowledge of the values to predict (market price), we notice no notable improvement compared to transferred K-Means. Every experiment gives us a debasement compared to the transferred K-Means, e.g. we reach a MAPE of 13.4 percentage points for the non-transferred models (a) in Tokyo, that is worse than the previously obtained 12.8 percentage points.

When exploring the effects of transferring the corresponding cluster model as well, we note the negative consequences for the predictions. For each target cluster, picking the model pretrained on the corresponding source cluster gives us significant debasements. With the transfer-models-architecture from Figure 5.1 we achieve best MAPEs of 12.4 percentage points in Tokyo, 21.5 in Osaka and 21.2 in Kyoto compared to 12.3, 20.1 and 19.7 for training new models on the target data.

Hence, we can summarize the outcome in the following way: Both clustering and transfer does not hold positive effects on the CBR+EA. Regarding DNNs, transfer of K-Means-clustering has positive effects, but the transfer of DNNs has not. Again, we note the non-improving usage of clustering for DNNs, already outlined in Section 6.4.1. While part (a) in Kyoto gives us small improvements to the non-clustered reference model, clustering has no significant positive effects for predictions with DNN. Therefore, future work on transfer clusterings includes the usage of linear regression models that are already found to handle similarity-based groups (see Section 6.4.1).

7 Conclusions & Outlook

In this thesis we analyze the value of clustering for predictions in the context of real estate valuation. We consider the traditional machine learning on one domain only as well as the area of transfer learning. For the former, we divide the data set into subgroups and evaluate the performance of a cluster-based prediction model. Thus, we train a model for each of the clusters. The price of an arbitrary sample is determined with the model of the cluster it belongs to. Further we use a number of ensemble techniques combining the results of different prediction models. As the data amount in the subsets is too small to properly train a DNN, future work includes the training of more simple models e.g. linear regression models. We want to deepen our studies on dynamically constructed models by the clusters characteristics via hyperparameter tuning. The low relation between k and the goodness of the corresponding model for DNN is pointing to a less qualitative market segmentation. This suggests a split into submarkets that relies on external knowledge as well, shown by Chen et al. [Che+07]. Therefore, future work includes clusterings based on administrative and expert-defined boundaries, as well.

For our transfer learning approach we first evaluate the effects of clustering on CBR+EA, which are not promising. Further we experience transferring clusterings from a source domain into a target domain not to be auspicious. Though, a transfer of DNNs trained on a clustering on the source domain and transferred to corresponding clusters on the target domain leads to improvements. Further work here would include the definition of distance functions translating the feature space of the source domain to the target domain.

As we notice significant performance differences upon the cluster size, we seek to find a way building clusterings that have a more equal size without biasing natural groups in the data. One way to do so would be data augmentation for to small clusters or the merge of similar groups or a fuzzy clustering.

Bibliography

[Ach+12] A. Acharya, E. R. Hruschka, J. Ghosh, and S. Acharyya. **Transfer Learning with Cluster Ensembles** (2012) (see page 4).

[Ang+21] S. Angrick, B. Bals, N. Hastrich, M. Kleissl, J. Schmidt, V. Doskoč, M. Katzmann, L. Molitor, and T. Friedrich. **Towards Explainable Real Estate Valuation via Evolutionary Algorithms** (2021) (see pages 2, 11, 15, 23, 24).

[Bal21] B. Bals. **Valuation of Real Estate Properties using Data-Driven Similarity Search**. 2021 (see page 15).

[BF76] S. Bozinovski and A. Fulgosi. **The influence of pattern similarity and transfer learning upon training of a base perceptron b2**. In: 1976 (see page 4).

[BHP02] S. Bourassa, M. Hoesli, and V. Peng. **Do Housing Submarkets Really Matter?** (2002) (see page 12).

[Bou+99] S. C. Bourassa, F. Hamelink, M. Hoesli, and B. D. MacGregor. **Defining Housing Submarkets** (1999) (see page 3).

[BS14] R. Bro and A. K. Smilde. **Principal component analysis** (2014) (see page 22).

[BWV06] J. Bacher, K. Wenzig, and M. Vogler. **SPSS TwoStep Clustering – A First Evaluation**. In: 2006 (see page 3).

[Che+07] Z. Chen, S. Cho, N. Poudyal, and R. K. Roberts. **Forecasting Housing Prices under Different Submarket Assumptions** (2007) (see pages 3, 12, 33).

[Don+15] J. Dong, X. Li, W. Li, and Z. Dong. **Segmentation of Chinese Urban Real Estate Market: A Demand-Supply Distribution Perspective** (2015) (see page 12).

[FGM00] M. Fletcher, P. Gallimore, and J. Mangan. **The modelling of housing submarkets** (2000) (see page 3).

[GLS11] I. Gilboa, O. Lieberman, and D. Schmeidler. **A similarity-based approach to prediction** (2011) (see page 15).

[GZL08] J. Guan, J. Zurada, and A. Levitan. **An Adaptive Neuro-Fuzzy Inference System Based Approach to Real Estate Property Assessment** (2008) (see page 3).

[Hay06] K. Hayles. **The use of GIS and cluster analysis to enhance property valuation modelling in Rural Victoria** (2006) (see pages 1, 3).

[KHH02] T. Kauko, P. Hooimeijer, and J. Hakfoort. **Capturing Housing Market Segmentation: An Alternative Approach based on Neural Network Modelling** (2002) (see page 3).

[Kry07] M. Kryvobokov. **What location attributes are the most important for market value?: Extraction of attributes from regression models** (2007) (see page 11).

[Lee+17] K. Lee, K. Kim, J. Kang, S. Choi, Y. Im, Y. Lee, and Y. Lim. **Comparison and Analysis of Linear Regression & Artificial Neural Network** (2017) (see page 12).

[LIF19] Ltd. LIFULL Co. *LIFULL HOME'S Data Set*. 2019 (see page 19).

[Mal+18] A. Malinowski, M. Piwowarczyk, Z. Telec, B. Trawiński, O. Kempa, and T. Lasota. "An Approach to Property Valuation Based on Market Segmentation with Crisp and Fuzzy Clustering." In: 2018 (see page 3).

[Mos+20] M. Moshkovitz, S. Dasgupta, C. Rashtchian, and N. Frost. **Explainable k-Means and k-Medians Clustering**. In: 2020 (see page 16).

[Org] Japan External Trade Organization. *Comparison - Regions in Japan* (see page 29).

[Rou87] P. Rousseeuw. **Silhouettes: a graphical aid to the interpretation and validation of cluster analysis** (1987) (see page 8).

[Shi+15] D. Shi, J. Guan, J. Zurada, and A. S. Levitan. **An Innovative Clustering Approach to Market Segmentation for Improved Price Prediction** (2015) (see page 3).

[Shi+19] A. Shinde, N. Dange, N. Patane, S. Gholap, and V. Beera. **Real Estate Properties Assessment Using Deep Neural Network** (2019) (see page 12).

[Str74] M. Straszheim. **Hedonic Estimation of Housing Market Prices: A Further Comment** (1974) (see page 3).

[TPH11] S. Trivedi, Z. A. Pardos, and N. T. Heffernan. In: *Proceedings of the International Conference on Artificial Intelligence in Education*. 2011 (see pages 1, 3, 11).

[TPH15] S. Trivedi, Z. A. Pardos, and N. T. Heffernan. **The Utility of Clustering in Prediction Tasks** (2015) (see pages 11, 12, 26).

[WS12] C. Wu and R. Sharma. **Housing submarket classification: The role of spatial contiguity** (2012) (see pages 1, 3).

[Yan+09] Q. Yang, Y. Chen, G. Xue, W. Dai, and Y. Yu. **Heterogeneous transfer learning for image clustering via the social web**. In: 2009 (see page 4).

[Yu+20] Y. Yu, V. Jindal, I. Yen, F. Bastani, J. Xu, and P. Garraghan. **Integrating clustering and regression for workload estimation in the cloud** (2020) (see page 3).

[Zha03] B. Zhang. **Regression clustering** (2003) (see page 3).

[Zhu+20] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. *A Comprehensive Survey on Transfer Learning*. 2020 (see page 7).

[ZLG11] J. Zurada, A. Levitan, and J. Guan. **A Comparison of Regression and Artificial Intelligence Methods in a Mass Appraisal Context** (2011) (see page 4).

Declaration of Authorship

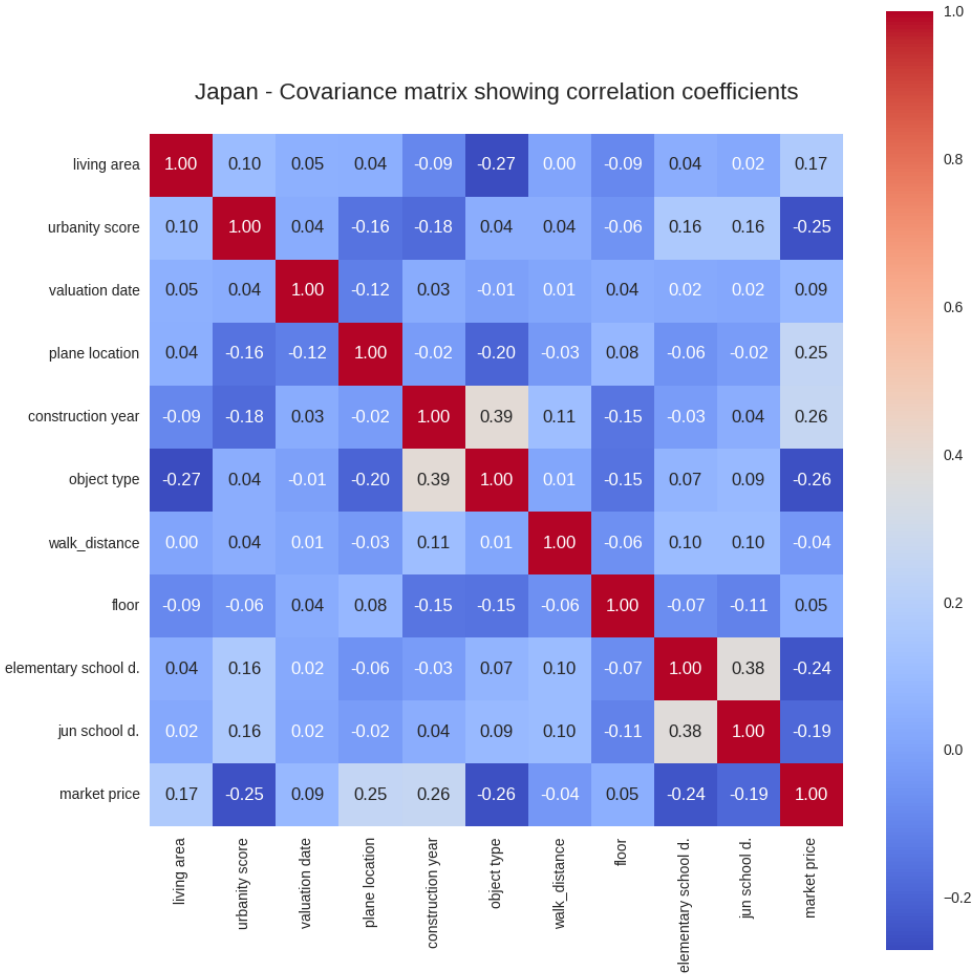
I hereby declare that this thesis is my own unaided work. All direct or indirect sources used are acknowledged as references.

Potsdam, June 28, 2022

Kathrin Thenhausen

A Appendix

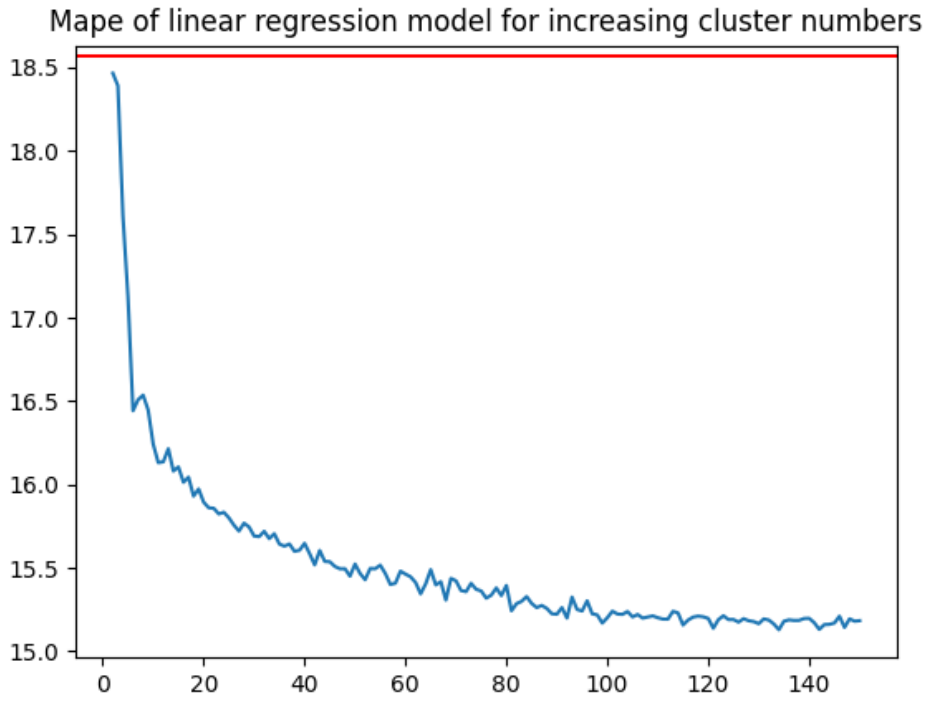
Figure A.1: Covariance matrix for Japan. The influences on the market price are noted in the last row/column.



We provide the covariance matrix of the whole Japanese real estate data set in Figure A.1. As the data set is much bigger than the parts containing properties

in Saitama only, the features have a higher variance and the matrix is less good understandable. Though, we notice the high importance of the object type, the urbanity score and the school distances.

Figure A.2: The behaviour of linear regression prediction models on clustering. The blue marked MAPEs of the prediction models trained on clusterings get lower for increasing numbers of clusters. The MAPE of a reference linear regression model is sketched in red.



In additional experiments we evaluated the metrics obtained by the choice of linear regression models for the construct on a prediction model as explained in Figure 4.1 (a). Figure A.2 shows the positive effects of increasing the number of clusters on the linear regression models, previously discussed in Section 6.4.1.