Uncertainty quantification for complex systems : application to the study of cities

Steentoft, Aike Alexander

2021

Steentoft, A. A. (2021). Uncertainty quantification for complex systems : application to the study of cities. Doctoral thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/152120>

<https://hdl.handle.net/10356/152120>

<https://doi.org/10.32657/10356/152120>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0
International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 17:10:31 SGT

Uncertainty quantification for complex systems - Application to the study of cities

Aike Alexander Steentoft

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

9. November 2020

.....

Date

.....

Aike Alexander Steentoft

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

9. November 2020

.....

Date

.....

Asst Prof (Adj) Markus Schläpfer

Authorship Attribution Statement

This thesis contains material from 1 paper published in the following peer-reviewed journal in which I am listed as an author.

Chapter 4 is published as Aike A Steentoft, Ate Poorthuis, Bu-Sung Lee, and Markus Schläpfer. The canary in the city: indicator groups as predictors of local rent increases. EPJ Data Science, 7(1):21, 2018. 6, 69

The contributions of the co-authors are as follows:

- I defined the initial project direction together with Asst Prof (Adj) Markus Schläpfer.
- I designed the study together with Asst Prof (Adj) Markus Schläpfer and Asst Prof Ate Poorthuis.
- Asst Prof Ate Poorthuis provided the Twitter data.
- I collected the census data.
- I performed the entire data analysis.
- I discussed the results with Asst Prof (Adj) Markus Schläpfer, Asst Prof Ate Poorthuis and Assoc Prof Bu Sung Lee.
- I prepared the manuscript drafts.
- I edited the manuscript drafts together with Asst Prof (Adj) Markus Schläpfer and Asst Prof Ate Poorthuis.

9. November 2020

.....

Date

.....

Aike Alexander Steentoft

Acknowledgements

Firstly, I would like to deeply thank my supervisor Asst Prof (Adj) Markus Schläpfer for his continuous academic and personal guidance, support, and trust. He was not only a great academic mentor but continuously supported me at all times.

I would like to thank my co-supervisor Assoc Prof Bu Sung Lee for his great support and for making me feel home at the Nanyang Technological University.

I would like to thank the Singapore-ETH Centre, the ETH Zurich and Singapore's National Research Foundation for making this thesis possible.

I would like to thank Prof Gerhard Schmitt for introducing me to the Singapore-ETH Centre and for being a continuous supporter, Thomas Rufener for always supporting me, and Dr Geri Schrotter and Fabian Clavier for being great team leaders.

I would like to thank Prof Luis Bettencourt and Dani Zünd for a great research stay in Chicago and Asst Prof Ate Poorthuis for being a great collaborator.

I would like to thank Lukas, Numa, Tibo, and many more colleagues from the Singapore-ETH Centre for a great and inspiring time in Singapore. Finally, I would like to thank my family for always being there for me.

“I am always looking, like a child, for wonders I know I am going to find. Maybe not every time, but every once in a while.”

—Feynman, Richard

To my dear family

Abstract

Human interactions are the fundamental drivers that establish cities as the hubs for innovation and economic growth. Traditional urban planning methods – aimed to facilitate these interactions – are however becoming questionable as they are typically based on ‘best practice’, thus unable to cope with the increasing complexity of today’s rapidly evolving cities. The quantitative description of interactivity and its relation to the spatial organization of urban spaces and transportation infrastructure is therefore of fundamental importance.

In the first part of the thesis I ask the – more obvious – question: ‘What methods can help us to gain additional insights into how human interactivity is shaped by the spatial organization of urban spaces?’; here human interactivity is treated as the dependent variable. Unfortunately, the existing spatial interaction models are unable to accurately capture empirical observations; indicating that relevant structure, present in the real world system, is not modeled. More recently machine learning systems have been developed that could overcome these challenges. But more flexible models such as deep neural networks are not interpretable and therefore not as accepted by urban planners and transportation engineers. From a statistical perspective, the main difficulty is to develop a model that (i) allows for many, potentially relevant, input variables with complex relational structure and (ii) is interpretable with respect to the relevance of the inputs and is interpretable with respect to the confidence of its predictions. I address this problem with Bayesian neural networks. The expressiveness and flexibility of neural network models allows to automatically identify complex structure in data. A Bayesian approach allows to model uncertainty in a principled way and is also helpful to automatically determine the relevance of input variables. Bayesian learning in neural networks is however challenging. First, one needs to define a prior probability distribution over the network parameters that encodes prior knowledge of the problem. Second, one needs to integrate function values of interest with respect to the high-dimensional updated prior distribution. I address these problems with

a new neural network model which offers both the flexibility of a deep neural network and the interpretability in terms of variable selection. In particular, I use the recently derived exact equivalence between infinitely wide deep neural networks and Gaussian processes to do exact Bayesian inference on the deep neural networks. I then introduce additional hyperparameters to the covariance function of the Gaussian process which enable automatic feature selection. Sparsity is handled via the horseshoe prior. I tested various implementations of network architectures and inference procedures on synthetic data sets and a real data set related to human interactivity. I found that my proposed approach can give more accurate predictions than traditional models while providing uncertainty estimates for its predictions and also providing additional insights into the importance of different urban measurements. Such insights can be of great value for urban planning – aimed to facilitate interactions.

In the second part of the thesis I asks the – more subtle – question: ‘What methods can help us to gain additional insights into how the spatial organization of urban spaces is shaped by human interactivity?’; here the spatial organization of urban spaces is treated as the dependent variable. More specifically, I address the problem of predicting from human interactivity patterns when and where urban neighborhoods will face urban change. To tackle this challenge, I adapt the concept of ‘bioindicators’, borrowed from ecology, to the urban context. The objective is to use an ‘indicator group’ of people to assess the quality of a complex environment and its changes over time. Specifically, I analyze millions of geolocated Twitter records across multiple US cities, allowing me to derive socio-economic user profiles based on individual movement patterns. As a proof-of-concept, I define users with a ‘high-income-profile’ as an indicator group and show that their visitation patterns are a suitable indicator for expected future rent increases in different neighborhoods. The concept of indicator groups highlights the potential of closely monitoring only a specific subset of the population, rather than the population as a whole. If the indicator group is defined appropriately for the phenomenon of interest, this approach can yield early predictions while simultaneously reducing the amount of data that needs to be collected and analyzed.

Contents

Acknowledgements	ix
Abstract	xiii
List of Figures	xvii
List of Tables	xxi
1 Introduction	1
1.1 Background and motivation	1
1.2 Problem statement	2
1.3 Objectives and value added	4
1.4 Outline of the thesis	5
I Predicting human interactivity	9
2 Uncertainty quantification for complex systems	11
2.1 Introduction	11
2.2 Related work	13
2.2.1 Bayesian statistics	14
2.2.2 Gaussian processes	15
2.2.3 Neural networks	17
2.2.4 Bayesian learning of neural networks	19
2.2.5 Sparse learning for variable selection	24
2.2.6 Markov chain Monte Carlo methods	29
2.3 Contributions	33
2.4 Proposed Method	34
2.4.1 Notation	36
2.4.2 Gaussian Processes and Deep Neural Networks	37
2.4.3 Automatic Feature Selection with Sparsity Inducing Prior	39
2.4.4 Bayesian Learning of Neural Networks Using Gaussian Process Priors	40
2.5 Experiments	41

2.5.1	Data sets	43
2.5.2	Model configurations	46
2.5.3	Inference configurations	46
2.5.4	Results	48
2.6	Summary	52
3	Application to spatial interaction modeling	53
3.1	Introduction	53
3.2	Related work	56
3.2.1	Spatial interaction models	56
3.2.2	Machine learning models	60
3.3	Contributions	62
3.4	Experiments	64
3.4.1	Data sets	64
3.4.2	Model configurations	69
3.4.3	Inference configurations	69
3.4.4	Results	70
3.5	Summary	76
II	Predicting neighborhood change	81
4	Spatial interactions as predictor of neighborhood change	83
4.1	Introduction	83
4.2	Related work	87
4.2.1	Urban change and Housing	87
4.2.2	Social sensing	88
4.2.3	Indicator group	89
4.3	Contributions	89
4.4	Method	90
4.5	Experiments	91
4.5.1	Data sets	91
4.5.2	Selection of the indicator group	93
4.5.3	Predicting neighborhood change	96
III	Conclusion and future directions	103
5	Conclusion and future directions	105
	List of Author's Publications	111
	Bibliography	113

List of Figures

2.1	(A) Correlation matrix for the informative, redundant, and repeated variables for the small Madelon data set. (B) Correlation matrix for the informative, and redundant variables for the large Madelon data set.	45
2.2	Heat map of the data matrix where rows correspond to data samples and columns correspond to variables. The last column corresponds to the class label. The color intensity represents the magnitude of the coefficient and the color represents the sign. (A) For the informative, redundant, repeated, and noninformative variables for the small Madelon data set. (B) For the informative, redundant, and repeated variables for the large Madelon data set.	45
2.3	Same matrix as in Figure 2.2 but rows and columns have been randomly shuffled and rows have been grouped per class.	46
2.4	(A) The red dots show MCMC samples, and the blue line and band show the mean and $\pm$ standard deviation for $\theta_3^* = \log \tau$. (B) The dots show the autocorrelations of the same Markov chain for different lags.	47
2.5	MCMC samples for all local shrinkage parameters λ_i with $i = 1, \dots, D$ for a selected model configuration for the small Madelon data set. .	48
2.6	Trajectories of Hamiltonian energy H , potential energy U , and kinetic energy K for one integration run for a selected model configuration for the small Madelon data set. The trajectory was initialized at the MAP and is specified by integration length T_* and step size ϵ . .	49
2.7	Test points are sorted according to the predicted mean squared error (MSE) and then binned. The predicted MSE on the x-axis is the mean of the predicted variance for a bin and the y-axis is the actual MSE for the test points in that bin, (A) for the small Madelon data set with bin size 10, (B) for the large Madelon data set with bin size 50.	51
2.8	The blue line shows the posterior mean value of the local shrinkage parameters λ that determine the shrinkage of the connected variables, the colored dots show which of the variables are true informative, redundant, and repeated, (A) for a selected model for the small Madelon data set, (B) for a selected model for the large Madelon data set.	52

3.1	Spatial mapping of binned and randomly selected OD flows for Taxi trips between New York City's census tracts in 2014 and 2015. The black dots represent New York City's census tracts and the colored lines represent the OD flow between them (the legend refers to the frequency bins).	68
3.2	Probability distribution of all OD flows for Taxi trips between New York City's census tracts in 2014 and 2015.	68
3.3	Spatial mapping of (A) the number of low income jobs living in each census tract, (B) the number of high income jobs living in each census tract, and (C) the number of amenities in each census tract. The black dots represent New York City's census tracts and brighter colors correspond to higher numbers.	69
3.4	The maps show the absolute percentage errors of the predicted flows in New York City for the binned test data. The errors are shown for flows occurring between 11pm - 1am (first row), 7am - 9am (second row), 3pm - 5pm (third row) and for the H-NNGP with zero hidden layers/linear Gravity model (first column), one hidden layers (second column), and three hidden layers (third column). The colormap is normalized within each row and brighter colors indicate larger errors.	72
3.5	Correlation plots of predicted mean squared error (MSE) versus actual MSE for the binned test data of flows in New York City. Means are computed by averaging over 50 data points. The correlation plots are shown for flows occurring between 11pm - 1am (first row), 7am - 9am (second row), 3pm - 5pm (third row) and for the H-NNGP with zero hidden layers/linear Gravity model (first column), one hidden layers (second column), and three hidden layers (third column).	74
3.6	Feature relevance for predicting flows that occurred between 11pm - 1am (A), 7am - 9am (B), and 3pm - 5pm (C) for the linear model with zero hidden layers (H-NNGP-0, equivalent to the linear Gravity model), the non-linear model with one hidden layer (H-NNGP-1) and the non-linear model with three hidden layers (H-NNGP-3). The detailed description of each feature is given in Table 3.1.	77
3.7	Correlation pairplot with all input variables for trips happening in the morning hours (7am - 9am) and plotted for 50k data points. Brighter colors indicate a higher number of trips.	78
3.8	Origin-destination maps with origins separated into three chunks (row 1, row 2, row 3) according to the number of high-income residents. The left column shows the origins and the right column shows the respective destinations. The colors represent the number of connected origins/destinations and thereby indicate the centrality of each location. Brighter colors correspond to a higher number of connected origins/destinations.	79

4.1	Model for monitoring indicator group. 2-mode network model that consists of a set of user nodes in the top layer (indicator users, green), a set of neighborhood nodes in the bottom layer, and a set of links representing the visits of users to neighborhoods. Adapted from [1].	91
4.2	Selecting an indicator group. a , Income-profile distribution for users in Chicago. I assign an income profile to each user based on the census median income of the visited neighborhoods. The 10 percent ‘richest’ users represent the indicator group (green) while the 10 percent ‘poorest’ users function as a reference group (red). b , For each neighborhood, I compare the number of unique indicator visitors (green) and the number of unique reference visitors (red) in 2012/2013 with the neighborhood’s median rent prices in 2012. Neighborhoods that bring together more indicator visitors show higher rent prices while more reference visitors indicate lower rent prices.	96
4.3	Comparing the predictions of the indicator group and the reference group. a , For each neighborhood bin in Chicago, I compare the number of unique indicator visitors (green) and the number of reference visitors (red) in 2012/2013 with the 2012-2015 change of the census median rent. Neighborhoods with more indicator visitors tend to experience a higher rent increase. b-c , Mapping the rent price predictions (a) for the indicator group, and d-e , for the reference group.	98
4.4	Robustness of the indicator group across multiple cities. Comparing the rent price predictions of the indicator group and the reference group for New York City (indicator group: $r_s = 0.70$; $p < 10^{-4}$; $N = 25$, reference group: $r_s = 0.46$; $p = 0.03$; $N = 22$), Los Angeles (indicator group: $r_s = 0.73$; $p < 10^{-4}$; $N = 23$), Boston (indicator group: $r_s = 0.57$; $p < 10^{-2}$; $N = 20$) and Portland (indicator group: $r_s = 0.67$; $p < 10^{-2}$; $N = 14$). The correlation values for the reference group in Los Angeles, Boston and Portland are not significant ($p > 0.05$).	99

4.5	Transforming neighborhoods. a , For each neighborhood in Chicago, I compare the number of unique indicator visitors in 2012/2013 with the census median rent in 2012; the neighborhood bins (green) generate a baseline (corresponding to Figure 4.2b). Neighborhoods below (above) the baseline show a disproportionately high (low) rent compare to other neighborhoods in the same bin. With Pilsen, Logan Square, and Lake View, I show three neighborhoods in different transformation stages. b , Mapping the level of disproportion with the Visitor-Rent-Index (VRI) (on the rise, $VRI < 0$). Here, I removed neighborhoods with less than 10 indicator visitors from the map. c , I compare the rent increase (2012-2015) in neighborhoods with a low VRI (first quartile) to neighborhoods with a high VRI (last quartile). Neighborhoods with a low VRI show significantly higher rent increase (notches indicate 95% confidence interval around the medians).	100
4.6	Upgrading the indicator group. a , Income-age-profile distribution for users in Chicago. I assign an income-age profile to each user based on the census median income and median age of the visited neighborhoods (darker colors indicate a higher frequency). The 10 percent ‘youngest’ of the ‘richest’ users represent the upgraded indicator group (green) while all ‘richest’ users represent the initial indicator group. b , For each neighborhood bin in Chicago, I compare the number of upgraded indicator visitors (green) and the number of initial indicator visitors (red) in 2012/2013 with the 2012-2015 change of the census median rent. Using visitors with a high-income-low-age profile as indicator group yields a stronger correlation than using all high-income-profile visitors (upgraded: $r_s = 0.78$; $p < 10^{-3}$; $N = 15$, initial: $r_s = 0.70$; $p < 10^{-3}$; $N = 22$).	102

List of Tables

2.1	Parameter choices for the small Madelon data set and the large Madelon data set.	44
2.2	Test accuracy for various model configurations for the small and large Madelon data set. The Gaussian process models are specified by (L, σ_g^2)	50
2.3	CPU time for various model configurations for the small and large Madelon data set. The Gaussian process models differ in the number of layers. Inference was performed with 10000 samples with 10 leapfrog steps each. Prediction was performed with 50000 samples. For the computations a 2.4 GHz 8-Core Intel Core i9 was used and the reported CPU times are summed over all cores (and thereby represent the time a single core would have needed).	51
3.1	Location attributes related to attraction of trips, production of trips, and interaction between origin and destination.	66
3.2	Prediction performance results for different times of the day, data bins, and model configurations. MSE is rounded to integers, MAE is rounded to the first decimal place, and MAPE and CPC are rounded to the the second decimal place. Standard errors are shown in brackets and the winning model configurations are highlighted in bold.	80
4.1	Network statistics for indicator group.	95

Chapter 1

Introduction

1.1 Background and motivation

Human interactions are the fundamental drivers that establish cities as the hubs for innovation and economic growth [2], [3, p. 15], [4, p. 239]. At the microscale, the daily interactivity of individuals is considered an accelerator of social and cultural and small-scale entrepreneurial activity [5, p. 10]. At the macroscale, connecting population groups is considered a precondition for innovation and lies at the heart of agglomeration forces in many spatial theories of economic growth [6].

The way people interact in the city is directly linked to the *spatial organization* of urban spaces and transportation infrastructures. One can describe the system for instance by analogy to an electrical network with voltages on nodes and currents on edges. In this analogy, the voltages on nodes correspond to attraction potentials of urban spaces, the currents on edges correspond to people flowing between urban spaces, and the electrical resistance corresponds to the efficiency of the transportation infrastructures. The level of interactivity then depends on if cities provide *attractive* urban spaces and *efficient* transportation infrastructures.

The rapid growth and changing dynamics of cities requires a continuous adaptation of the system. Importantly, this can lead to a complex feedback loop when the *present* interactivity informs the planning of *new* spaces and infrastructures. Planners and engineers are ever more challenged by this complexity of cities. The

stakes are high as decisions affect many people, cost lots of resources, and are difficult to reverse. The *quantitative* description of interactivity and its relation to the spatial organization of urban spaces and transportation infrastructures is therefore of fundamental importance.

Spatial interaction models make an important class of models that describe the aggregate flow of many individuals from a set of origins to a set of destinations. The flow is described as a function of a few variables of the locations considered, such as mutual distance and population levels [7]. Two main approaches have emerged over the years. *Gravity models* assume that the flow between two locations is a decreasing function of their distance [8]. *Intervening opportunities models* assume that the flow between two locations is determined by the number of potential destinations between them [9]. Another important class of models are Land Use and Transport Integrated models (LUTIs) that explicitly formulate the two-way interaction between land use and transportation infrastructure within an agent-based framework. Most of the agent-based models described in the literature are calibrated on specific regions; popular examples are UrbanSim [10] and ILUTE [11]. A good overview over existing LUTI models can be found in [12].

Importantly, human interactivity is not only a function of urban spaces but urban spaces are also shaped by the interactivity (e.g. in the case of gentrification, the transformation of neighborhoods is typically preceded by an influx of the creative class). The studies that focus on the latter try to understand and model neighborhood change as a process, often using ecological concepts such as the invasion-succession model [13, 14].

1.2 Problem statement

The first problem statement relates to the existing spatial interaction models. From the vast palette of mobility models, I am particularly interested in spatial interaction models which aim to predict aggregated mobility flows (number of trips) based on the spatial distribution and socio-economic characteristics of locations [7]. Such models take an important stage in times where data privacy often won't allow to develop models that are dependent on individual mobility trajectories. Models that are able to describe the complex mobility patterns of today's cities solely

based on aggregated mobility data are therefore of great value for urban planning. However, with the increasing complexity of cities, the existing spatial interaction models are often not able to accurately capture empirical observations in particular at fine-grained spatial scales [15]. The *high inaccuracy* of the estimator indicates that relevant relations between explanatory variables and outcome predictions are missed. At the same time, decisions that are based on such models can be very critical, expensive, and non-reversible. Consider e.g. the zoning decision to allow for more shopping space at a certain location in order to attract more people. In this context, decisions should make use of all available information instead of being constrained by too simplistic models.

To that end, a degree of automated data analysis could assist urban experts in making best informed decisions. The success of machine learning in various real-world applications [16] suggests that it may also be helpful to unlock the full potential of urban data. In particular, in the field of urban mobility with its high data density and complexity of the problem it seems ideal to apply machine learning to assist decision making.

Perhaps the most promising approach to achieve high prediction accuracy are deep neural networks. Deep neural networks have been deployed in several complex real-world tasks such as object recognition [17], speech recognition [18], image segmentation [19], and machine translation [20]. The non-linearity of neural networks allows them to capture complex non-linear dependencies, which often result in high predictive performance. But neural networks often function as black box and lack interpretability. For example, after training a deep neural network it is difficult to obtain uncertainty estimates of its predictions and it is difficult to determine which input features are relevant for the prediction [21]. This is highly undesirable for urban experts; making critical, expensive, non-reversible decisions without being able to justify them to experts and the public. As a consequence, deep neural networks are not applied in urban planning practice. Advancing the interpretability of deep neural networks is key towards increasing their impact in urban applications.

The second problem statement relates to the existing approaches that aim at understanding and predicting neighborhood change. Previous work has often evaluated the characteristics of the neighborhood itself: e.g., distance to employment centers, schools, restaurants and other amenities, crime level, and a wide array of demographic data such as income, education and age of its residents [22]. However,

collecting such data is resource-intensive, which often results in data only collected decennially or, in a best case, annually. This limits the applicability in analysis of urban processes that operate and change on a much finer temporal scale, such as the early prediction of rising housing prices in urban neighborhoods. While recent research has been increasingly looking at monitoring the city in near real-time by making use of new types of user-generated data [1], the link between collecting and monitoring this type of data and its actual use for predicting future changes in urban social processes remains understudied.

1.3 Objectives and value added

The first objectives relate to the first problem statement. In this thesis, I propose an interpretable deep neural network approach to spatial interaction modeling. Specifically, the objectives are:

- The thesis develops an interpretable deep neural network approach to spatial interaction modeling. Interpretability is defined as (i) providing uncertainty estimates of its predictions, and (ii) automatically determining the relevance of the input variables.
- The thesis applies the developed machine learning approach to real data to test if the developed method can provide more detailed insights compared to more traditional methods.

The developed models and new insights can greatly benefit urban planning. Traditional urban planning methods that aim to facilitate spatial interactions are becoming questionable as they are often based on rules of thumb [23, p. 2] or over-simplified models with limited explanatory power, thus unable to cope with the increasing complexity of today’s rapidly evolving cities. In order to effectively support urban planning it requires (i) to develop more accurate and less over-simplified models that can handle the complexity of today’s cities and (ii) to make these models more accessible to urban planners. In this thesis, I will only tackle the first challenge but it is important to keep in mind that the second challenge is just as important if the models will be applied in practice. The proposed machine learning approach overcomes some of the limitations by *automatically* learning the

relations between interactivity patterns and *relevant* location attributes. The predictions, uncertainty estimates, and insights can help urban experts with making intervention decisions and communicating to public. For example, to predict how the zoning decision of more commercial space in a given neighborhood will affect the existing origin-destination patterns. And more importantly, insights into which urban measures are relevant for the prediction. This is an important advance because the model interpretability determines if the model predictions will be accepted by experts and the public. Another important factor that determines the applicability in practice is the explainability of the models. Explainability describes how well the internal mechanics of a model can be described in human terms. There is lots of recent work on the subject of explanatory artificial intelligence [24] and it will be an important and interesting subject for future work.

The second objective relates to the second problem statement. In this thesis, I propose a methodology to better understand and predict neighborhood change. Specifically, the objective is:

- The thesis develops a methodology to monitor urban spaces and predict neighborhood change based on the characteristics of its visitors. New types of social sensing data enable a close to real-time monitoring.

The developed models and new insights allow for the development of a broad range of applications with relevance for various stakeholders from urban planning and governance to the real estate industry. The methodology can potentially be applied to a broad range of urban challenges, such as predicting rising rents, detecting social segregation processes or predicting spatio-temporal crime patterns in cities.

1.4 Outline of the thesis

In part I, the objective is to develop an interpretable deep learning approach to spatial interaction modeling.

Chapter 2 presents the methodological approach. Section 2.1 motivates for the use of a Bayesian approach to improve the interpretability of the neural network. The related work in this field is presented in Section 2.2. I review Bayesian statistics

in Subsection 2.2.1 and its application to neural network models in Subsection 2.2.3. Subsection 2.2.5 then introduces the concept of sparse Bayesian learning which will be important to automatically select the important input variables. As the inference method of choice I use Markov chain Monte Carlo which I review in Subsection 2.2.6. Section 2.3 summarizes my contributions. Section 2.4 presents the method in detail. In order to allow for exact Bayesian inference in deep neural networks using only matrix calculations, I use the recently developed exact equivalence between infinitely wide deep neural networks and Gaussian processes which I review in Subsection 2.4.2. In Subsection 2.4.3, I introduce automatic feature selection to the Gaussian kernel by using sparsity-inducing prior distributions on the corresponding neural network weights that connect inputs with the first hidden layer. In Subsection 2.4.4, I review how the resulting Gaussian process prior can be used to train the corresponding infinitely wide deep neural network. Section 2.5 presents various experiments on synthetic data sets. I test for various neural network architectures and inference procedures and compare the results.

In Chapter 3, the objective is to gain scientific understanding of what makes urban spaces attractive. Section 3.1 introduces the topic of spatial interaction modeling and its importance for urban planning. The related work in this field is presented in Section 3.2. In Subsection 3.2.1, I start with reviewing traditional models such as the Gravity models and Intervening opportunities models. Subsection 3.2.2 then reviews machine learning approaches that have been developed more recently. Section 3.3 states the added value of my deep neural network approach and summarizes my contributions. In general, I deploy the method developed in Chapter 2. The details of the implementation are presented in Section 2.4. As typical in spatial interaction modeling, the goal is to analyze the relation between a set of explanatory variables and the aggregated flow between a pair of locations, and ultimately to predict origin-destination matrices. In Section 3.4, I then test the developed methodology on a real data set (taxi ride data) and compare various model configurations. Importantly, the proposed approach can give more accurate predictions than traditional models and provides additional insights into the importance of urban measurements. This suggests that the model could help urban experts to make critical decisions.

In part II, the objective is to better understand and predict neighborhood change.

Chapter 4 presents the results previously published in [25]. Section 4.1 introduces the topic of neighborhood change and social sensing and motivates for studying the link between collecting and monitoring new types of social sensing data and its actual use for predicting future changes in urban social processes. Related work is presented in Section 4.2. I review the field of neighborhood change and housing in Subsection 4.2.1, Social Sensing in Subsection 4.2.2, and the ecological concept of indicator groups in Subsection 4.2.3. Section 4.3 summarizes my contributions and highlights the added value of our method. I developed a methodology to monitor neighborhoods based on the characteristics of its visitors, which allows to predict neighborhoods with expected rent increase in subsequent years. As the core idea, I combine the bioindicator concept, borrowed from ecology, with social sensing data from location-based social networks. The details are presented in Section 4.4. The significance of the developed method is tested on real data sets including Twitter data and census data. I find that the proposed method can be used for the early detection of rising rents in urban neighborhoods. This suggests that the adaptation of the concept of bioindicators to urban residents, monitored through social sensing data, is of great relevance to various stakeholders from urban planning and governance to the real estate industry.

Part I

Predicting human interactivity

Chapter 2

Uncertainty quantification for complex systems

2.1 Introduction

Modeling a phenomenon from data typically involves dealing with two types of uncertainties: (i) aleatoric uncertainties that describe the noise inherent in the observations, for example origin-destination flows might change due to seemingly random trip decisions of people such as where to eat on a certain day and (ii) epistemic uncertainties that describe the uncertainties in the model such as different parameter configurations and the chosen class of model itself, for example if a linear or non-linear model is used to model origin-destination flows [26]. Bayesian machine learning is the principled approach of dealing with these uncertainties when learning models from data [27]. Two steps are essential in Bayesian machine learning: (i) defining a probabilistic model, and (ii) inference of the model parameters from the observed data. The trained models then not only provide uncertainty estimates of the parameters but also of the predictions. Importantly, in the Bayesian paradigm there is a clear separation between learning a model from data and the risk associated with a prediction (in contrast to non-Bayesian modelling where models are typically trained with a risk function that penalizes errors between prediction and training data in a certain way, e.g. squared or absolute). This separation in the Bayesian paradigm is crucial in order to uncouple the modeling process from the decision making process. Finally, it has been proven that

models which provide uncertainty estimates of the predictions are extremely useful in critical applications in various disciplines such as medicine and autonomous vehicles [28], thus it is also promising to apply to urban planning. For example, the zoning decision to increase the amount of shopping area in a certain neighborhood might be intended at attracting more visitors. Such critical decisions should not rely on a single model estimate but should also take prediction uncertainties into account.

Deep learning models, on the other hand, are typically defined in a non-probabilistic way. They are deterministic functions where parameter values and predictions are point estimates rather than probability distributions. As a consequence, risks associated with the prediction must already be incorporated into the training process. There is no separation between training the model and the decision making process. The missing uncertainty estimates for predictions means that decision makers don't know when the model is uncertain about its predictions and might even predict on random. This is obviously highly undesirable for certain critical applications like the ones mentioned above. In less critical tasks however, deep learning models had great success [16, 29]. This can be explained with their prediction performance and the comparatively simple way of training the models.

Bayesian deep learning defines deep learning models in a probabilistic way with the goal to keep the best of both worlds [28]. The hope is to develop models that are not only as easy to train as existing deep learning models but that also provide uncertainty information. Such models are of great value for critical applications where uncertainty information are required and high prediction accuracy is desired. However, the main difficulty in Bayesian deep learning is the inference part since deep learning models typically have a large amount of parameters which leads to very high-dimensional, intractable probability distributions [28]. Bayesian computation is an active research field that tackles this problem where the two main directions can be summarized into (i) stochastic inference approaches such as Markov chain Monte Carlo (MCMC) methods, and (ii) deterministic inference approaches such as Variational Inference (VI) [30]. The former is characterized by its theoretical convergence to the true model but practical problems in fully exploring the distributions of a neural network and an expensive inference procedure. The latter is characterized by an efficient inference procedure but has the

drawback of only providing an approximate model without knowing how close the approximation is to the true model.

Recently, the exact equivalence between infinitely wide deep neural networks (which is a deterministic function) and Gaussian processes (which is a probability distribution over functions) has been derived [31]. This approach promises to overcome many of the challenges of the previous approaches by providing exact and efficient Bayesian inference. The main goal of this part of the thesis is to explore the potential of this approach for applications in urban planning.

As discussed above, uncertainty estimates help to interpret the *outputs* of a model. In this thesis, I additionally consider sparse learning as a measure to improve the interpretability of the *inputs* to the model. In the typical sparse learning setting, the model allows for many covariates that are potentially relevant for prediction. Sparsity then has the underlying assumption that only a subset of these covariates is relevant but we don't know beforehand which of the covariates are relevant. By selecting only the relevant variables, sparse learning can help to better understand the process that generated the data, can reduce measurement requirements, and can improve prediction performance especially if the number of data points is small compared to the number of covariates. One approach for sparse learning is to use filters that rank covariates according to correlation coefficients but the limitation is that variable dependencies are mainly ignored.

In this part of the thesis, I develop an embedded variable selection method for the “neural network as Gaussian process” approach discussed above. I then evaluate the proposed method on synthetic data and report prediction performance, uncertainty estimation performance, and variable selection performance. This part lays the foundation for Chapter 3, where I apply the method to real data and explore the potential of sparse learning in spatial interaction modeling where the number of potentially relevant covariates that describe origin-destination trip patterns is large and where relevant covariates vary according to the trip purpose.

2.2 Related work

In this section, I introduce and review work that is related to the proposed method in Section 2.4. I start with the fundamentals of Bayesian statistics which I use as

the statistical framework to formalize uncertainty. It follows an introduction to Gaussian processes, a specific type of Bayesian model that I use as my modelling framework. I then introduce neural networks from the traditional non-Bayesian perspective before I review (recent) approaches of Bayesian formulations of neural networks. It follows an introduction and a review to sparse learning for variable selection in non-Bayesian and Bayesian models. Finally, I introduce Markov chain Monte Carlo and specifically the Hamiltonian Monte Carlo algorithm that I will use for the inference part.

2.2.1 Bayesian statistics

In this thesis, I use Bayesian statistic as the statistical framework to formalize model uncertainty. A great review on the topic can be found in [30]. Given a set of N input-target pairs $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, the main goal of Bayesian regression is to find the parameters θ of the function $f(x; \theta)$ that are *likely* to have generated the targets $\mathbf{y}$. In the following, I briefly review the two main parts of Bayesian inference, (i) setting up a full probability model, and (ii) conditioning the model on observed data.

The first step of setting up a full probability model is defining a prior model which means defining a prior distribution $p(\theta)$ over the parameter space of θ . This prior distribution quantifies all relevant domain expertise and is available before a measurement is made. While often being criticized for being arbitrary, the prior distribution allows to encode domain expertise that can be worth thousands of observations. The second step is defining an observational model which means defining a likelihood function $p(\mathbf{y} \mid \theta, \mathbf{x})$, for example, the Gaussian likelihood function is $p(\mathbf{y} \mid \theta, \mathbf{x}) = \mathcal{N}(\mathbf{y}; f(\mathbf{x}; \theta), \sigma_n^2 I)$, where σ_n^2 is the noise variance. The likelihood function quantifies which model configurations are consistent with the seen data, and is used to update the prior distribution. The posterior distribution $p(\theta \mid \mathbf{y}, \mathbf{x})$ can be constructed with Bayes' theorem as

$$p(\theta \mid \mathbf{y}, \mathbf{x}) = \frac{p(\theta)p(\mathbf{y} \mid \theta, \mathbf{x})}{p(\mathbf{y} \mid \mathbf{x})}.$$

It aggregates what was known before and what was learned into what is known after the measurement.

Once the model is specified, we can condition the model on observed data. However, our goal is not to construct the full posterior distribution but instead we are trying to extract information out of it where the only way of extracting information out of a distribution is by computing expectation values. To that end, Bayesian computation is concerned with computing the posterior expectation values

$$\mathbb{E}[g(\theta)] = \int g(\theta)p(\theta \mid \mathbf{y}, \mathbf{x})d\theta,$$

where $g(\theta)$ is some function over the parameter space that encodes a specific question about the system, such as means, quantiles, and variances. For example, we can set $g(\theta) = f(x; \theta)$ to compute the prediction mean.

The described two-step process of Bayesian statistics is conceptually very elegant; the entire process consists of building the model and then computing expectation values. However, computing the expectation values for a reasonable complex model is very difficult in practice. The main difficulty is computing the marginal likelihood which is the integral of the likelihood times the prior over the entire parameter space

$$p(\mathbf{y} \mid \mathbf{x}) = \int p(\mathbf{y} \mid \theta, \mathbf{x})p(\theta).$$

For simple models, this marginalization can be done analytically as long as the likelihood is conjugate to the prior. However, this is not the case for reasonably complex models and the posterior distribution $p(\theta \mid \mathbf{y}, \mathbf{x})$ becomes intractable. In such cases we need approximations or sampling methods.

2.2.2 Gaussian processes

A stochastic process is the generalization of a probability distribution over a finite-dimensional random variable to a probability distribution over functions. In this context, a Gaussian process is any distribution over functions such that any finite set of function values have a joint Gaussian distribution. A great review on the topic can be found in [32]. In the language of Bayesian statistics, the prior model (before conditioning on data) of a Gaussian process is specified by a mean function $\mathbb{E}[f(x)] = \mu(x)$ and a covariance function (kernel) $\text{cov}[f(x), f(x_*)] = k(x, x_*)$. The mean function is typically set to zero since the kernel can capture such structure as well. The kernel determines which structure in the data can be captured by the

model, e.g. there is a corresponding kernel for linear regression. There are many other choices of kernels and the main difficulty becomes to construct a kernel that represents a particular structure.

A great property of Gaussian processes is that the marginal likelihood for a particular model and data can be computed analytically. This property allows to compare models with different kernels and configurations of kernel parameters and eventually select the best one. The marginal likelihood is the integral over the likelihood times the prior $p(\mathbf{y} \mid \mathbf{x}) = \int p(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) p(\mathbf{z} \mid \mathbf{x}) d\mathbf{z}$, where $\mathbf{z} = [f(x_1), f(x_2), \dots, f(x_N)]$. The term marginal comes from the fact that we marginalize over function value z_* for all locations $\mathbf{x}_*$ where there is no data. The prior is Gaussian $\mathbf{z} \mid \mathbf{x} \sim \mathcal{N}(0, k(\mathbf{x}, \mathbf{x}))$ and the likelihood is Gaussian as well $\mathbf{y} \mid \mathbf{z} \sim \mathcal{N}(\mathbf{z}, \sigma_n^2 I)$. Prior and likelihood are therefore conjugate and we can compute the log marginal likelihood as

$$\log p(\mathbf{y} \mid \mathbf{x}) = -\frac{1}{2} \mathbf{y}^T (k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I| - \frac{N}{2} \log 2\pi \quad (2.1)$$

The best model can then be found by finding the kernel and the hyperparameter configuration of the kernel that maximize the log marginal likelihood.

We can use the selected model to make predictions for any location x_* conditioned on the observed data $\mathbf{x}$. The predictive distribution for the value $z_* = f(x_*)$ is a conditional multivariate Gaussian and can be computed analytically by

$$\begin{aligned} p(z_* \mid \mathbf{x}, \mathbf{y}, x_*) &= \mathcal{N}(\bar{z}_*, \text{cov}(z_*, z_*)) \\ \bar{z}_* &= k(x_*, \mathbf{x}) (k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I)^{-1} \mathbf{y} \\ \text{cov}(z_*, z_*) &= k(x_*, x_*) - k(x_*, \mathbf{x}) (k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I)^{-1} k(\mathbf{x}, x_*) \end{aligned} \quad (2.2)$$

Sampling functions from the prior or the posterior then only means drawing from multivariate Gaussian distributions.

Gaussian processes have a few very useful properties. First, we can compute the exact marginal likelihood which provides a principled way of comparing models and eventually selecting the best model. Second, we can compute the exact predictive distribution at test points which is simply a multivariate Gaussian distribution and can be easily combined with decision procedures.

Gaussian processes also have a few limiting properties. First, inference time scales with $\mathcal{O}(N^3)$ which makes exact inference only feasible up to a few thousand data points. Beyond that approximate inference methods are available. Second, exact predictive distributions need to be Gaussian which means that for a non-Gaussian likelihood approximate inference has to be applied [32]. Third, there is the need to choose a kernel which typically is done by a human expert.

2.2.3 Neural networks

A detailed review on deep learning can be found for example in [16]. To introduce neural networks, I start with the most simple statistical tool of linear regression. We have given N pairs of inputs and targets $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where $x_i \in \mathbb{R}^D$ denotes the D -dimensional input and $y_i \in \mathbb{R}^T$ denotes the T -dimensional target. For example, in the case of housing price prediction, the inputs would be the features of the house and the neighborhood and the target would be the most recent selling price. Linear regression then assumes a linear function mapping $f(x)$ from inputs to targets and often includes a noise term: $z_i = w^T x_i + b + \epsilon_i$, where z_i is the predicted output, w^T is the transpose of the D -dimensional weight vector, $b \in \mathbb{R}$ is the bias and ϵ_i is the noise component. Different parameter settings of w and b result in different linear transformations from input to outputs. Typically, the parameters are determined by minimising a risk (loss) function such as the squared error $L = \frac{1}{N} \sum_{i=1}^N \|y_i - z_i\|_2^2$.

In the case where the relation between inputs and targets is not linear, we need some form of non-linear function mapping $f(x)$. A common approach is to use linear basis function regression. Here the input x_i is given to a number of K non-linear transformations $\phi_k(x)$ to get a feature vector $\phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_K(x)]$. Then linear regression is performed with the feature vector as input. The non-linear transformations are called basis functions and can be designed with e.g. polynomial bases x^k or Fourier bases with different frequencies. The basis functions are fixed and often chosen to be orthogonal. The goal is to find the optimal combination of basis functions that can explain the observed data.

An even more flexible approach is to use parameterized basis functions which means that the basis functions are not fixed and orthogonal anymore. For example, we could first apply K different linear transformations to the inputs, $w_k^T x + b_k$, and

then apply the same non-linear transformation to each of the outputs, $\phi_k(x) = \sin(w_k^T x + b_k)$. Then again, linear regression is performed with the feature vector as input. The entire function can be compactly written as $f(x) = w_2^T \phi(W_1 x + b_1) + b_2$, where $W_1 \in \mathbb{R}^{K \times D}$ is the weight matrix, $b_1 \in \mathbb{R}^K$ is the bias vector, $\phi(\cdot)$ is the non-linear transformation and is applied component-wise, $w_2 \in \mathbb{R}^K$ is the second weight vector, and $b_2 \in \mathbb{R}$ is the second bias term. The parameters are then determined by minimizing a risk function as seen before.

A deep neural network can be described as a hierarchical stacking of these parameterized basis functions. Each feature vector is now called a network layer and each component within a feature vector is called a network unit. The layers can be seen as the building blocks of a neural network and regression can be performed by composing various forms of these building blocks; this modularity gives the great versatility of neural networks. There exists a great variety of neural network architectures, from the classic feed-forward neural network, to the convolutional neural networks for image data, to recurrent neural networks for sequence data. In this thesis, I am only concerned with regression with feed-forward neural networks and I review the case of the single-hidden layer neural network below. We denote x as the input to the network. The input is then linear transformed to $W_1 x + b_1$, where $W_1 \in \mathbb{R}^{K \times D}$ is the weight matrix and $b_1 \in \mathbb{R}^K$ is the bias vector. To each element of this output a non-linearity $\phi(\cdot)$ such as the rectified linear unit is then applied. The resulting vector is the hidden network layer and each vector component is a network unit. This vector is then transformed with a second linear transformation to the network output $z = W_2 \phi(W_1 x + b_1) + b_2$, where $W_2 \in \mathbb{R}^{T \times K}$ is the second weight matrix, $b_2 \in \mathbb{R}^T$ is the second bias vector, and $z \in \mathbb{R}^T$ is the predicted output. We then define a loss function, e.g. the Euclidean loss $L = \frac{1}{N} \sum_{i=1}^N \|y_i - z_i\|_2^2$. The final model is obtained by minimizing the loss function with respect to the parameters W_1 , W_2 , b_1 , and b_2 . The hope is that this model generalizes well to unseen data $\mathbf{x}_{\text{test}}$, $\mathbf{y}_{\text{test}}$. However, the main difficulty with this approach is overfitting, which means that during optimization the loss on the training set $\mathbf{x}$, $\mathbf{y}$ is decreasing while the loss on the test set $\mathbf{x}_{\text{test}}$, $\mathbf{y}_{\text{test}}$ is increasing. To avoid overfitting we can add a regularization term that penalizes large weights and has the effect of generating a more general model. For example, the $L2$ objective function becomes $L = \frac{1}{N} \sum_{i=1}^N \|y_i - z_i\|_2^2 + \lambda_1 \|W_1\|_2^2 + \lambda_2 \|W_2\|_2^2 + \lambda_3 \|b_1\|_2^2 + \lambda_4 \|b_2\|_2^2$ (the name $L2$ comes from the fact the 2-norm is used to penalize large weights).

It is straightforward to extend the described single-hidden layer case to multiple hidden layers.

2.2.4 Bayesian learning of neural networks

Bayesian neural networks is a class of probabilistic models that gets increasing attention in the field of deep learning. In contrast to traditional neural network, the network parameters are probability distributions rather than point estimates. The resulting models are typically more robust against overfitting, provide uncertainty estimates for parameters and predictions, and can handle small data sizes. In line with the Bayesian paradigm, we have to define a prior distribution as well as a likelihood function. The prior distribution over the network parameters θ gives a prior distribution over parametric functions $f(x; \theta)$, where $f(\cdot)$ is the network function and x is the network input. The weights W_i and biases b_i in layer i are often given standard Gaussian prior distributions $p(W_i) = \mathcal{N}(0, I)$ and $p(b_i) = \mathcal{N}(0, I)$. The likelihood function is treated as in Generalized Linear Models where instead of a linear model we now use the neural network, e.g. for the Gaussian likelihood we get $p(\mathbf{y} \mid \theta, \mathbf{x}) = \mathcal{N}(f(\mathbf{x}; \theta), \sigma_n^2 I)$. In general, the formulation of a Bayesian neural network is straightforward but the inference part is more challenging as it involves the integration over an intractable posterior distribution. The efforts to approach the inference problem can be structured into three lines of work, (i) deterministic inference, (ii) stochastic inference, and (iii) leveraging the relation between neural networks and Gaussian processes for inference. Each approach has its own benefits and limitations. In the following, I briefly review the related work.

The key idea of deterministic inference is to approximate the intractable posterior distribution using some tractable distribution. In the classical posterior approximation the posterior is expressed in a simple form, e.g. as a multivariate Gaussian. One of the first work was done by [33] who used the Laplace method to approximate the posterior. In a first step, the neural network is optimized to find the mode. In a second step, a Gaussian is fitted to the mode where the covariance is determined with the Hessian of the likelihood. The authors of ref. [34] then used these approximations to compute the model evidence and compare various model configurations. As an alternative analytical approximation to Gaussian/Laplace approximation we

can define an approximating class $q(\theta)$, e.g. the product of exponential family distributions parameterized by ζ , define a discrepancy measure between $q(\theta)$ and $p(\theta \mid \mathbf{y}, \mathbf{x})$, and then optimize ζ to minimize the discrepancy to get the resulting $\hat{q}(\theta)$. This procedure is called variational Bayes if the discrepancy is measured with the Kullback-Leibler divergence as $\text{KL}(q(\theta) \parallel p(\theta \mid \mathbf{y}, \mathbf{x}))$, and it is called expectation propagation if the discrepancy is measured as $\text{KL}(p(\theta \mid \mathbf{y}, \mathbf{x}) \parallel q(\theta))$. Specifically, in variational Bayes, we maximize a lower bound to the model evidence and discard an intractable term in the KL divergence. The resulting objective function is called Evidence Lower Bound (ELBO) and can be written as

$$\text{KL}(q(\theta) \parallel p(\theta \mid \mathbf{y}, \mathbf{x})) \propto - \int q(\theta) \log p(\mathbf{y} \mid \theta, \mathbf{x}) d\theta + \text{KL}(q(\theta) \parallel p(\theta)),$$

where the integral is the expected log likelihood. This integral is typically intractable for Bayesian neural networks which makes the optimization difficult. The author of [35] was the first that formulated the corresponding objective function for a neural network model (although with a different motivation and only for a single-hidden layer neural network). A big step towards the variational Bayes as we know it today was made by [21]. This work introduced data subsampling to the objective function which allowed to scale to large data, and it approximated the expected log likelihood with a Monte Carlo estimates which allowed to scale to complex models. Blundell et al. [36] reparameterized the expected log likelihood from [21] and applied it to Bayesian neural networks. An alternative approach was taken by [37] who formulated the objective function based on expectation propagation and named their approach probabilistic back propagation. Finally, [38] used α -divergence instead of KL-divergence as the discrepancy measure. By changing the parameter α this method can interpolate between variational Bayes $\alpha \rightarrow 0$ and expectation propagation $\alpha \rightarrow 1$ which typically leads to superior performance. However, a major limitation of all variational Bayes approaches is that maximizing the lower bound involves discarding an intractable term in the KL divergence. As outlined in [39], variational inference methods still lack theoretical guarantees and also in practice it is difficult to assess the quality of the posterior approximation.

Stochastic inference provides an alternative where the key idea is to sample from the true posterior distribution using Markov chain Monte Carlo methods. Markov chain Monte Carlo is a sequential algorithm to obtain correlated draws from the posterior distribution and it bypasses the need to approximate the intractable

marginal likelihood $p(\mathbf{y} \mid \mathbf{x})$. One can argue that the samples are more useful than the analytic form for $p(\theta \mid \mathbf{y}, \mathbf{x})$ since for high-dimensional θ we are not interested in the full posterior distribution but rather in the functionals $f(\theta)$ that are interpretable. The samples can then be used to calculate a variety of posterior and predictive summaries. As the number of samples increases, the summaries become more accurate. Markov chain Monte Carlo constructs a Markov chain with the stationary distribution $p(\theta \mid \mathbf{y}, \mathbf{x})$ where a particular transition kernel is chosen. Most MCMC algorithms belong to the family of Metropolis-Hastings algorithms:

1. Sample a proposal $\theta' \sim T(\theta^{t-1})$, where $T(\cdot)$ is the transition kernel and θ^{t-1} is the draw of the previous time step
2. Accept proposal by letting $\theta^t = \theta'$ with probability

$$\min \left\{ 1, \frac{p(\theta')p(\mathbf{y} \mid \theta', \mathbf{x})}{p(\theta^{t-1})p(\mathbf{y} \mid \theta^{t-1}, \mathbf{x})} \frac{T(\theta^{t-1})}{T(\theta')} \right\}$$

The design of an efficient Metropolis-Hastings algorithm involves as a key step choosing the transition kernel $T(\cdot)$. For example, the Gibbs sampler draws a subset of θ from their exact conditional distributions after fixing the others. As another example, the random walk sampler is a distribution centered at θ^{t-1} with a tunable covariance. As the last example, Hamiltonian Monte Carlo exploits the gradient information to generate samples far from θ^{t-1} . The choice which MCMC algorithm to use should be guided by the computational bottlenecks that arise. The first bottleneck is that the time per iteration increases with the number of parameters p and the sample size N due to the cost of sampling and calculating the acceptance probability. The second bottleneck is that the MCMC draws are autocorrelated, and samples with a high autocorrelation contain less information about the posterior distribution. The autocorrelation increases as p increases and as N increases. Among the beforementioned transition kernels, Hamiltonian Monte Carlo results in the lowest autocorrelation for complex distributions and it is the gold standard for Bayesian neural networks. Neal [40] first applied Hamiltonian Monte Carlo (also Hybrid Monte Carlo) to sample from the posterior distribution of a neural network. This work was a milestone that showed that (for simple) network models Markov chain Monte Carlo was a feasible inference method and can lead to superior predictive performance. However, it requires tuning highly sensitive parameters such

as the length and the number of integration steps (also leapfrog step) for each new transition proposal. It also doesn't scale to large data sizes as the gradient must be evaluated on the entire data set. The authors of [41] introduced Stochastic Gradient Langevin Dynamics which could overcome some of these issues. Langevin dynamics is a special case of Hamiltonian dynamics with only a single leapfrog step. By adding the right amount of noise to the standard stochastic gradient optimization, the iterates converge to samples from the posterior distribution as the step sizes are annealed.

$$\delta\theta_t = \frac{\epsilon_t}{2} \left(\nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^n \nabla \log p(y_{ti} | \theta_t) \right) + \nu_t$$

$$\nu_t \sim \mathcal{N}(0, \epsilon_t),$$

where ϵ_t is the step size that is annealed over time. Only having one leapfrog step simplifies the use of the inference method and the stochastic estimates of the gradient of the likelihood allows the method to scale to large data sizes. The main difficulty is that the method typically collapses to a single mode and then only explores this mode. The reason is that as ϵ_t decreases very fast the probability to jump outside the mode becomes very low. The small ϵ_t additionally leads to highly autocorrelated samples which slows down the convergence of the Markov chain. The author of [42] introduces Stochastic Gradient Fisher Scoring which extends the Stochastic Gradient Langevin Dynamics algorithm to mitigate the poor mixing and the highly correlated samples. The basic idea is for large step sizes to sample from an approximate Gaussian and for smaller step sizes to sample from a non-Gaussian approximation. A problem with the methods that rely on first-order Langevin dynamics is that they do not include the crucial momentum term of Hamiltonian Monte Carlo. Chen et al. [43] develop Stochastic Gradient Hamiltonian Monte Carlo that aims at keeping the characteristics of the traditional Hamiltonian Monte Carlo but that can be combined with subsampling. They find that the natural implementation of stochastic approximation to the gradients can be arbitrarily bad. But adding friction to the momentum update can counteract the effect of the gradient noise. Despite the great progress that has led to scalable MCMC methods there is a remaining, more fundamental, limitation when applied to neural networks. Neural networks are essentially big mixture models where the posterior distribution is non-identified. MCMC methods are able to explore this posterior but only in a small neighborhood, and where this neighborhood is located

strongly depends on the initial conditions. Sampling then may produce useful predictive models but it still fails to sample correctly from the whole posterior.

An alternative to either deterministic or stochastic inference is the recent work from [44]. They showed that stochastic gradient descent can implement approximate Bayesian inference. The iterates simulate a Markov chain with a stationary distribution which approximates the posterior. This approach can be seen as a hybrid of Monte Carlo and variational Bayes.

Another promising alternative to deterministic or stochastic inference with the original neural network model is the reformulation of the prior distribution of the neural network as a Gaussian process prior. While neural networks are parametric models and Gaussian processes are nonparametric models it can be shown that Bayesian neural networks are closely related to Gaussian processes. Given a feed-forward neural network and placing Gaussian distributions over the weights, then in the limit of infinite network width the model becomes a Gaussian process with the kernel being dependent on the network architecture and prior specification. The details are given below in Section 2.4. This reformulation then allows fully Bayesian prediction with deep neural networks where the computation only requires building the kernel and straightforward matrix computations without the need of stochastic gradient-based training. Neal [40] first showed that neural networks with specific prior distributions in the limit of infinite network width converge to stochastic processes, e.g. if the prior is Gaussian then the network converges to a Gaussian process. The authors of [45] and [46] derived for a single hidden-layer neural network the analytic kernel of the corresponding Gaussian process. They also noted its use for exact Bayesian inference in regression. Lee et al. [31] build upon this first work for single hidden-layer networks and derived the exact equivalence between infinitely wide *deep* neural networks and Gaussian processes. They used the Gaussian processes for Bayesian inference on regression tasks and compared their performance to stochastic gradient training. They found that the neural network accuracy approaches that of the Gaussian process with increasing layer width, that the Gaussian process predictions typically outperform the neural networks with finite width, and that the Gaussian process uncertainty is strongly correlated with the prediction error. In a parallel work [47] additionally derives the convergence rate towards a Gaussian process if all layers are taken to infinity. They then compared the Gaussian process to the neural networks where inference was

done with Hamiltonian Monte Carlo. Going beyond feed-forward neural networks, [48] derived the exact equivalence between (residual) convolutional neural networks and Gaussian processes in the limit of infinitely many convolutional filters. They then compared the performance to the traditional networks trained with stochastic gradient descent. A general limitation of the approach described above is scalability to large N due to the matrix inverse problem in Gaussian process inference. Recent progress in scalable Gaussian processes, e.g. [49], may provide paths for future research directions.

2.2.5 Sparse learning for variable selection

Sparse learning and its application to variable selection has received increased attention in the recent years and is an active research field in machine learning. An introduction to variable selection in machine learning can be found e.g. in [50], and [51] gives a review on popular variable selection methods and systematically tests their performance on synthetic data. To start, let's consider a statistical model with a large number of parameters $\theta = (\theta_1, \dots, \theta_p)$ where only a subset of those parameters must be far from zero to explain some observed data. Typically, such setting arises when there are many potentially relevant variables but it is not known beforehand which ones are important. For example, in spatial interaction modelling the goal is to estimate the number of trips for an origin-destination pair based on various potentially relevant attributes such as the number of people living and working at origin and destination, the density of amenities at origin and destination, the geographic and travel time distances, etc. Variable selection can then help to better understand the data, reduce measurement requirements, reduce inference times, and improve prediction performance since less relevant parameters are informed by the same amount of data [50]. The variable selection methods found in literature can be structured into (i) filters that select variables by ranking them with e.g. correlation coefficients, (ii) wrappers that use the fixed model to score how useful a certain subset of variables is for the prediction, and (iii) embedded methods that are similar to wrappers but integrate the scoring directly into the objective function.

To introduce filters, let's consider a set of input-output pairs $\{(x_{ij}, y_i)\}$, where $i = 1, \dots, N$ with N being the number of data points and $j = 1, \dots, D$ with D

being the number of variables. The idea is to build the predictor with only a subset of the variables based on some variable ranking. To rank the variables we define a scoring function. A common choice is a correlation criteria such as Pearson or Spearman's correlation coefficient e.g. the estimate of the Pearson correlation coefficient between variable and target is

$$\hat{R}_j = \frac{\sum_{i=1}^N (x_{ij} - \bar{x}_j)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_{ij} - \bar{x}_j)^2 \sum_{i=1}^N (y_i - \bar{y})^2}}.$$

However, Pearson (Spearman) correlation can only detect linear (monotonic) relations. An alternative is an information theoretic criteria, e.g. the mutual information between variable and target. If the variable is discrete (or discretized) the mutual information is given by

$$I_j = \sum_X \sum_Y P(X = x_j, Y = y) \log \frac{P(X = x_j, Y = y)}{P(X = x_j)P(Y = y)}.$$

The benefit of filters is that they are typically computationally very efficient. Their limitation is that they select the variable subset independent of the prediction model. For example, selecting the most relevant variables would result in suboptimal prediction if e.g. (i) some variables are redundant or (ii) variables are only predictive in combination as in the XOR problem.

In contrast, wrapper methods use the prediction performance of the model as the scoring function to select variable subsets. The main challenge becomes that the search over all possible variable subsets is NP-hard. Various search strategies have been developed e.g. the best-first strategy. The benefit of wrappers is that they are simple and universal. Their limitation is that the prediction model has to be retrained from scratch each time.

Embedded methods (sparse learning) do the variable selection as part of training the model (if non-Bayesian) or as part of the model inference (if Bayesian) which avoids the computational inefficiency of wrapper methods. In the non-Bayesian scenario, we optimize an objective function which now consists of two terms, one term that relates to the fit of the model and is maximized, and one term (regularization term) that relates to the number of variables and is minimized. The regularization term is typically $R_p(\theta) = \lambda \|\theta\|_p$, where λ is a scale parameter and

$\|\cdot\|_p$ is the l_p -norm with $\|\theta\|_p = \left(\sum_{i=1}^M |\theta_i|^p\right)^{\frac{1}{p}}$. The l_0 regularization then effectively penalizes the number of variables but the optimization becomes NP-hard and requires heuristics. Much more common is the l_1 regularization

$$R_1(\theta) = \lambda \sum_{i=1}^M |\theta_i|$$

which forces a parameter to zero when it falls below a certain threshold. This method was introduced by [52] under the name lasso (least absolute shrinkage and selection operator); a review on lasso and its generalizations can be found in [53]. To build intuition why this method leads to sparse solutions let's consider a two-dimensional parameter space with elliptical contours for the fit term and rectangular contours for the regularization term. The optimal solution of the constraint optimization problem will be located where the lowest level set of the fit term intersects with the constraint surface given by the regularization term. This will most likely (and more likely with increasing dimensionality) be in the corners of the regularization contours which correspond to sparse solutions.

The above mentioned sparse learning methods can also be recovered in the Bayesian setting. A review on sparse Bayesian learning and its connection to non-Bayesian sparse learning can be found e.g. in [27]. Instead of using a regularization term, the sparsity behavior is determined by the choice of prior distribution on the parameters. For example, the l_1 regularization is equivalent to maximum a posteriori (MAP) estimation when a Laplace prior centered at zero is used

$$p(\theta \mid \lambda) = \prod_{i=1}^M \text{Lap}\left(\theta_i \mid 0, \frac{1}{\lambda}\right) \propto \prod_{i=1}^M \exp(-\lambda|\theta_i|)$$

which leads to the penalized negative log likelihood

$$f(\theta) = -\log p(\mathbf{y} \mid \theta, \mathbf{x}) - \log p(\theta \mid \lambda) = -\log p(\mathbf{y} \mid \theta, \mathbf{x}) + \lambda \|\theta\|_1.$$

However, when we act Bayesian and integrate over the posterior (as opposed to finding the mode in MAP), the Laplace prior is not necessarily sparsity-inducing anymore as the entire parameter space of the prior is taken into account (as opposed to a single point in parameter space in MAP) and the posterior therefore stretches away from the desired neighborhood.

To enable Bayesian sparsity we need a prior that collapses all the posterior mass to either relevance or irrelevance. Sparsity-inducing priors with such characteristic can be categorized into two component discrete mixture priors (spike-and-slab) and continuous shrinkage priors. The spike-and-slab prior was introduced by [54] and further extended by others. In its general form it can be formulated as

$$\begin{aligned}\theta_i \mid \lambda_i, \sigma_0, \epsilon &\sim \lambda_i \mathcal{N}(0, \sigma_0^2) + (1 - \lambda_i) \mathcal{N}(0, \epsilon^2), \\ \lambda_i &\sim \text{Ber}(\pi), \quad i = 1, \dots, p,\end{aligned}$$

where ϵ is much smaller than σ_0 and λ_i decides if a parameter comes from the spike (and is therefore close to zero) or comes from the slab. This prior often leads to good performance but the results are very sensitive to choices in prior parameters e.g. the slab width σ_0 . An alternative from the family of continuous shrinkage priors is the horseshoe prior

$$\begin{aligned}\theta_i \mid \lambda_i, \tau &\sim \mathcal{N}(0, \tau^2 \lambda_i^2), \\ \lambda_i &\sim \mathcal{C}^+(0, 1), \quad i = 1, \dots, p, \\ \tau \mid \sigma_g &\sim \mathcal{C}^+(0, \sigma_g^2)\end{aligned}$$

where the ‘global’ shrinkage hyperparameter τ shrinks all parameters to zero while the heavy-tailed half-Cauchy prior for the ‘local’ hyperparameters λ_i allows some parameters to escape the shrinkage. The performance is comparable to the spike-and-slab prior but is much more easy to implement since the performance of the model is not as sensitive to the choices in the prior parameters [55, p. 5019]. One issue with this prior is that relevant parameters far from zero are almost not regularized which can cause them to diffuse to very large values. Piironen et al. [55] solved this issue with their regularized horseshoe prior

$$\begin{aligned}\theta_i \mid \lambda_i, \tau, \sigma_0 &\sim \mathcal{N}(0, \tau^2 \tilde{\lambda}_i^2), \quad \tilde{\lambda}_i^2 = \frac{\sigma_0^2 \lambda_i^2}{\sigma_0^2 + \tau^2 \lambda_i^2}, \\ \lambda_i &\sim \mathcal{C}^+(0, 1), \quad i = 1, \dots, p, \\ \sigma_0^{-2} &\sim \text{Gamma}(\alpha_0, \beta_0), \quad \alpha_0 = \nu_0/2, \quad \beta_0 = \nu_0 s_0^2/2 \\ \tau \mid \sigma_g &\sim \mathcal{C}^+(0, \sigma_g^2),\end{aligned}$$

where $\tau^2 \lambda_i^2 \ll \sigma_0^2$ shrinks small values as in the original horseshoe prior and $\tau^2 \lambda_i^2 \gg \sigma_0^2$ regularizes large parameters with a Gaussian slab. Here the marginalization of

the Gamma distribution on σ_0^{-2} results in a Student- $t_{\nu_0}(0, s_0^2)$ for the parameters far from zero which is a weakly-informative prior. The regularized horseshoe can be understood as the spike-and-slab prior with a finite slab while the original horseshoe can be understood as the spike-and-slab prior with an infinite slab.

The presented research above is mainly focused on linear models. For non-linear models, a straightforward approach is to first, find a subset of variables on a linear model and second, use that subset to build the non-linear model. However, important non-linear relations between variables may be missed. Other research therefore focuses on integrating the variable selection process directly into the inference process of non-linear models. One of the first approaches is Automatic Relevance Determination (ARD) which was developed by [56] and [40]. ARD is an hierarchical Bayesian approach that was originally applied to neural network models to automatically determine the relevance of the input variables. The hierarchical prior on the weights w_{ij} that connect the input variables $i = 1, \dots, D$ with the units of the first hidden layer $j = 1, \dots, H_1$ is given by

$$\begin{aligned} w_{ij} \mid \lambda_i &\sim \mathcal{N}(0, \lambda_i^2) \\ \lambda_i^{-2} \mid \alpha, \beta &\sim \text{Gamma}(\alpha, \beta) \end{aligned}$$

where the hyperparameter λ_i determines the range of variation for parameters connected to input variable i and thereby encodes it's relevance (if the width of the zero mean Gaussian is close to zero the input variable is irrelevant). Neal [45] used the same idea for Gaussian process models with linear or squared exponential covariance functions. For example, in the case of the linear kernel the prior is given by

$$\begin{aligned} k(x, x_*) &= \sum_{i=1}^D x_i x_{*i} \sigma_i^2 \\ \sigma_i^{-2} \mid \alpha, \beta &\sim \text{Gamma}(\alpha, \beta), \end{aligned}$$

where σ_i controls how relevant input variable i is for predicting the targets. If σ_i is close to zero then variable i has little effect on the covariance between cases and is therefore irrelevant. However, it can be shown that the hierarchical prior in ARD implies an independent Student-t priors for each w_{ij} [57] and when integrating over this prior the heavy tails of the Student-t distribution limit the ability to set

variables to zero. In contrast, the horseshoe prior sets variables to either relevant or irrelevant. The authors of [58] and [59] use this quality of the horseshoe prior to prune inactive units in Bayesian neural networks to achieve model compression and less overfitting when the data size is small. Popkes et al. [60] uses a similar approach but with the goal of variable selection in a Bayesian neural network with an application to intensive care prediction. Importantly, [55] argued that for all ARD-like approaches with a non-linear model the term relevance is misleading since relevance of a variable is typically defined by its predictive power on the target variable while ARD-like approaches rather measure the non-linearity of a parameter along one dimension. This is important to keep this in mind when interpreting the results.

2.2.6 Markov chain Monte Carlo methods

In this section, I introduce Markov chain Monte Carlo (MCMC) and specifically the Hamiltonian Monte Carlo (HMC) algorithm that will be used for the inference of the Gaussian process models in Section 2.4. Whereas Section 2.2.4 already gave a brief introduction to MCMC in the context of Bayesian neural network, I will here give a more thorough introduction to MCMC independently of the Bayesian context and try to build intuition why the (relatively) sophisticated HMC algorithm is needed for the high-dimensional models that we use. A great review on MCMC can be found e.g. in [30]. I additionally base the following introduction on a series of lectures given by Michael Betancourt which can be accessed at [61] and in written form at [62]. To start, we can abstract Bayesian inference and simply have given a target distribution $p(\theta)$ where the goal is to compute expectation values

$$\mathbb{E}[f(\theta)] = \int f(\theta)p(\theta)d\theta,$$

where $f(\theta)$ is any function of interest. For other than the most simple models this integral has no analytic answer and the integral needs to be computed numerically. Numerical integration is a hard computational challenge and triggered much research in various fields around mathematical analysis. One of the most simple approaches is quadrature or equivalently Riemann integration. The target density is evaluated at a grid of points that have the same volume inbetween each other and the resulting values are averaged up. But the cost of this algorithm scales with the

number of grid points which scales exponentially with the number of dimensions of the parameter space N^p . Instead of a uniform grid we therefore need to focus on the most relevant neighborhoods in parameter space where the probability mass, $p(\theta)d\theta$, contributes the most to the expectation value $E[f(\theta)]$ (assuming a well behaved $f(\theta)$). In high dimensions the volume $d\theta$ behaves counterintuitive and the probability mass concentrates on a hypersurface called typical set which becomes narrower with increasing dimensions. Any method for estimating the expectation values aims at quantifying that typical set [62].

One method is Monte Carlo which draws exact samples from the target distribution and which gives a stochastic grid of samples that quantify the typical set. Monte Carlo estimators are then computed as

$$\hat{f} = \frac{1}{N} \sum_{i=1}^N f(\theta_i),$$

where the expectation values is the average of the function evaluated at the samples. Importantly, the central limit theorem gives

$$\hat{f} \sim \mathcal{N}(E[f(\theta)], \text{MC-SE}[f(\theta)])$$

$$\text{MC-SE}[f(\theta)] = \sqrt{\frac{\text{var}(f(\theta))}{N}},$$

which makes the great properties of Monte Carlo estimators: they are unbiased and the number of samples to reduce the error is known. However, typically it is only possible to draw exact samples for known distributions, e.g. by using the inverse CDF of a one-dimensional Gaussian.

For more complex models, the Monte Carlo approach can be extended to Markov chain Monte Carlo. The idea is to construct a Markov chain that converges to the target distribution and to use the history of the states to quantify the typical set. In contrast to Monte Carlo the samples are now correlated. The Markov chain is constructed with a Markov transition which is typically a probability density. Importantly, the Markov transition only targets the target distribution if the invariance term holds $p(\theta) = \int p(\theta')T(\theta | \theta')d\theta'$. The Markov chain Monte Carlo estimator then becomes

$$\hat{f} = \frac{1}{N} \sum_{i=1}^N f(\theta_i),$$

where the expectation values is the average of the function evaluated at the samples. The central limit theorem holds

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f(\theta)], \text{MCMC-SE}[f(\theta)])$$

$$\text{MCMC-SE}[f(\theta)] = \sqrt{\frac{\text{var}(f(\theta))}{\text{ESS}[f(\theta)]}},$$

but in contrast to simple Monte Carlo the error is computed using $\text{ESS}[f(\theta)]$ instead of N . Here $\text{ESS}[f(\theta)]$ is the effective sample size which is the equivalent number of Monte Carlo samples that would have given the same error; it is the key performance criterion when measuring the efficiency of a Markov chain Monte Carlo implementation. The effective sample size is then determined by the sum of autocorrelations (which limit the independent information in the Markov chain) across all lags

$$\text{ESS}[f(\theta)] = \frac{N}{1 + 2 \sum_{i=1}^{\infty} \rho_i[f(\theta)]}$$

$$\rho_i[f(\theta)] = \frac{\mathbb{E}[(f(T^i \theta) - \mu_f)(f(\theta) - \mu_f)]}{\text{var}[f(\theta)]}, \quad \mu_f = \mathbb{E}[f(\theta)].$$

In reality the Markov chain only has a finite length and therefore ρ_i , $\text{ESS}[f(\theta)]$, and $\text{MCMC-SE}[f(\theta)]$ needs to be estimated; details can be found in [30].

In the discussion above the assumption was that a Markov transition $T(\theta \mid \theta')$ exists that satisfies the invariance property and therefore targets the target distribution. For complex models, approaches such as Gibbs sampling are not feasible anymore and we use the density information $p(\theta)$ instead to construct the transition automatically. In the following, I introduce two concrete examples of such Markov transitions. The most famous and first MCMC algorithm is Random Walk Metropolis (RWM) that was developed in the Manhattan project [63, 64]. In it's basic form the transition takes the form

$$T(\theta \mid \theta') = \mathcal{N}(\theta \mid \theta', \sigma^2 I) \min \left(1, \frac{p(\theta)}{p(\theta')} \right).$$

The algorithm consists of two steps: the proposal phase and the correction term. The intuition behind is that the proposal jumps to regions of high volume and the correction only accepts proposals with high density. But in high dimensions Random Walk Metropolis becomes infeasible. Since there is exponentially more

volume outside of the typical set than inside, the proposal constantly jumps there. But since the density is low outside the typical set, the acceptance rate is very low which leads to a very slow exploration of the typical set. Instead of a Random Walk we need a sampler that senses when it is leaving the typical set to correct itself. One way is to construct a vector field that is aligned with the typical set by transforming the gradients of the target density. Hamiltonian Monte Carlo (HMC), established in the statistical community by [65], is the procedure to achieve such exploration and is the gold standard for Bayesian inverse problems in high-dimensional parameter spaces. I use the HMC algorithm for the inference of the Gaussian process models from Section 2.4 and give some intuition about the workings of the algorithm as well as the basic technical steps below. A great technical review can be found e.g. in [65]. In its essence, the Hamiltonian Monte Carlo algorithm is used within the Markov chain to make proposals for the next parameter state that are distant from the current parameter state, thereby leading to good exploration of the probability density. The Hamiltonian algorithm originates in physics but can be applied to sampling in statistics by introducing fictitious momentum variables. By analogy, the flipped probability density becomes a valley in which an ice hockey puck, the sampler, is given some momentum from a starting position. The ice hockey puck will then follow some trajectory (going up and down the valley) which is defined by Hamiltonian dynamics. At a certain point the puck is stopped and the position is used as proposal for the next starting point of the puck which corresponds to the next proposal for the parameter state. In the framework of statistical mechanics this takes the following form: First, the original parameter space is extended by an auxiliary variable $p(\theta) \rightarrow p(\theta, q) = p(\theta)p(q | \theta)$, where $\theta \in \mathbb{R}^p$ and $q \in \mathbb{R}^p$. The density function in this extended parameter space is set to

$$\begin{aligned} p(\theta, q) &\propto \exp\left(\frac{-H(\theta, q)}{T^*}\right) \\ &= \exp\left(\frac{-U(\theta)}{T^*}\right) \exp\left(\frac{-K(q)}{T^*}\right), \end{aligned}$$

where $H(\theta, q)$ is the Hamiltonian, T^* is the temperature of the system, $U(\theta)$ is the potential energy, and $K(q)$ is the kinetic energy. The energies are set to

$$U(\theta) = -\log p(\theta), \quad K(q) = -\log \mathcal{N}(q | 0, M^{\frac{1}{2}}),$$

where $M \in \mathbb{R}^{p \times p}$ is the mass matrix. The dynamics in the extended parameter space are then described by Hamilton's equations

$$\frac{d\theta_i}{dt} = \frac{\partial H(\theta, q)}{\partial q_i}, \quad \frac{dq_i}{dt} = \frac{-\partial H(\theta, q)}{\partial \theta_i},$$

where $t \in \mathbb{R}$ is the time variable. The transition density function in the original parameter space is given by

$$T(\theta \mid \theta') = \theta_{\epsilon, T_*}(\theta', q') \min \left(1, \frac{p(\theta, q)}{p(\theta', q')} \right),$$

where θ_{ϵ, T_*} is the leapfrog integrator with stepsize ϵ and total integration time T_* . With appropriate tuning of ϵ and T_* , Hamiltonian Monte Carlo allows large jumps in parameter space combined with high acceptance probabilities.

2.3 Contributions

In this thesis, I propose an explainable machine learning model that predicts urban mobility flows. The proposed approach gives the pattern recognition capabilities of deep neural networks but also explainable predictions in terms of selecting the most relevant variables for the problem at hand. Hereafter I refer to the model as Horseshoe Neural Network Gaussian Process (H-NNGP). The model is based on the probabilistic formulation of an infinitely wide deep neural network as a Gaussian process, first described in [31] and termed Neural Network Gaussian Process (NNGP). I choose the formulation as a Gaussian process because it explicitly models the uncertainty in the data set, as well as in the prediction function which is important when decision making involves many uncertainties as in the case of urban planning.

Typically, the probabilistic formulation of a neural network would be a Bayesian neural network with either stochastic inference [40] or deterministic inference [36]. However, as outlined in Section 2.2.4, the former inference method limits exploration of the posterior distribution to a small neighborhood in parameter space due to the non-identifiability of neural networks while for the latter inference method it is difficult to assess the quality of the variational posterior approximation. Both methods *may* produce useful predictive models but they still fail to sample correctly

from the whole posterior which contradicts the power of the Bayesian formulation which in its exact form is very good in characterizing uncertainty. In contrast to these traditional inference methods, the chosen formulation as a Gaussian process allows for *exact* Bayesian inference in deep neural networks.

While there is some research on automatic variable selection for Bayesian neural networks either with stochastic inference [40] or deterministic inference [60], to the best of my knowledge no research is done on integrating automatic variable selection in the formulation of deep neural networks as Gaussian processes. In this thesis, I propose a novel model based on the formulation of deep neural networks as Gaussian processes that automatically selects relevant urban variables for the prediction of urban mobility flows. In particular, I integrate the Horseshoe prior into the neural network kernel of the Gaussian process by introducing additional hyperparameters to the kernel. Similar in nature to the Automatic Relevance Determination (ARD), after training, some of the hyperparameters associated to an input variable will be way smaller than others, indicating that the corresponding input variable is not relevant for the predictions. The detailed method including prior specification and inference procedure is described in Section 2.4. I then verify the effectiveness of the proposed method with synthetic data in Section 2.5 and real-world data in Section 3.4.

2.4 Proposed Method

In this section, I describe my proposed method and justify the choice of the design that allows for fully Bayesian inference of deep neural networks with automatic variable selection.

The entire analysis in Chapter 3 is based on the direct comparison between linear and non-linear spatial interaction models in terms of predictions performance and variable importance. The choice of the design in this chapter was therefore guided by the idea of a framework which can replicate the linear *as well as* the non-linear model and thereby ensure a maximum of comparability. The chosen framework of Neural Networks as Gaussian Processes with sparsity-inducing priors can indeed replicate the linear model (simply by setting the number of hidden layers to zero) as well as powerful non-linear models (with increasing number of hidden layers).

A thorough experimental comparison between various models, variable selection methods, and inference methods might indeed be useful to better understand how these different choices in design perform in different data scenarios but such a comparison is outside the scope of the thesis. Instead, the choice of the design is mainly based on theoretical and experimental insights that come from existing literature:

- How do different inference methods compare? One of the alternatives to the ‘Neural Network as Gaussian Process’ design would have been a Bayesian Neural Network with, for example, Hamiltonian Monte Carlo as the inference method. Neural networks are essentially big mixture models where the posterior distribution is non-identified due to weight symmetries [66]. MCMC methods are able to explore this posterior but only in a small neighborhood, and where this neighborhood is located strongly depends on the initial conditions. Sampling then may produce useful predictive models but it still fails to sample correctly from the whole posterior. This explains the success in prediction performance of Hamiltonian Monte Carlo for neural networks [40] but there is no theoretical guarantee that the posterior parameters (and thereby the variable importance) are correct. This is in contrast to exact Bayesian inference with the chosen Gaussian process approach.
- How do different variable selection methods compare? The focus of the modeling framework was on the end-to-end comparison between linear with non-linear models in terms of prediction performance and variable importance. Filter methods based on Pearson criteria or Information criteria would not add to the understanding of the differences between these two model types and were therefore not compared. Within the set of embedded variable selection methods for non-linear models, Automatic Relevance Determination like approaches have proven to be effective [40] and can be applied to linear models in the same way.
- How do different priors compare? The choice of the Horseshoe prior is mainly based on existing literature which extensively compared different priors (among others the spike-and-slab prior and the Horseshoe prior) in theoretical terms as well as with practical experiments [55, 67, 68]. Based on this work sparsity-inducing priors can be categorized into two component

discrete mixture priors (spike-and-slab), e.g. [54], and continuous shrinkage priors, e.g. Horseshoe prior [67]. Spike-and-slab priors often lead to good performance but the results are very sensitive to choices in prior parameters. In contrast, the Horseshoe prior from the family of continuous shrinkage priors is more robust to the choice of prior parameters and therefore easier to implement while the performance stays comparable to the spike-and-slab prior.”

In summary, the overall strategy is the reformulation of the prior distribution of the neural network as a Gaussian process prior. This reformulation then allows fully Bayesian prediction with deep neural networks where the computation only requires building the kernel and straightforward matrix computations without the need of stochastic gradient-based training. First, I revisit the correspondence between Gaussian processes and deep, infinitely wide neural networks which results in the NNGP kernel as described in [31]. Second, I discuss my modification to the NNGP kernel that allows for automatic variable selection. Specifically, I propose a horseshoe prior on the weight parameters connected to the input variables which induces sparsity in the first layer of the corresponding neural network; this gives the final H-NNGP kernel. Third, I revisit how Gaussian process priors can be used for exact Bayesian inference as described e.g. in [32].

2.4.1 Notation

I start by describing the notation for the deep neural network model and the corresponding Gaussian process model which I keep close to the original notation in [31]. For the newly introduced sparsity-inducing prior, I use my own notation. For the neural network model, let us consider a feed-forward fully-connected architecture with L hidden layers and N_l hidden units in layers $l = 1, \dots, L$. The input for the model is $x \in \mathbb{R}^D$ and the output is $z \in \mathbb{R}$. In each layer l and unit i , x_i^l is the value after the activation and z_i^l is the value after the linear transformation and before the next activation. In the case of the input, x_i^α is the value of variable i for a particular input α . The non-linear transformation is denoted by $\phi(\cdot)$. The weights

within a layer are denoted by W_{ij}^l and have the following hierarchical prior

$$\begin{aligned} W_{ij}^l &\sim \mathcal{N}(0, \frac{\sigma_w^2}{N_l}) \\ \sigma_w^{-2} &\sim \text{Gamma}(\alpha_w, \beta_w) \end{aligned}$$

The biases within a layer are denoted by b_i^l and have the following hierarchical prior

$$\begin{aligned} b_i^l &\sim \mathcal{N}(0, \sigma_b^2) \\ \sigma_b^{-2} &\sim \text{Gamma}(\alpha_b, \beta_b) \end{aligned}$$

For the base layer $l = 0$, I introduce the regularized Horseshoe prior as follows

$$\begin{aligned} W_{ij}^0 \mid \lambda_j, \tau, \sigma_0 &\sim \mathcal{N}(0, \tau^2 \tilde{\lambda}_j^2), \quad \tilde{\lambda}_j^2 = \frac{\sigma_0^2 \lambda_j^2}{\sigma_0^2 + \tau^2 \lambda_j^2}, \\ \lambda_j &\sim \mathcal{C}^+(0, 1), \\ \sigma_0^{-2} &\sim \text{Gamma}(\alpha_0, \beta_0), \quad \alpha_0 = \nu_0/2, \quad \beta_0 = \nu_0 s_0^2/2 \\ \tau &\sim \mathcal{C}^+(0, \sigma_g^2), \end{aligned}$$

where all weights W_{ij}^0 that connect to the same input component x_j are drawn from a scale mixture of normal distributions with variance $\tau^2 \tilde{\lambda}_j^2$, where τ is the global shrinkage parameter and $\tilde{\lambda}_j$ is the local shrinkage parameter. Finally, for the corresponding Gaussian process model, $\text{GP}(\mu, k)$ is the Gaussian process with mean function $\mu(\cdot)$ and covariance function $k(\cdot, \cdot)$.

2.4.2 Gaussian Processes and Deep Neural Networks

The correspondence between single-layer neural networks and Gaussian processes was first derived in [40]. The correspondence can be seen in the network function

$$z_i^1(x) = b_i^1 + \sum_{j=1}^{N_1} W_{ij}^1 x_j^1(x), \quad x_j^1(x) = \phi \left(b_j^0 + \sum_{k=1}^D W_{jk}^0 x_k \right).$$

If the weights W_{jk}^0 and biases b_j^0 are drawn independently from their prior distributions then $x_j^1(x)$ are independent. And if the weights W_{ij}^1 and biases b_i^1 are drawn independently from their prior distributions and N_l goes to infinity then the Central

Limit Theorems says that $z_i^1(x)$, which is the sum of independent terms, is Gaussian distributed. Further, the multidimensional Central Limit Theorem states that scaled sums converge to multivariate Gaussian distributions. That shows that the collection $\{z_i(x^{\alpha=1}), \dots, z_i(x^{\alpha=m})\}$ is multivariate Gaussian distributed is therefore a Gaussian process. The network function can therefore be formulated as a Gaussian process as

$$\begin{aligned} z_i^1 &\sim \text{GP}(\mu^1, k^1) \\ \mu^1 &= \text{E} [z_i^1(x)] = 0 \\ k^1(x, x_*) &= \text{cov} [z_i^1(x), z_i^1(x_*)] = \text{E} [z_i^1(x)z_i^1(x_*)] \end{aligned}$$

The expectation $\text{E} [z_i^1(x)z_i^1(x_*)]$ can be solved analytically using integration as shown in [40].

The work in [31] builds on this discovery and further derived the correspondence between *deep* neural networks and Gaussian processes. The correspondence can again be seen in the network function that now allows for multiple layers

$$z_i^l(x) = b_i^l + \sum_{j=1}^{N_l} W_{ij}^l x_j^l(x), \quad x_j^l(x) = \phi(z_j^{l-1}(x)).$$

By induction it can be shown that if N_{l-1} goes to infinity then the input to layer l is a Gaussian process and therefore $z_i^l(x)$ is a Gaussian process with

$$\begin{aligned} z_i^l &\sim \text{GP}(\mu^l, k^l) \\ \mu^l &= \text{E} [z_i^l(x)] = 0 \\ k^l(x, x_*) &= \text{cov} [z_i^l(x), z_i^l(x_*)] = \text{E} [z_i^l(x)z_i^l(x_*)] \\ &= \sigma_b^2 + \sigma_w^2 \text{E}_{z_j^{l-1} \sim \text{GP}(0, k^{l-1})} [\phi(z_j^{l-1}(x))\phi(z_j^{l-1}(x_*))] \\ &= \sigma_b^2 + \sigma_w^2 F(k^{l-1}(x, x_*), k^{l-1}(x, x), k^{l-1}(x_*, x_*)) \end{aligned}$$

where F is a deterministic function that describes the relationship between k^l and k^{l-1} . k^l can then be solved analytically, e.g. if $\phi(\cdot)$ is the Rectified Linear Unit (ReLU) non-linearity then k^l becomes the arccosine kernel [69] and k^L can be computed iteratively.

2.4.3 Automatic Feature Selection with Sparsity Inducing Prior

To allow for automatic variable selection, I propose the use of the regularized Horseshoe prior [55] in the first layer of the corresponding neural network

$$\begin{aligned} W_{ij}^0 \mid \lambda_j, \tau, \sigma_0 &\sim \mathcal{N}(0, \tau^2 \tilde{\lambda}_j^2), \quad \tilde{\lambda}_j^2 = \frac{\sigma_0^2 \lambda_j^2}{\sigma_0^2 + \tau^2 \lambda_j^2}, \\ \lambda_j &\sim \mathcal{C}^+(0, 1), \\ \sigma_0^{-2} &\sim \text{Gamma}(\alpha_0, \beta_0), \quad \alpha_0 = \nu_0/2, \quad \beta_0 = \nu_0 s^2/2 \\ \tau &\sim \mathcal{C}^+(0, \sigma_g^2), \end{aligned}$$

where all weights W_{ij}^0 that connect to the same input component x_j are drawn from a scale mixture of normal distributions with variance $\tau^2 \tilde{\lambda}_j^2$, where τ is the global shrinkage parameter that shrinks all parameters to zero while $\tilde{\lambda}_j$ is the local shrinkage parameter that allows individual parameters to escape that shrinkage if that is useful for the prediction. τ and λ_j are themselves unknown parameters. The prior for τ ensures that the data can inform the global shrinkage beyond the initial guess σ_g . And the heavy-tailed Cauchy prior of λ_j allows individual parameters to go to large values and go above the global shrinkage τ if needed. By using $\tilde{\lambda}_j^2$ instead of λ_j^2 directly the main problem of the *original* Horseshoe prior can be solved which is that parameters that escape the shrinkage are almost unregularized and therefore can go to very large values (for details see Section 2.2.5). After marginalizing, the Gamma distribution on σ_0^{-2} then results in a Student- $t_{\nu_0}(0, s_0^2)$ for the parameters far from zero. Finally, σ_g must be set which is the threshold below which parameters are considered irrelevant for the prediction. A good starting point is to relate the impact of σ_g on the prediction to the expected noise in the measurement process σ_n ; details can be found in [55]. This process is challenging for non-linear models and I instead choose to carefully determine the parameter using a sensitivity analysis.

Next, I convert the above formulation of the prior distribution for the first layer of the neural network into the formulation of the Gaussian process prior for deep neural networks. With $W_{ij}^0 \sim \mathcal{N}(0, \tau^2 \tilde{\lambda}_j^2)$ and $b_i \sim \mathcal{N}(0, \sigma_b^2)$ the base case k^0

becomes

$$\begin{aligned} k^0(x, x_*) &= \text{cov} [z_i^0(x), z_i^0(x_*)] = \text{E} [z_i^0(x) z_i^0(x_*)] \\ &= \sigma_b^2 + \tau^2 \sum_{j=1}^D \tilde{\lambda}_j^2 x_j x_{*j} \end{aligned}$$

We can now utilize the recursion relating k^1 and k^0 to complete the iterative series of computations to obtain k^L for the Gaussian process that describes the network's final output.

2.4.4 Bayesian Learning of Neural Networks Using Gaussian Process Priors

Consider a dataset $\{(x^1, y^1), \dots, (x^N, y^N)\}$ with N input-target pairs, for example in spatial interaction modelling, N origin-destination pairs where $\mathbf{x}$ are the input variables such as number of people living and working at origin and destination and $\mathbf{y}$ are the flows between the origins and destinations. The goal is to construct a distribution over functions $z(x)$ that can be used to make predictions at a test point x_* . The distribution over functions is modelled with the above described Gaussian process model and for the noise model I assume a Gaussian centered at $\mathbf{z}$ which allows for exact Bayesian inference with pure matrix calculation. Importantly, the chosen noise model is not optimal for count data (such as counting mobility flows from origins to destinations). However, acceptable results can be expected if the mean value of the count data is large enough, which I test for using sensitivity analysis.

The deep neural network kernel and the noise model are determined by the set of parameters $\theta = \{\sigma_w, \sigma_b, \tau, \sigma_0, \lambda, \sigma_n\}$. Other parameters such as the depth of the network are not modelled explicitly and I simply test for different configurations. The log of the posterior distribution over the parameter space is then proportional to the log prior plus the log marginal likelihood $\log p(\theta | \mathbf{x}, \mathbf{y}) \propto \log p(\theta) + \log p(\mathbf{y} | \theta, \mathbf{x})$. The log of the hierarchical prior $p(\theta)$ is described above except for the prior of the noise variance which is $\sigma^{-2} \sim \text{Gamma}(\alpha_n, \beta_n)$ and the log marginal likelihood can be computed using Equation 2.1.

I choose a fully Bayesian inference procedure for the posterior distribution which promises much more robust results than e.g. Maximum Likelihood Type II especially in the case of a hierarchical prior and many parameters. The entire computations including the construction of the kernel are differentiable which makes it possible to use the Hamiltonian Monte Carlo algorithm described in Section 2.2.6. The output is a set of M samples $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_M\}$ that can be used to compute expectation values of any function of interest. For Bayesian prediction we can draw for each θ_i a few samples from the Gaussian process model $z_{*ij} \sim p(z_* \mid \theta_i, x_*)$ where $i = 1, \dots, M$ are the MCMC samples of θ and $j = 1, \dots, L$ are the samples of z_* for each θ_i . The samples z_{*ij} can then be used to compute summaries such as mean and standard deviation.

2.5 Experiments

The goal in this section is to test the performance of the H-NNGP with experiments. The most thorough way would be to (i) run all experiments for the most common variable selection methods in literature and compare to the H-NNGP approach, (ii) test for various performance metrics such as computer space and time, accuracy, variable selection, etc. and (iii) test with various data sets which each contain different challenges such as regression/classification, redundant variables, repeated variables, non-informative variables, small ratio of number of samples to number of variables, noise in the data, etc. However, a comparative study that includes all of these factors is infeasible for the scope of this thesis and I limit the experiments to the H-NNGP model for which I use accuracy, uncertainty estimates and variable selection as performance metrics and where I test for two different synthetic data sets with different numbers of informative, redundant, and repeated variables. For a comparative study of the most common variable selection methods see [51]. The results for the H-NNGP are contextualised by results on similar data sets in the NIPS 2003 variable selection challenge [70]. Using synthetic data before using real data has the advantage of testing the model in a more controlled setting where the set of true informative variables is known beforehand and various different experimental conditions such as number of non-informative variables can be tested. When the model performs well in terms of variable selection on the synthetic data there is more trust that it will also perform well on the real data

where the number of important variables is not known beforehand. To ensure a high degree of comparability between the synthetic Madelon data sets and the real Taxi data set, both problems are formulated as regression problems with continuous input variables, the number of features (20 for the s-Madelon and 100 for the l-Madelon) is comparable to the Taxi data (23 features), and the Madelon data tries to cover the most important variable selection challenges of real data: a mix of informative features, redundant features, and noise features as well as noise in the target variable. Eventually, I will apply the H-NNGP to real data in Section 3.

The H-NNGP is trained with Bayesian inference on versions of the Madelon data set and performance is compared for different model configurations. The two chosen versions of the original Madelon data set [71] are two binary classification tasks that together cover different challenges related to the number of redundant, repeated, and non-informative variables, the number of training samples, and the nonlinearity of the data. In the experiments the classification task is formulated as a regression task [72] which enables the use of a Gaussian noise model and therefore allows exact inference for the H-NNGP model (the typical cross entropy loss in classification could only be used with approximate inference which is not within the scope of this thesis). Similar to [31], the class labels are encoded as one-hot zero mean multidimensional regression targets; for the binary case that gives -0.5 when the class is incorrect and 0.5 when the class is correct. The design of the H-NNGP is as described in Section 2.4 where the multidimensional target is treated as multiple independent targets which leads to the addition of two log likelihood terms in the formulation of the posterior density. I test for various model configurations which are determined by the depth of the corresponding network L and the prior variance of the global shrinkage parameter σ_g^2 ; the remaining hyperparameters are fixed. The model parameters are then estimated with Bayesian inference using Hamiltonian Monte Carlo as described in Section 2.4; specifically, I use the original Hybrid Monte Carlo algorithm as described in [73]. I report the results of the experiments for prediction performance, uncertainty estimation performance, and variable selection performance for both data sets and various model configurations. The details for data sets, model configurations, inference configurations, and results are given below.

2.5.1 Data sets

The synthetic data sets that are used in the experiments are created with the template of the Madelon data set that was first proposed in the NIPS 2003 variables selection challenge [71]. The Madelon data set template covers many challenges that arise in variable selection problems such as redundant variables, repeated variables, non-informative variables, non-linearity of the data, and noise in the data. By adjusting the parameters of the template it is possible to generate versions of the original Madelon data set that each can focus on specific variable selection challenges. Madelon is a two-class classification data set with continuous input variables and a binary target variable. To generate the data I use the method of [74] which is designed to generate the data in [71]. The design of the data set consists of the following steps:

1. Create a number of Gaussian clusters in `n_informative`-dimensional space. Draw the points in a cluster from a multivariate Gaussian $\mathcal{N}(0, A)$ where A is a random covariance matrix with entries $A_{ij} \sim \mathcal{U}(-1, 1)$.
2. Position the clusters randomly on vertices of a `n_informative`-dimensional hypercube where the vertices of the hypercube are located at $\pm \text{class_sep}$.
3. Randomly assign an equal number of clusters to each class.
4. Add `n_redundant` variables that are created by multiplying the useful features with a covariance matrix B where $B_{ij} \sim \mathcal{U}(-1, 1)$.
5. Add `n_repeated` variables that are created by drawing randomly with replacement from the useful and redundant variables.
6. Add `n_noninformative` useless variables that are created with random noise $\mathcal{N}(0, 1)$.
7. Draw `flip_y` fraction of labels randomly which adds noise to the target.
8. Shuffle the order of variables and samples.

Based on this Madelon template I generate two data sets: one small and easy problem, denoted as small Madelon (s-Madelon), and one large and more difficult problem, denoted as large Madelon (l-Madelon). Table 2.1 gives an overview of the

parameter choices for both example data sets used in this thesis. We can see that l-Madelon is way more difficult than s-Madelon due to the larger number of variables and a smaller ratio of samples to informative variables. The l-Madelon comes close to the original Madelon for which performance results for various algorithms are reported in [70]. However, l-Madelon is still simpler than the original Madelon dataset which has 8000 datapoints with 500 features [71]. Testing the original Madelon dataset was not feasible because of too long runtimes of the Gaussian process model due to the matrix inverse problem described in this chapter. In future work this bottleneck could be solved with a scalable Gaussian process approach similar to [49].

Parameter	Description	s-Madelon	l-Madelon
n_class	Number of classes	2	2
n_clusters_per_class	Number of clusters per class	2	16
n_samples_per_cluster	Number of samples per cluster	50	25
n_informative	Number of informative variables	2	5
n_redundant	Number of redundant variables	2	5
n_repeated	Number of repeated variables	2	10
n_noninformative	Number of noninformative variables	14	80
class_sep	Factor multiplying hypercube size	2	2
flip_y	Fraction of random labels	0.01	0.01

TABLE 2.1: Parameter choices for the small Madelon data set and the large Madelon data set.

Figure 2.1a shows scatter plots for all pairs of informative, redundant, and repeated variables for s-Madelon. We can see that for the first two variables there is a XOR pattern, the fourth variable separates the classes reasonable well, and the fifth and sixth variables repeat the second and fourth variables. Figure 2.1b shows scatter plots for all pairs of informative, and redundant variables for l-Madelon. We can see that none of the two-dimensional variable combinations separates the classes reasonable well.

Figure 2.2 shows the heat map for the data matrix for s-Madelon and l-Madelon, revealing the clustered structure of the data.

Figure 2.3 shows the same data matrix as in Figure 2.2 but rows (samples) and columns (variables) have been randomly shuffled and rows have been grouped by class. We can see that it now visually appears uninformative.

FIGURE 2.1: (A) Correlation matrix for the informative, redundant, and repeated variables for the small Madelon data set. (B) Correlation matrix for the informative, and redundant variables for the large Madelon data set.

FIGURE 2.2: Heat map of the data matrix where rows correspond to data samples and columns correspond to variables. The last column corresponds to the class label. The color intensity represents the magnitude of the coefficient and the color represents the sign. (A) For the informative, redundant, repeated, and noninformative variables for the small Madelon data set. (B) For the informative, redundant, and repeated variables for the large Madelon data set.

I then draw 100 training samples and 100 test samples for the s-Madelon experiment, and 400 training samples and 400 test samples for the l-Madelon experiment. The training sets are used for the inference part and the test sets are used to report the performance.

FIGURE 2.3: Same matrix as in Figure 2.2 but rows and columns have been randomly shuffled and rows have been grouped per class.

2.5.2 Model configurations

The H-NNGP performance is tested for different choices of (L, σ_g^2) . The hyperparameter network depth, L , was tested for $\{0, 1, 3\}$, where $L = 0$ has zero hidden layer and is equivalent to a GP with a linear kernel. The hyperparameter σ_g^2 for the prior of the global shrinkage parameter τ was tested on log scale for $\{0.01, 1, 100\}$. The hyperparameters for the remaining priors are fixed: (i) $\alpha_w = 1$ and $\beta_w = 10^{-2}$ which gives a weakly informative prior for σ_w^{-2} , (ii) $\alpha_b = 1$ and $\beta_b = 10^{-2}$ which gives a weakly informative prior for σ_b^{-2} , (iii) $\nu_0 = 1$ and $s_0 = 1$ which results in a weakly informative Student- $t_{\nu_0}(0, s_0^2)$ prior for the parameters far from zero, and (iv) $\alpha_n = 10^{-6}$ and $\beta_n = 10^{-6}$ which gives an uninformative prior for the noise σ_n^{-2} .

2.5.3 Inference configurations

I use the Hybrid Monte Carlo algorithm for the model inference and follow the experimental configurations for Hamiltonian Monte Carlo in hierarchical models from [75]. In each experiment the Markov chain is started at the maximum a posteriori (MAP) where the MAP is found with the Adam optimizer. I then run warm-up iterations before the sampling begins. The sampling efficiency is determined by the parameters stepsize ϵ and the integration time T_* . Here the integrator step-size ϵ is adapted to reach the target Metropolis acceptance rate

$r = 0.8$ (as used in [76]) and the integration time is set by hand, approximately to the half-period of the oscillating trajectories, see Figure 2.6. The autocorrelations ρ_i are computed with the monotone sequence estimator [77] and the effective sample size is $\text{ESS}[\theta_i] = \frac{N}{1 + 2 \sum_{i=1}^N \rho_i[\theta_i]}$ where N is the number of samples.

Figure 2.4a shows the samples, the mean, and the one standard deviation bounds for the log of the global shrinkage parameter for a selected model configuration for s-Madelon. We can see that the chain still exhibits high autocorrelation but is constrained to a relatively small range in parameter space. Figure 2.4b shows the autocorrelations for the same Markov chain for different lags.

FIGURE 2.4: (A) The red dots show MCMC samples, and the blue line and band show the mean and $\pm$ standard deviation for $\theta_3^* = \log \tau$. (B) The dots show the autocorrelations of the same Markov chain for different lags.

Figure 2.5 shows MCMC samples for all local shrinkage parameters λ for a selected model configuration for the s-Madelon where the magnitude represents the importance of the connected variable. We can see two clusters where it would be reasonable to only select the two topmost variables in a variable selection process.

Figure 2.6 shows the Hamiltonian energy $H(\theta, q)$, the potential energy $U(\theta)$, and the kinetic energy $K(q)$ for one integration run for a selected model configuration for the s-Madelon. We can see that the trajectories are oscillating; half of the largest period of oscillation defines the optimal integration time for minimizing autocorrelation.

FIGURE 2.5: MCMC samples for all local shrinkage parameters λ_i with $i = 1, \dots, D$ for a selected model configuration for the small Madelon data set.

2.5.4 Results

The first part of results concerns the prediction performance. As the metric of prediction performance I report the accuracy on the binary classification task. Table 2.2 shows the prediction performances of the H-NNGP model in various configurations for s-Madelon and l-Madelon. The Gaussian process configuration is determined by the depth of the corresponding neural network, L , and the prior variance of the global shrinkage parameter, σ_g^2 . We can see that for s-Madelon the linear models with zero hidden layers have an accuracy close to 50 percent which, for a binary classification task, corresponds to guessing at random and is not surprising given the XOR classification task. The non-linear models reach almost perfect accuracy considering the noise in the data. The picture is similar for l-Madelon. The linear models perform slightly better than guessing at random since it's not a perfect XOR task anymore. The non-linear models perform slightly worse than for the s-Madelon due to the higher complexity of the classification task. Noteworthy, the classification accuracy is insensitive to the number of hidden layers as long as the model is non-linear ($L \geq 1$) and the classification accuracy is also insensitive to different global shrinkage parameters. This is likely due to simplicity of the synthetic data sets where non-linear patterns can be detected relatively

FIGURE 2.6: Trajectories of Hamiltonian energy H , potential energy U , and kinetic energy K for one integration run for a selected model configuration for the small Madelon data set. The trajectory was initialized at the MAP and is specified by integration length T_* and step size ϵ .

easy. Interestingly, the hyperparameter σ_g^2 that shapes the prior of the global shrinkage parameter τ has no significant effect on the prediction accuracies. This suggests that the parameter τ is mainly informed by the data and less effected by the prior. The authors of [70] report the results for the NIPS 2003 variable selection challenge for the original Madelon data set. These results could in theory be used as benchmarks, but the different data set configurations in our examples don't allow a direct comparison. Importantly, in the chosen Madelon version for this thesis the ratio of samples to informative features is much smaller in order to reduce computation time which makes the classification task and the variable selection task more challenging.

The second part of results concerns the uncertainty estimation performance. A great feature of the Bayesian formulation is that it allows to assign a variance to each test point prediction which encodes how certain the model is about a prediction. Figure 2.7a and Figure 2.7b show the correlation between uncertainty estimates and prediction error for a selected model configuration for s-Madelon and l-Madelon respectively. We can see that test points with a higher uncertainty estimate tend to have higher prediction error. This result is encouraging as the model seems to 'know when it doesn't know'.

The third part of results concerns the variable selection performance. Variables are selected based on the magnitude of the local shrinkage parameters λ , where a

Data set	Model	Accuracy
s-Madelon	H-NNGP-0-0.01	0.47
s-Madelon	H-NNGP-0-1	0.47
s-Madelon	H-NNGP-0-100	0.47
s-Madelon	H-NNGP-1-0.01	0.96
s-Madelon	H-NNGP-1-1	0.98
s-Madelon	H-NNGP-1-100	0.98
s-Madelon	H-NNGP-3-0.01	0.94
s-Madelon	H-NNGP-3-1	0.95
s-Madelon	H-NNGP-3-100	0.94
l-Madelon	H-NNGP-0-0.01	0.58
l-Madelon	H-NNGP-0-1	0.58
l-Madelon	H-NNGP-0-100	0.58
l-Madelon	H-NNGP-1-0.01	0.87
l-Madelon	H-NNGP-1-1	0.87
l-Madelon	H-NNGP-1-100	0.87
l-Madelon	H-NNGP-3-0.01	0.87
l-Madelon	H-NNGP-3-1	0.87
l-Madelon	H-NNGP-3-100	0.87

TABLE 2.2: Test accuracy for various model configurations for the small and large Madelon data set. The Gaussian process models are specified by (L, σ_g^2) .

large λ_i indicates that the connected variable i is relevant for explaining the observed data. In order to select variables a relevance threshold has to be defined. Importantly, the magnitude does not represent the relative relevance of individual variables but can only be used to distinguish between relevant and irrelevant, see Section 2.2.5. Figure 2.8a shows the posterior values for all local shrinkage parameters for a selected model for s-Madelon where the values correspond to the mean values shown in Figure 2.5. Additionally, the true informative, redundant, and repeated variables are marked. We can see that the model selects variables $\{2, 11\}$ which before shuffling correspond to variables $\{1, 2\}$ in Figure 2.2. The model therefore selected all true informative variables $\{1, 2\}$ without selecting redundant or non-informative variables. Similarly, Figure 2.8b shows the posterior values for all local shrinkage parameters for a selected model for l-Madelon. Additionally, the true informative, redundant, and repeated variables are marked. We can see that the model selects five variables and all of them contain helpful information, either by being informative (directly or by repeating an informative variable that was not selected) or by a combination of informative variables (redundant), without selecting non-informative variables.

FIGURE 2.7: Test points are sorted according to the predicted mean squared error (MSE) and then binned. The predicted MSE on the x-axis is the mean of the predicted variance for a bin and the y-axis is the actual MSE for the test points in that bin, (A) for the small Madelon data set with bin size 10, (B) for the large Madelon data set with bin size 50.

As outlined in Chapter 2.2.4, it can be computationally expensive to solve the chosen Gaussian process model when the data size is large. Table 2.3 shows the CPU times for s-Madelon and l-Madelon for various model configurations. The results show that the Gaussian process model is indeed computationally expensive. However, the presented CPU times most likely won't be a problem for applications in urban planning. To reduce the CPU time and enable larger data sizes, future work should explore scalable Gaussian Processes [49] which are based on approximation methods.

Data set	Model	Inference CPU time [s]	Prediction CPU time[s]
s-Madelon	H-NNGP-0-1	412	183
s-Madelon	H-NNGP-1-1	822	202
s-Madelon	H-NNGP-3-1	1486	235
l-Madelon	H-NNGP-0-1	3369	843
l-Madelon	H-NNGP-1-1	7035	1054
l-Madelon	H-NNGP-3-1	10054	1411

TABLE 2.3: CPU time for various model configurations for the small and large Madelon data set. The Gaussian process models differ in the number of layers. Inference was performed with 10000 samples with 10 leapfrog steps each. Prediction was performed with 50000 samples. For the computations a 2.4 GHz 8-Core Intel Core i9 was used and the reported CPU times are summed over all cores (and thereby represent the time a single core would have needed).

FIGURE 2.8: The blue line shows the posterior mean value of the local shrinkage parameters λ that determine the shrinkage of the connected variables, the colored dots show which of the variables are true informative, redundant, and repeated, (A) for a selected model for the small Madelon data set, (B) for a selected model for the large Madelon data set.

2.6 Summary

In this chapter, I introduced the concepts of Bayesian modelling, neural networks, Gaussian processes and sparse learning and reviewed related literature. I then presented a novel variable selection method for the “neural network as Gaussian process” approach based on the Horseshoe prior and tested its prediction, uncertainty, and variable selection performance on synthetic datasets. The synthetic datasets are chosen to have similar characteristics as the real world mobility datasets from Chapter 3 and help to systematically analyze the chosen modelling approach. I found that the developed approach is able to confidently predict the synthetic data while providing uncertainty estimates and correctly selecting the important variables. A limitation of the chosen approach is that it is computationally expensive when the data size increases.

Chapter 3

Application to spatial interaction modeling

3.1 Introduction

A fundamental interest of policy makers, urban planners, and transportation engineers is to understand how attributes of locations and transportation infrastructure relate to mobility flows in the city. Such understanding is crucial, for example, to estimate the impact of new urban developments on the mobility flows. However, developing such understanding is very challenging due to the large number of interacting factors such as spatial characteristics (e.g. distance between locations), temporal factors (e.g. varying trip purposes over the course of a day), demographic factors (e.g. population size of a location), and socio-economic factors (e.g. average income of employees living at a location).

To that end, human mobility models can help to quantify the relation between human mobility and the various factors and can assist decision-making in a complex environment. To be applicable in practice, the models have to have a certain predictive power and, at the same time, they have to be explainable since the stakes are high, decisions may affect many people, cost lots of resources, are difficult to reverse, and it is fundamentally important to be able to justify decisions to experts and the public. Model characteristics that can improve explainability are, for example, uncertainty estimates for predictions and importance estimates for input variables.

A very important class of human mobility models are spatial interaction models that describe the aggregates flow of people. The key input element for such models is the origin-destination (OD) matrix which contains all origin-destination pairs and for each pair stores the people flow between the two regions (e.g. two neighborhoods) per unit time (e.g. one day). The matrix entries can be estimated with a variety of methods such as transportation surveys, traffic counts, and individual devices. Spatial interaction models [7] then aim at reproducing the OD matrices based on a set of input variables. The standard procedure is to divide the region (e.g. city) into N regions (e.g. neighborhoods) and then predict T_{ij} , the number of travelers from region i to region j , based on two types of variables; one type of variables for attributes of the two regions i and j (e.g. population sizes) and one type of variables that relate the pair of regions ij (e.g. geographic distance). The difference between the models then often lays in the choice of variables and the functional form in which the variables enter the model.

There exists an extensive literature on spatial interaction modeling ranging from more traditional approaches to modern machine learning approaches. Traditional approaches are e.g. the gravity model and its variations which assume that the flow between two regions is a decreasing function of their distance [8], the intervening opportunities models which assume that the flow between regions is determined by the number of potential destinations between them [9], and the radiation model and its variations which assume that the destination for an individual travel is determined by its fitness (the quality of opportunity for the traveler) and the distance to the origin [78]. Due to the mechanistic nature of such models, the results are highly explainable which helps scientific understanding and builds trust for decision-making. However, with the increasing complexity of cities and as data are increasingly becoming available such simple models struggle to see all important relationships and patterns which then often leads to inaccurate predictions [15].

On the other hand, various machine learning approaches have been applied to spatial interaction modelling, especially decision tree models [79, 80] and neural network models [79–83]. The strength of such models lies in pattern recognition where the non-linearity of the models allows them to detect complex regularities in the data and improves the predictive performance. However, the existing models don't provide uncertainty estimates for their predictions, e.g. the user can not detect when the model is guessing at random. Another drawback is that neural

network models don't provide information on which input variables are important for making predictions. This is highly undesirable for urban planners which then had to make critical, expensive, not reversible decisions without being able to justify them to experts and the public. As a consequence machine learning models are not yet applied in practice.

Importantly, to be applicable in practice, the models have to have a certain predictive power and, at the same time, they have to be explainable. To that end, I apply the developed machine learning approach described in Chapter 2, H-NNGP, to the prediction of OD flows in the city. The input variables are the attributes of origin and destination regions such as population size, number of jobs, etc. as well as the attributes that describe the relation of origin and destination such as geographic distance, intervening number of jobs, etc. The target variable is the mobility flow between origin and destination.

I apply the model to a taxi ride data set from New York City where the task is to predict the aggregate OD taxi flows between census tracts for 2014 and 2015 based on a set of census tract variables (population size, number of jobs, ...) and a set of 'link' variables (geographic distance, number of intervening jobs, ...). In the best case the data set would contain data points that were collected before and after an urban development happened. Since such data is hard to obtain I instead opt for a quasi experimental setting where for the training set I randomly select mobility flows from 2014 and for the test set I randomly select mobility flows from 2015.

The approach is evaluated according to prediction performance, uncertainty estimation performance, and variable selection performance. By comparing the results obtained with the H-NNGP to traditional models I gain new insights into how urban attributes relate to mobility flows. The practitioner is then free to choose a complex model and test for many potentially relevant input variables which can help scientific understanding, facilitates the future improvement of the existing mechanistic models, assists decision-making and helps to communicate results to the public.

3.2 Related work

In this section, I introduce and discuss work that is related to spatial interaction modeling. I start by introducing models with a mechanistic nature and I refer to them as spatial interaction models. Specifically, I review the gravity model and its variations, intervening opportunities models, and the radiation model and its variations. It follows an introduction to machine learning approaches which have been applied to spatial interaction modeling. Specifically, I review various forms of decision tree models and neural network models.

3.2.1 Spatial interaction models

A great review on human mobility models, and spatial interaction models in particular, can be found in [7]. The extensive literature on spatial interaction models can be structured into gravity models, intervening opportunities models, and radiation models. In the following, I will present each approach in detail which includes the modelling framework, the inference method, as well as applications to real data sets.

I start by introducing the gravity model which was proposed by [8] and is based on Newton's law of gravitation. In its general form it can be written as

$$t_{ij} = c m_i m_j f(r_{ij}),$$

where t_{ij} is the number of travelers from region i to region j , c is a constant that depends on the overall number of trips, m_i is the total number of trips leaving i , m_j is the total number of trips attracted by j , and $f(\cdot)$ is a decreasing function of the distance r_{ij} . In the original form the masses are functions of the population size; here a common choice is $m_i = P_i^\alpha$ and $m_j = P_j^\beta$, where P_i and P_j are the population sizes of the regions i and j , and α and β are parameters of the model that are calibrated with data. Importantly, Wilson showed that the gravity model is not a single model but rather a family of models which can be generalized and extended [84]. For example, extended versions of the original gravity model can include additional variables such as the gdp-per-capita [85]. Often the function $f(\cdot)$ takes the form of an exponential or power-law $f(r_{ij}) = r_{ij}^{-\gamma}$ where γ is a parameter

that is calibrated with data. Extended versions of the original gravity model can also include additional variables such as travel time.

Next to the unconstrained gravity model outlined above, there are constrained versions of the gravity model that require additional input information but can improve the accuracy of the model. In the single-constrained gravity model the flow of individuals leaving i is given as $o_i = \sum_j t_{ij}$ and the gravity model is used to estimate the the flow to destination j as

$$t_{ij} = c_i o_i m_j f(r_{ij}), \quad c_i = \left(\sum_k m_k f(r_{ik}) \right)^{-1},$$

where c_i is the normalization constant. In the doubly-constrained gravity model additionally the number of travelers attracted by j is given as $d_j = \sum_i t_{ij}$ and the model can be written as

$$t_{ij} = c_i o_i l_j d_j f(r_{ij}), \quad c_i = \left(\sum_j l_j d_j f(r_{ij}) \right)^{-1}, \quad l_j = \left(\sum_i c_i o_i f(r_{ij}) \right)^{-1},$$

where c_i and l_j are normalization constants.

An important line of work also comes from Ira Lowry [86, 87] who developed computational models that describe the spatial organization of human activities in cities. The models were intended to be used from urban planners to evaluate the impact of their decisions concerning e.g. transportation infrastructure and urban renewal.

Another approach is the Huff model [88] which is a probabilistic formulation of the single-constrained gravity model and is often applied to location choice problems, e.g. where to locate a store. In this model the OD flow can be determined by the normalized attractiveness and accessibility of the destination. The proportion of flows originating in i and attracted by j is then given by

$$P_{ij} = \frac{m_j f(r_{ij})}{\sum_j m_j f(r_{ij})}.$$

Foundational work also comes from the field of urban economics which derived many of the above mobility models from a theory of individual, rational behavior, where each individual tries to maximize some utility function. That means,

for example, optimizing the place of residence and place of work with respect to commuting times and the presence of amenities nearby. Seminal work was done by Anderson and Van Wincoop [89] who derived the foundations for gravity equations of trade flows; more recent work in this direction was done by Monte et al. [90], Persyn and Torfs [91], and Agarwal [92] who used credit card transaction data to derive the foundations of consumption mobility.

In addition to the original framework of the gravity model and Huff model, there exists various extensions that account for spatial dependencies between regions and non-stationarity in the data that means when variables have a different impact in different regions [93, 94]. Such statistical measures can help to improve the accuracy of the models.

Below I discuss the inference part only for the traditional unconstrained version of the gravity model and neglect the constrained versions since this chapter is concerned with estimating mobility flows from the attributes of origin and destination regions as well as their relation (as in the unconstrained gravity model) rather than allocating mobility flows in the best way (as in the constrained gravity models). Michael Batty, for example, outlines various spatial interaction models and proposes several inference procedures for them [95]. In general, it is common to first log-transform the model, which then becomes linear in the parameters, and then apply ordinary least squares (OLS) to estimate the parameters. However, such procedure assumes normally distributed targets while OD flows are count data. To that end, Generalized Linear Models (GLM), e.g. the Poisson model, can provide more realistic probabilistic models [94].

The ability to estimate the traffic demand between two regions as a function of the attributes of the regions and their relation, and the simple and explainable functional form of the models, explains the popularity of the gravity models and Huff models for applications in urban planning [96], transportation planning [97], and location choice problems [98, 99]. However, the simple functional form is a strong simplification of reality which often leads to high prediction errors especially when predicting intra-city mobility flows [15, 100–102].

Intervening opportunities models are another type of spatial interaction model first introduced and discussed by sociologist Samuel Stouffer [9]. In those models distance is not a key predictor for travel demand anymore. Instead distance is replaced

by the number of intervening opportunities between origin and destination. Here an intervening opportunity is a potential destination of the traveler that is closer located to the origin than the actual destination but is not selected by the traveler. The intervening opportunities model can be also be written as a gravity model by replacing the distance factor $f(r_{ij})$ by a factor that takes into account the number of intervening opportunities between origin and destination, e.g. in power-law form $f(v_{ij}) = v_{ij}^{-\gamma}$ where γ is a parameter of the model that is calibrated with data. Such models were applied in urban planning [96] but they lost popularity over time since they have similar strengths and weaknesses as gravity models.

More recently the concept of intervening opportunities was used in another type of spatial interaction models called radiation models [78]. The general idea is that each traveler's destination decision can be formulated in a two-step process. In step one, each opportunity in each region is assigned a fitness score z (e.g. when searching for a new job the fitness score depends on factors such as income, working hours, etc.). where z is drawn from a distribution $p(z)$. In step two, the opportunities are ranked according to the distance to the origin and the traveler chooses the closest opportunity whose fitness score either exceeds a fitness threshold drawn from $p(z)$ or exceeds the highest fitness score obtained in the origin. It can be shown that when averaging the number of travelers between locations, the average flow is predicted by the radiation model [100],

$$t_{ij} = o_i \frac{1}{1 - \frac{m_i}{M}} \frac{m_i m_j}{(m_i + s_{ij})(m_i + m_j + s_{ij})},$$

where m_i is the number of opportunities at the origin, m_j is the number of opportunities at the destination, s_{ij} is the number of opportunities in a circle with radius r_{ij} centered at i , and $M = \sum_i m_i$ is the total number of opportunities. Originally, the number of opportunities was estimated with the population size of the region. The greatest strength of the model is that there are no parameters to calibrate which means it can be applied in the complete absence of data. However, it could be shown that the lack of parameters comes with great limitations and the model is not robust to different spatial scales [15, 100, 101]. To overcome this challenge, an extended version of the radiation model has been developed [103] which introduces a scaling parameter α . This scaling parameter determines the effect of the number of opportunities between origin and destination and improves the prediction performance of the model.

[15] makes a thorough comparison of the prediction performance of various spatial interaction models, including the production constrained gravity model, the attraction constrained gravity model, the doubly constrained gravity model, the intervening opportunities models, and the radiation models. The models are compared on various data sets with different temporal and spatial scales, specifically country level and city level. This work founds that overall the gravity models perform better than the intervening opportunities models and radiation models but they fail to predict flows for large distances. Importantly, the prediction performance of all models is generally low for intra-city mobility flows. The main reason is that in contrast to inter-city mobility flows there is more locational variation due to less aggregation (i.e. local characteristics do not smooth out as when looking across the entire city).

3.2.2 Machine learning models

More recently machine learning approaches showed promising results in predicting OD matrices. Early work from [104] aimed at the prediction of traffic OD matrices from flow data on road network links and its application to the real-time correction of OD matrices and dynamic traffic management. The developed model was a multilayer feed-forward neural network and Principal Component Analysis (PCA) was used to reduce the dimensionality of the input vector. The proposed neural network model outperformed all state-of-the-art OD prediction models. However, the PCA-transformed input variables are not as interpretable as the original variables and the model also does not provide any uncertainty estimates.

The work from [80] aimed at predicting human migration between US counties as well as migration between different countries. They compared (i) tree models, (ii) neural network models, and (iii) traditional spatial interaction models, which together included XGBoost, multilayer feed-forward neural network, gravity models and radiation models. They found that machine learning models in general outperformed traditional models in terms of prediction performance and that neural network models have the best prediction performance. Additionally, they found that more features than those used in the traditional spatial interaction models result in much better predictions. The importance of the various features was

determined based on information gain for the XGBoost algorithm but was not researched for neural networks. For example, for the between-counties data set, the most important feature was the number of intervening counties between origin and destination. However, the work did not apply the developed approach to intra-city data, did not determine feature importance for the neural network models and did not provide uncertainty estimates for any prediction.

The authors of [105] split the task of predicting the commuters network between Brazilian cities into two tasks: first, link prediction in the complex network using a binary classification model, and second, flow prediction on the identified links using a regression model. For each city 22 urban indicators were given that characterize the city. They compared various machine learning models and traditional spatial interaction models and found that the best approach was XGBoost which outperformed gravitation and radiation models in link prediction and flow prediction. To identify relevant features for link prediction and flow prediction they used SHapley Additive exPlanations (SHAP) values. Here SHAP values can explain individual predictions (which can be aggregated to gain model-level insights) by comparing the impact of the feature values of a given variable on the prediction to the impact of a baseline value. For the considered data set, the important features were distance, Gross Domestic Product (GDP), unemployment rate and a few more. However, the work did not apply the developed approach to intra-city data and did not provide uncertainty estimates for any prediction.

One of the first work that applied machine learning approaches to spatial interaction modelling *in the city* was done by [79]. In this work the goal was to predict commuting flows in New York City at the census tract level based on urban indicators that not only included ‘traditional’ data sources but also social media data. They compared tree models such as Random Forest, neural network models, and gravity models and found that the machine learning models outperformed the traditional models in terms of prediction performance. Feature importance was determined for the Random Forest model and the most important features were population size, distance between census tracts, the number of Twitter users, and employment rate. However, this work did not provide any feature importance technique for the neural network models and did not give uncertainty estimates for any of the considered models.

The authors of [106] aimed at predicting the impact of urban developments on the mobility flows in a city. They modeled the system as an attributed graph with node and edge features which encode regions in the city and the relationships between them. Their work focused on comparing different neural network architectures with traditional spatial interaction models. The comparison included fully-connected feed-forward neural networks, graph neural networks, random forest, gravity model, Huff model, and more. They found that the neural network approaches have a better prediction performance than the traditional models, and that graph neural networks do not substantially improve the prediction performance compared to feed-forward neural networks. However, no work was done on feature importance and uncertainty estimates.

In general, the previous work on machine learning in spatial interaction modeling presented above shows that machine learning models outperform traditional models in terms of prediction performance. The results especially encourage the use of neural networks when predicting intra-city flows. However, no previous work includes feature selection techniques for neural network models and no previous work provides uncertainty estimates for any of the machine learning models. Those two limitations greatly limit the applicability of machine learning to spatial interaction modelling in practice.

3.3 Contributions

In this chapter, I make the following contributions:

I apply the developed machine learning model, H-NNGP (see Section 2.4), to the prediction of intra-city OD matrices. The proposed approach uses the pattern recognition capabilities of deep neural networks while providing explainable predictions in terms of uncertainty estimates and variable selection. The model is based on the probabilistic formulation of an infinitely wide deep neural network as a Gaussian process. With the formulation as a Gaussian process I can explicitly model the uncertainty in the data set, as well as in the prediction function which is important for the critical decision making in the urban context. Additionally, the introduced hyperparameters in the covariance function of the Gaussian process

allow a principled way of automatic variable selection without the limitations of commonly used variable filters or wrapper methods.

The developed models are evaluated for prediction performance, uncertainty estimation, and variable selection on a real-world data set. Here the task is to predict the (yearly) aggregated taxi trips between New York City's census tracts based on attributes of the census tracts such as population size and number of jobs, and attributes of the relation between census tracts such as geographic distance and number of intervening jobs. I additionally analyze how the flow dynamics change over the course of a day. To that end, the selected variables from the different day chunks offer a better understanding about the trip purposes.

I compare various versions of the proposed non-linear machine learning model, H-NNGP, with traditional linear spatial interaction models in prediction performance, uncertainty estimation, and variable selection. The comparison between the different versions of H-NNGP shows the following:

- **Prediction performance:** The non-linear models clearly outperform the linear models. Further, for the non-linear models a higher number of hidden layers doesn't necessarily add to prediction performance.
- **Uncertainty estimation:** The more uncertain the models are about their predictions the larger the actual error will be.
- **Variable selection:** The linear models and non-linear models are both able to detect the predominant trip patterns but only the non-linear models can detect trips that diverge from predominant patterns and are related to more specialized places, such as trips to and from the airports.

The developed method allows for more accurate and explainable predictions than existing methods and the resulting insights can help urban experts to make decisions and communicate to the public, for example, to predict how a planned urban development in a given neighborhood will affect mobility flows between neighborhoods in the city. Importantly, the predictions come with the additional information about how certain the model is about its predictions and which urban measurements are important. These characteristics of the proposed greatly improve the existing models since model explainability is essential for critical decisions in the urban context.

3.4 Experiments

3.4.1 Data sets

In the human mobility literature one can find a variety of potential data sources which each cover different scales, location accuracies and trip purposes. For example, Census data mostly cover larger scales such as migration and mobility between counties. A notable exception is the US Longitudinal Employer-Household Dynamics (LEHD) data [107] for commuting trips on the census block level. However, the main limitation of Census data is that they focus mainly on migration flows and commuting trips and typically aggregate the data over one year. It makes these data extremely valuable for the traditional spatial interaction models that work with highly aggregated data but it is not sufficient to detect fine-grained trip patterns in space and time over various trip purposes (beyond commuting). Local travel surveys are an alternative which offers much more fine-grained data including a variety of meta data such as trip purpose and mode of transport. However, the big effort of collecting such data means that surveys are often only available in a local context, e.g. in one neighborhood, or only cover a small number of samples. For example, The Chicago travel survey [108] collected data of about 10,000 households. Yet another alternative data source are mobile phone records. This is an extremely valuable source of data since it typically allows to recover individual trajectories but at the same time can be scaled in space and time [109]. However, mobile phone data are not publicly available due to privacy reasons and commercial interests and they can be difficult to access for research purposes. Finally, GPS data provide the highest accuracies and allow to analyze individual trajectories. However, GPS data are often tied to a specific mode of transport and trip purpose (e.g. GPS data of Taxi trips).

For the following experiments I selected an extensive Taxi trip dataset as the main dataset that contains GPS information for most of the Taxi trips in New York City in 2014 and 2015. The Taxi dataset provides very fine-grained data and is a great fit to test if the developed model is able to detect hidden patterns within intra-city mobility flows. To that end, the small discretization of space and time in the Taxi GPS data allows for the detection of hidden trip purposes and patterns in addition to the most dominant trip purposes such as home-work trips which would not be possible with, for example, commuting statistics provided by the Census. The Taxi

data alone is certainly not representative of population movement in general since scales and modes of transport matter. However, the focus of this thesis is not so much on finding generalizable new insights about human mobility but rather on developing new methodologies that enable such insights. Moreover, taxi data cover various trip purposes besides home-work trips (e.g. nightlife-home trips, etc.), and due to the accuracy of GPS data they are a great resource to analyze if the developed algorithms are able to detect hidden patterns that could not be detected with existing methods. One promising approach for future work would be to enrich the GPS information with local travel surveys that would provide additional insights to trip purpose.

For the experiments, I create a data set that contains i) aggregated Origin-Destination (OD) flows of Taxi trips between New York City's (NYC) census tracts and ii) attributes data that describes the census tracts as well as the relations between the census tracts. The workflow to build the data set is as follows:

1. The city is subdivided into 2167 census tracts which follow administrative boundaries and for which shapefile data is publicly available [110]; each census tract represents an origin or destination in the OD-matrix.
2. The pick-up and drop-off times as well as the pick-up and drop-off locations are collected for all yellow and green taxis in NYC for 2014 and 2015 from the TLC Trip Record Data [111].
3. The latitude and longitude of origin and destination are converted into NYC census tract ids.
4. The trips are aggregated spatially (same origin and destination) and temporally (same year, weekday chunk, and hour chunk). Here weekday chunk subdivides into weekdays/weekends and hour chunk subdivides into 2-hour slots starting at 11pm. Figure 3.1 shows the mapping of a selection of binned OD flows. One can see that the frequency of trips for an origin-destination pair was highest in Manhattan and decreases with increasing distance from the city center; a noteworthy exception are the airports outside the city center. Figure 3.2 shows the probability distribution of OD flows aggregated over all temporal cases. One can see that the low-frequency flows are heavily overrepresented in the data set and that there is only a relatively small share of high-frequency flows.

5. The location attributes are selected in a systematic way with an heuristic approach. First, various trip purposes are defined (e.g. home-work); second, for each trip purpose, factors are included that relate to i) trip production, e.g. number of jobs with a specific income range living in the origin, ii) trip attraction, e.g. number of jobs with a specific income range located in the destination and iii) interaction affinity, e.g. distance. The final list includes the 22 attributes described in Table 3.1. Figure 3.3 shows the spatial

Trip Relation	Attribute
Attraction	Low-income-job individuals living at destination
	Medium-income-job individuals living at destination
	High-income-job individuals living at destination
	Low-income-job individuals working at destination
	Medium-income-job individuals working at destination
	High-income-job individuals working at destination
	Amenities at destination
Production	Low-income-job individuals living at origin
	Medium-income-individuals living at origin
	High-income-job individuals living at origin
	Low-income-job individuals working at origin
	Medium-income-job individuals working at origin
	High-income-job individuals working at origin
	Amenities at origin
Interaction	Distance between origin and destination
	Low-income-job individuals living closer to origin
	Medium-income-job individuals living closer to origin
	High-income-job individuals living closer to origin
	Low-income-job individuals working closer to origin
	Medium-income-job individuals working closer to origin
	High-income-job individuals working closer to origin
	Amenities closer to origin

TABLE 3.1: Location attributes related to attraction of trips, production of trips, and interaction between origin and destination.

distribution for three of the attributes: the number of low income jobs, the number of high income jobs, and the number of amenities. For example, one can see how low-income jobs cluster in the Bronx, high-income jobs cluster in the Upper East Side and amenities cluster at Lower Manhattan. Additionally, a synthetic attribute is added to the data set which represents a set of random samples from a standard normal distribution. This attribute allows to sanity-check the variable selection method and thereby gain confidence in the results.

The total number of attributes then adds to 23. For all job-related attributes the LEHD Origin-Destination Employment Statistics (LODES) data [112] is used which provides information to workforce dynamics. For the amenity related attributes OpenStreetMap [113] and overpass turbo [114] which is a web-based data filtering tool for OpenStreetMap are used. Importantly, all of the Points-of-Interests that are declared as amenities within OpenStreetMap have been included. In future work it might be interesting to further divide and analyze specific subsets of amenities, e.g. leisure related, consumption related, etc. This might give additional insights into the relevance of different types of amenities over the day.

6. The flow data and the attribute data are merged into one data table with the following columns: ID of origin, ID of destination, year, weekday chunk, hour chunk, number of trips, and the 23 attributes associated with the origin and destination.
7. The data table is filtered for weekday chunk (e.g. only weekdays) and for hour chunk (e.g. only trips occurring between 7am and 9am). Additionally, all trips where the origin equals the destination are filtered out and all rows with at least one missing value are filtered out.
8. The data table is split into a training set (all flows occurring in 2014) and a test set (all flows occurring in 2015). By using the data of the consecutive year as test data it is possible to simulate the practical use case of predicting origin-destination flows in the near future.
9. From the training set a small subset is selected that is used as training data. This simulates the use case of a limited access to training data. This is a likely scenario since it is very difficult for urban planners to get access to data that covers the entire population, for example, due to privacy concerns or the unwillingness of private companies to share their data. The training data is then uniformly sampled into equally sized data bins until each bin is filled. The data bin edges are set to [1, 10, 100, 1000, 10000] and the bin size is set to 250. The training set therefore contains 1000 data points. The test data is treated in a similar fashion but additionally an unbinned test set is created to simulate a real world application.

10. Training and test sets are finally log-transformed (which is common practice in spatial interaction modeling) and then linearly shifted to zero mean and scaled to one standard deviation.

FIGURE 3.1: Spatial mapping of binned and randomly selected OD flows for Taxi trips between New York City's census tracts in 2014 and 2015. The black dots represent New York City's census tracts and the colored lines represent the OD flow between them (the legend refers to the frequency bins).

FIGURE 3.2: Probability distribution of all OD flows for Taxi trips between New York City's census tracts in 2014 and 2015.

FIGURE 3.3: Spatial mapping of (A) the number of low income jobs living in each census tract, (B) the number of high income jobs living in each census tract, and (C) the number of amenities in each census tract. The black dots represent New York City’s census tracts and brighter colors correspond to higher numbers.

3.4.2 Model configurations

Similar in nature to the experiments in Section 2.5, I test the H-NNGP performance for different choices of (L, σ_g^2) . The hyperparameter network depth, L , was tested for $\{0, 1, 3\}$, where $L = 0$ has zero hidden layers and is equivalent to a linear model. Since the input data is log-transformed $L = 0$ corresponds to a Gravity model. The hyperparameter σ_g^2 for the prior of the global shrinkage parameter τ was tested on log scale for $\{0.01, 1, 100\}$. Since all initial results were insensitive to the choice of σ_g^2 it was set to 1 for all following experiments. The hyperparameters for the remaining priors are fixed: (i) $\alpha_w = 1$ and $\beta_w = 10^{-2}$ which gives a weakly informative prior for σ_w^{-2} , (ii) $\alpha_b = 1$ and $\beta_b = 10^{-2}$ which gives a weakly informative prior for σ_b^{-2} , (iii) $\nu_0 = 1$ and $s_0 = 1$ which results in a weakly informative Student- $t_{\nu_0}(0, s_0^2)$ prior for the parameters far from zero, and (iv) $\alpha_n = 10^{-6}$ and $\beta_n = 10^{-6}$ which gives an uninformative prior for the noise σ_n^{-2} .

3.4.3 Inference configurations

For the model inference I use the Hybrid Monte Carlo algorithm and follow the experimental configurations presented in Section 2.5.

3.4.4 Results

The first part of results concerns the prediction performance. In the following, I describe the metrics that I use to measure the prediction performance where z_i denotes the predicted flow for an origin-destination pair i , y_i denotes the actual flow, $|\cdot|$ denotes the absolute difference, and N denotes the number of pairs in the test data.

- Mean squared error (MSE). The mean squared error can be computed as $\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - z_i)^2$. All models with a Gaussian noise assumption ultimately try to minimize this error.
- Mean absolute error (MAE). The mean absolute error can be computed as $\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - z_i|$. Large differences between predicted and actual flows are penalized less compared to the mean squared error.
- Mean absolute percentage error (MAPE). The mean absolute percentage error can be computed as $\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - z_i}{y_i} \right|$. This error shows the model's accuracy with respect to the true flows.
- Common part of commuters (CPC). The common part of commuters can be computed as $\sum_{i=1}^N \frac{2 \min(y_i, z_i)}{y_i + z_i}$ and is often used in spatial interaction modeling.

Table 3.2 shows the prediction performance results from the experiments. It shows the four metrics (MSE, MAE, MAPE, and CPC), each for three different time periods (11pm - 1am, 7am - 9am, and 3pm - 5pm), each for different data bins (flows between 1 - 10, flows between 10 - 100, flows between 100 - 1,000, flows between 1,000 - 10,000, all bins together, and an unbinned subset of flows), each for different model configurations of the H-NNGP (with zero hidden layers which is equivalent to the linear Gravity model, with one hidden layers, and with three hidden layers). The winning model configurations are highlighted in bold. When focusing first on the CPC results for all three time periods (here CPC is only computed for the unbinned data as is common practice in spatial interaction modeling) we can see that both non-linear models clearly outperform the linear model. Further, we can see that for the non-linear models a higher number of hidden layers doesn't necessarily add to the prediction performance. This is also true for all other metrics for the unbinned data. We can see a similar picture for the binned data ('all') where

the non-linear models outperform the linear ones. Compared to the unbinned results discussed before, the MSE and MAE are much higher and the MAPE is much lower, which indicates the highly right-skewed nature of the flow data (with many small flows and a few very large flows). By zooming into the individual bins we can see the contributions of each bin to the overall errors. Generally, compared to the bins with smaller flows, the bins with larger flows contribute more to the MSE and MAE but contribute less to the MAPE. This shows that the models are in absolute numbers better in predicting small flows but in relative numbers are better in predicting large flows. With one exception, the non-linear models always outperform the linear models for all individual bins. Interestingly, we can also see that the errors for time period 11pm - 1am are much smaller than for the other two time periods 7am - 9am and 3pm - 5pm. The higher accuracy might be due to the more specialized trips (such as taking a cab back home from nightlife) where amenities cluster at certain locations and origin-destination patterns are more predictable.

Figure 3.4 shows the absolute percentage error (with brighter colors representing larger errors) for the binned test data for the three different time periods (row 1 shows 11pm - 1am, row 2 shows 7am - 9am, and row 3 shows 3pm - 5pm) for the three different H-NNGP versions (column 1 shows zero hidden layers which is equivalent to a linear Gravity model, column 2 shows 1 hidden layers, and column 3 shows 3 hidden layers). We can see the higher accuracy of the non-linear models (columns 2 and 3) compared to the linear Gravity models (column 1) for all three different time periods. The differences between one hidden layer (column 2) and three hidden layers (column 3) is not as pronounced. The maps allow additional insights about where (spatially) the improvements come from. Interestingly, the non-linear models seem to predict flows originating or ending at the airports (John F. Kennedy International Airport and LaGuardia Airport) much better than the linear models. I discuss this point in more detail below when analyzing the variable selection.

The second part of results concerns the uncertainty estimation performance. A great feature of the Bayesian formulation is that it allows to assign a variance to each test point prediction which encodes how certain the model is about a prediction. Figure 3.5 shows the correlation between uncertainty estimates and prediction error for the three different time periods (row 1 shows 11pm - 1am,

FIGURE 3.4: The maps show the absolute percentage errors of the predicted flows in New York City for the binned test data. The errors are shown for flows occurring between 11pm - 1am (first row), 7am - 9am (second row), 3pm - 5pm (third row) and for the H-NNGP with zero hidden layers/linear Gravity model (first column), one hidden layers (second column), and three hidden layers (third column). The colormap is normalized within each row and brighter colors indicate larger errors.

row 2 shows 7am - 9am, and row 3 shows 3pm - 5pm) for the three different H-NNGP versions (column 1 shows zero hidden layers which is equivalent to a linear Gravity model, column 2 shows 1 hidden layers, and column 3 shows 3 hidden layers). We can see that for all cases there is a monotonic relationship between uncertainty estimates and prediction error: the more uncertain the model is about it's predictions (high predicted MSE) the larger the actual error will be (high actual MSE). This characteristic is a very valuable feature when the models are used in practice and highlights the value of a Bayesian approach. For example, it allows urban planners to only rely on the model's predictions when the model is very certain. Interestingly, the relationship between predicted and actual MSE in non-linear and all models underestimate the MSE in small regions. In all regions, the non-linear models (column 2 and column 3) give much better estimates of the MSE compare to the linear models (column 1).

The third part of results concerns the variable selection performance. Variables are selected based on the magnitude of the local shrinkage parameters λ_i , where a large λ_i indicates that the connected variable i is relevant for explaining the observed data. In order to select variables it is necessary to define a relevance threshold. Importantly, the magnitude of λ_i does not represent the relative relevance of individual variables but can only be used to distinguish between relevant and irrelevant, see Section 2.2.5. Figure 3.6 shows the posterior values for all local shrinkage parameters for the three versions of H-NNGP with zero hidden layers (equivalent to the linear Gravity model), one hidden layers, and three hidden layers, for the three time periods 11pm - 1am, 7am - 9am, and 3pm - 5pm. The relevance values represent the mean values of the corresponding MCMC chains and indicate the importance of a particular feature. High relevance values in the non-linear models additionally indicate high non-linearity of the feature.

The following analysis of the results mainly focuses on the time period 7am - 9am (morning rush hours). We can assume that most of the trips happening throughout this time period are related to commuting and that the resulting mobility patterns are most comparable to the commuting patterns analyzed in existing spatial interaction studies. In the following, I will show that the developed models are able to discover hidden mobility dynamics that are not related to commuting and that could not have been detected with existing models. For the purpose of clarity, I don't interpret the absolute feature values but instead categorize them

FIGURE 3.5: Correlation plots of predicted mean squared error (MSE) versus actual MSE for the binned test data of flows in New York City. Means are computed by averaging over 50 data points. The correlation plots are shown for flows occurring between 11pm - 1am (first row), 7am - 9am (second row), 3pm - 5pm (third row) and for the H-NNGP with zero hidden layers/linear Gravity model (first column), one hidden layers (second column), and three hidden layers (third column).

into low/medium/high importance: low importance corresponds to values around zero, medium importance corresponds to values up to 20 percent of the maximum value (within a model category), and high importance corresponds to all values larger than 20 percent of the maximum value. Eventually, I use the features with high importance to infer the detected trip purposes and relate features to trip production, trip attraction and trip friction.

In Figure 3.6b, we can see that the *random* feature is zero for all models which means that all models pass this first sanity check. When zooming into the individual model results we can see that the linear Gravity model detects the home-work trip pattern. The features related to trip production are `opportunity_jobs_L_residence`, `opportunity_jobs_M_residence` and `opportunity_jobs_H_residence` which all relate to where the workers are living. The features related to trip attraction are `jobs_M_workplace_B` and `jobs_H_workplace_B` which relate to where the jobs are located. And the feature related to trip friction is `distance`. The remaining features with high importance that are not related to home-work trips are `jobs_H_workplace_A`, `amenity_A`, and `opportunity_amenity`.

For the non-linear models, we can see that the variable selection results are the same for the model versions with 1 and 3 hidden layers. This indicates that models with more hidden layers don't necessarily discover more patterns in the dataset. This result is in line with the performance results presented in Table 3.2. We can see that the non-linear models detect the home-work trip pattern as well. The features related to trip production are `jobs_M_residence_A`, `jobs_H_residence_A`, `opportunity_jobs_M_residence` and `opportunity_jobs_H_residence` which all relate to where the workers are living. The features related to trip attraction are `jobs_M_workplace_B`, `jobs_H_workplace_B`, and `opportunity_jobs_H_workplace` which relate to where the jobs are located. And the feature related to trip friction is `distance`. The non-linear models additionally detect trip patterns that start respectively end at the airports. The features related to these patterns are `jobs_H_residence_A` and `opportunity_jobs_H_workplace` for trips starting at the airports and `jobs_H_residence_B` and `opportunity_jobs_H_workplace` for trips ending at the airports (detailed explanation below). The remaining features with high importance that are not related to the previously mentioned trip patterns are `jobs_H_workplace_A`, `amenity_A`, `jobs_M_residence_B`, `amenity_B` and `opportunity_amenity`.

When comparing the linear Gravity model with the non-linear models, we can see that all models are able to detect the home-work trip patterns but only the non-linear models can detect trips that diverge from the predominant home-work patterns and are related to more specialized places such as trips to and from the airports. The linear model is not able to detect such hidden dynamics. Figure 3.7 shows the correlation pairplot for all input variables for 50k subsampled trips that happened in the morning rush hour between 7am and 9am. The correlation matrices provide a visual proof for existing non-linearities in the data. For example, we can clearly see strong non-linearities in the correlation matrices of row `opportunity_jobs_H.workplace` with columns `jobs_H.residence_A` and `jobs_H.residence_B`. This means there are many trips coming from/going to a location with very small number of high-income residents, and where many intervening job opportunities on the trip are not taken (during morning rush-hour!). This indicates a hidden dynamic not related to home-work trips which in this case are the trips originating from/ending at the airport. We can see this spatial dynamic very clearly in Figure 3.8. The top left shows where trips from locations with a low number of high-income residents start and the top right shows where these trips end. The majority of trips start from the airports (John F. Kennedy International Airport and LaGuardia Airport) and end all over the city. The linear models are not able to see this dynamic and perform very poorly for all trips coming from/ending at the airport. These results are inline with the residual maps presented in 3.4.

We can draw similar conclusions for the time periods from 11pm - 1am and 3pm - 5pm; only the trip purposes are different. For example, the predominant trip pattern for the time period 11pm - 1am are nightlife-home trips.

3.5 Summary

In this chapter, I applied the machine learning approach developed in Chapter 2 to the prediction of intra-city origin-destination matrices and compared it to linear models with respect to prediction performance, uncertainty estimation and variable selection. Specifically, the experiments were performed on aggregated taxi trips between New York City's census tracts for different times of the day. The comparison showed that the proposed non-linear approach clearly outperforms the

FIGURE 3.6: Feature relevance for predicting flows that occurred between 11pm - 1am (A), 7am - 9am (B), and 3pm - 5pm (C) for the linear model with zero hidden layers (H-NNGP-0, equivalent to the linear Gravity model), the non-linear model with one hidden layer (H-NNGP-1) and the non-linear model with three hidden layers (H-NNGP-3). The detailed description of each feature is given in Table 3.1.

FIGURE 3.7: Correlation pairplot with all input variables for trips happening in the morning hours (7am - 9am) and plotted for 50k data points. Brighter colors indicate a higher number of trips.

linear models in prediction performance while maintaining a high degree of trust in terms of uncertainty estimation. Interestingly, the linear as well as non-linear models are both able to detect the predominant trip patterns but only the non-linear models can detect trips that diverge from predominant patterns and are related to more specialized places.

FIGURE 3.8: Origin-destination maps with origins separated into three chunks (row 1, row 2, row 3) according to the number of high-income residents. The left column shows the origins and the right column shows the respective destinations. The colors represent the number of connected origins/destinations and thereby indicate the centrality of each location. Brighter colors correspond to a higher number of connected origins/destinations.

Time	Bin	Model	MSE		MAE		MAPE		CPC	
11pm - 1am	$10^0 - 10^1$	$L = 0$	179	(27)	5.7	(0.3)	2.95	(0.17)	-	(-)
11pm - 1am	$10^0 - 10^1$	$L = 1$	136	(49)	3.9	(0.2)	1.90	(0.09)	-	(-)
11pm - 1am	$10^0 - 10^1$	$L = 3$	129	(37)	4.0	(0.2)	1.92	(0.10)	-	(-)
11pm - 1am	$10^1 - 10^2$	$L = 0$	13913	(6842)	33.8	(2.0)	1.21	(0.10)	-	(-)
11pm - 1am	$10^1 - 10^2$	$L = 1$	3774	(797)	28.1	(1.1)	0.97	(0.04)	-	(-)
11pm - 1am	$10^1 - 10^2$	$L = 3$	3660	(641)	28.2	(1.1)	0.98	(0.04)	-	(-)
11pm - 1am	$10^2 - 10^3$	$L = 0$	343332	(34812)	258.4	(6.4)	0.97	(0.03)	-	(-)
11pm - 1am	$10^2 - 10^3$	$L = 1$	86273	(9191)	191.2	(7.7)	0.77	(0.03)	-	(-)
11pm - 1am	$10^2 - 10^3$	$L = 3$	82016	(7574)	189.5	(7.4)	0.77	(0.03)	-	(-)
11pm - 1am	$10^3 - 10^4$	$L = 0$	928468	(38985)	736.9	(10.6)	0.53	(0.01)	-	(-)
11pm - 1am	$10^3 - 10^4$	$L = 1$	347263	(11731)	443.4	(6.9)	0.31	(0.01)	-	(-)
11pm - 1am	$10^3 - 10^4$	$L = 3$	326449	(12087)	426.2	(7.5)	0.30	(0.01)	-	(-)
11pm - 1am	all	$L = 0$	321473	(7718)	258.4	(1.8)	1.41	(0.05)	-	(-)
11pm - 1am	all	$L = 1$	109362	(4727)	166.7	(3.3)	0.99	(0.03)	-	(-)
11pm - 1am	all	$L = 3$	103064	(4475)	162.0	(3.4)	0.99	(0.04)	-	(-)
11pm - 1am	unb.	$L = 0$	24	(2)	2.5	(0.1)	1.72	(0.07)	0.51	(0.01)
11pm - 1am	unb.	$L = 1$	17	(3)	1.9	(0.2)	1.14	(0.15)	0.60	(0.02)
11pm - 1am	unb.	$L = 3$	15	(2)	1.9	(0.2)	1.12	(0.13)	0.60	(0.01)
7am - 9am	$10^0 - 10^1$	$L = 0$	1174	(319)	11.5	(0.9)	6.36	(0.40)	-	(-)
7am - 9am	$10^0 - 10^1$	$L = 1$	771	(274)	6.7	(0.5)	3.54	(0.26)	-	(-)
7am - 9am	$10^0 - 10^1$	$L = 3$	810	(304)	6.7	(0.5)	3.55	(0.27)	-	(-)
7am - 9am	$10^1 - 10^2$	$L = 0$	16656	(2828)	49.2	(2.0)	1.82	(0.09)	-	(-)
7am - 9am	$10^1 - 10^2$	$L = 1$	22438	(7805)	45.6	(2.4)	1.60	(0.10)	-	(-)
7am - 9am	$10^1 - 10^2$	$L = 3$	24548	(8107)	46.6	(2.4)	1.64	(0.10)	-	(-)
7am - 9am	$10^2 - 10^3$	$L = 0$	419559	(58402)	285.9	(9.9)	0.91	(0.01)	-	(-)
7am - 9am	$10^2 - 10^3$	$L = 1$	161593	(11255)	253.1	(6.2)	0.89	(0.02)	-	(-)
7am - 9am	$10^2 - 10^3$	$L = 3$	158017	(10628)	252.9	(6.0)	0.90	(0.03)	-	(-)
7am - 9am	$10^3 - 10^4$	$L = 0$	3914048	(415887)	1280.0	(30.1)	0.67	(0.01)	-	(-)
7am - 9am	$10^3 - 10^4$	$L = 1$	1689091	(88888)	906.1	(21.3)	0.47	(0.01)	-	(-)
7am - 9am	$10^3 - 10^4$	$L = 3$	1641515	(84581)	885.9	(20.4)	0.45	(0.01)	-	(-)
7am - 9am	all	$L = 0$	1085868	(97241)	406.1	(7.5)	2.44	(0.11)	-	(-)
7am - 9am	all	$L = 1$	467710	(24895)	302.6	(7.0)	1.63	(0.01)	-	(-)
7am - 9am	all	$L = 3$	455499	(23352)	297.7	(6.6)	1.64	(0.01)	-	(-)
7am - 9am	unb.	$L = 0$	169	(14)	5.1	(0.2)	3.33	(0.11)	0.38	(0.01)
7am - 9am	unb.	$L = 1$	66	(2)	2.8	(0.1)	1.57	(0.07)	0.55	(0.01)
7am - 9am	unb.	$L = 3$	77	(3)	2.9	(0.1)	1.65	(0.08)	0.54	(0.01)
3pm - 5pm	$10^0 - 10^1$	$L = 0$	791	(172)	9.6	(0.5)	5.27	(0.35)	-	(-)
3pm - 5pm	$10^0 - 10^1$	$L = 1$	734	(274)	5.8	(0.5)	2.90	(0.31)	-	(-)
3pm - 5pm	$10^0 - 10^1$	$L = 3$	694	(244)	5.8	(0.5)	2.93	(0.32)	-	(-)
3pm - 5pm	$10^1 - 10^2$	$L = 0$	14147	(4847)	43.6	(2.5)	1.73	(0.10)	-	(-)
3pm - 5pm	$10^1 - 10^2$	$L = 1$	9247	(2032)	35.6	(1.5)	1.34	(0.07)	-	(-)
3pm - 5pm	$10^1 - 10^2$	$L = 3$	9467	(2178)	36.0	(1.6)	1.35	(0.06)	-	(-)
3pm - 5pm	$10^2 - 10^3$	$L = 0$	646645	(183725)	298.2	(15.7)	0.94	(0.03)	-	(-)
3pm - 5pm	$10^2 - 10^3$	$L = 1$	128027	(9896)	217.2	(6.0)	0.76	(0.02)	-	(-)
3pm - 5pm	$10^2 - 10^3$	$L = 3$	124870	(8893)	216.0	(5.8)	0.76	(0.02)	-	(-)
3pm - 5pm	$10^3 - 10^4$	$L = 0$	3434681	(395363)	1208.0	(28.8)	0.71	(0.02)	-	(-)
3pm - 5pm	$10^3 - 10^4$	$L = 1$	1183909	(31780)	731.7	(10.6)	0.41	(0.01)	-	(-)
3pm - 5pm	$10^3 - 10^4$	$L = 3$	1120180	(34112)	706.1	(11.0)	0.40	(0.01)	-	(-)
3pm - 5pm	all	$L = 0$	1023451	(108364)	389.6	(8.3)	2.17	(0.09)	-	(-)
3pm - 5pm	all	$L = 1$	330176	(8474)	247.4	(2.4)	1.35	(0.07)	-	(-)
3pm - 5pm	all	$L = 3$	313529	(9196)	240.8	(2.5)	1.36	(0.08)	-	(-)
3pm - 5pm	unb.	$L = 0$	118	(13)	4.7	(0.3)	3.00	(0.15)	0.40	(0.01)
3pm - 5pm	unb.	$L = 1$	101	(20)	3.1	(0.2)	1.73	(0.15)	0.52	(0.02)
3pm - 5pm	unb.	$L = 3$	110	(21)	3.3	(0.2)	1.81	(0.15)	0.51	(0.02)

TABLE 3.2: Prediction performance results for different times of the day, data bins, and model configurations. MSE is rounded to integers, MAE is rounded to the first decimal place, and MAPE and CPC are rounded to the the second decimal place. Standard errors are shown in brackets and the winning model configurations are highlighted in bold.

Part II

Predicting neighborhood change

Chapter 4

Spatial interactions as predictor of neighborhood change

The work in this chapter has been published in [25]. The original paper version has been restructured to fit into the format of the thesis. The uncertainty component of the thesis is not the focus of this Chapter. The focus is rather on the complex systems component. While Chapter 3 presented a ‘top-down’ view on spatial interaction modelling (i.e. how space characteristics determine people’s movement), this Chapter presents a ‘bottom-up’ view on the same topic (i.e. how people’s movement determine space characteristics).

4.1 Introduction

Cities around the world continue to attract additional residents who, in many cases, can be quite discerning about what type of neighborhood they want to live in. As a result, the housing market in many urban neighborhoods has experienced increased pressure, often leading to escalating prices and the loss of affordable housing for urban residents. However, despite far-reaching consequences, it remains difficult to predict when and where neighborhoods experience these pressures. Previous work has often evaluated the characteristics of the neighborhood itself: e.g., distance to employment centers, schools, restaurants and other amenities, crime level, and a wide array of demographic data such as income, education and age of its residents [22]. However, collecting such data is resource intensive, which often results in data

only collected decennially or, in a best case, annually. This limits the applicability in analysis of urban processes that operate and change on a much finer temporal scale, such as the early prediction of rising housing prices in urban neighborhoods. While recent research has been increasingly looking at monitoring the city in near real-time by making use of new types of user-generated data [1], the link between collecting and monitoring this type of data and its actual use for predicting future changes in urban social processes remains understudied.

“What can the canary in the coal mine tell us? Historically, canaries accompanied coal miners deep underground. Their small lung capacity and unidirectional lung ventilation system made them more vulnerable to small concentrations of carbon monoxide and methane gas than their human companions. As late as 1986, the acute sensibility of these birds served as biological indicator of unsafe conditions in underground coal mines in the United Kingdom. [...] All species (or species assemblages) tolerate a limited range of chemical, physical, and biological conditions, which we can use to evaluate environmental quality.” [115]

In this chapter, I apply the concept of bioindicators, borrowed from ecology [115], to a social science context. The overall objective is to use an ‘indicator group’ of people to assess the ‘quality’ of a neighborhood and its change over time. Conventional approaches typically conduct social, economic and infrastructural studies *on the entire population* to directly measure urban parameters. Instead, I use a much smaller *subset of the population* – the indicator group – as a proxy for a neighborhood’s condition. The starting premise is that certain subgroups within the population (e.g., young affluent singles) have a different tolerance level for certain neighborhood characteristics than others (e.g., less affluent families). Just like the canary was employed as a practical proxy for levels of carbon monoxide and methane, so the presence or absence of such a subgroup can be used as a practical proxy for difficult-to-measure neighborhood characteristics that might precede rising housing prices (e.g., the ‘hipness’ of an area). Instead of trying to measure, for example, trendy restaurants, I monitor the people who might make use of them.

The underlying assumption of my approach is that a change in human mobility patterns often precedes physical neighborhood change by several years. Taking gentrification as an example, the influx of artists into a deprived neighborhood might happen years before the neighborhood becomes hip and attracts the interest of investors. The hypothesis is that by monitoring changing mobility patterns of a

carefully selected indicator group, one can predict neighborhood change years into the future. In contrast, the presence of amenities is only changing years after the bottom-up change already has started and can therefore not be used to predict neighborhood change at such an early stage.

The use of indicator groups is fundamentally different from more conventional measures of neighborhood characteristics and offers a key advantage. In contrast to location-based measures such as census data, it is an individual-based measure that allows to track in near real-time how people make use of urban space. This, in turn, allows to detect changing usage patterns *before* they are manifested in census data. An illustrative example are neighborhoods that attract more young affluent singles before housing prices are increasing [116]. Thereby, seemingly small events may have a large impact on these usage patterns. For example, a few isolated crime incidents can seem insignificant in the official statistics (and might not even be easily detected), but might have a drastic effect on the neighborhood's attractiveness. As such, the specific 'tolerance' of an indicator group to such small incidents can provide early warning signals of potentially strong changes in the neighborhood characteristics.

While bioindicator studies in ecology often depend on the development of systems that identify and track specific, individual species, urban studies differ from this in a significant way: human beings all belong to the same species. As such, the challenge is not selecting a pre-defined group but rather defining a specific subset, or indicator group, from within the whole population. The great advantage of the current computational landscape is that human beings, through their engagement with 'smart' devices and the Internet, have become 'sensors' themselves and are now continuously producing a wide variety of data about their behavior. Many of today's computing applications, social media platforms being a prime example, allow users to connect with each other and share content as well as their location, allowing the extraction of the 'who' and the 'where' at an unprecedented scale. In contrast to classic census or survey studies, social sensors can provide a high spatial and temporal resolution, allowing studies on different spatial scales in near real-time [117]. All of this data allows us to define and adapt appropriate and precise indicator groups for different social phenomena.

In this chapter I develop a methodology to monitor neighborhoods based on the characteristics of its visitors, allowing me to predict neighborhoods with expected

rent increase in subsequent years. I combine the bioindicator concept, borrowed from ecology, with social sensing data from location-based social networks. In the present work, I use data from Twitter, which allows me to study the spatial behavior of 1.6 million users in the US cities Chicago, New York City, Los Angeles, Boston and Portland over the course of one year (July 2012 - June 2013). During this time, these users created 92 million geotagged tweets. I set up a bipartite network of user nodes, location nodes and visitor links, allowing me to measure how often each user tweeted within each neighborhood. Based on this, I can create a profile for each user based on the socio-economic characteristics of the visited neighborhoods. Specifically, I assign income and age profiles that are then used to identify appropriate indicator groups. In the following, I refer to the users outside the 90 percent quantiles as high-income-profile and low-income-profile users and high-age-profile and low-age-profile users respectively. A given user with a high-income profile might not necessarily have a high income himself but just *visits* high-income neighborhoods more often. I find that the number of visitors with a high-income (low-income) profile shows a strong positive (negative) correlation with the neighborhood's rent level. Further, I find that the number of high-income-profile visitors between July 2012 and June 2013 is strongly correlated with an increase of the rent values over the *following* years (2012 - 2015). Importantly, by examining a number of neighborhoods that are currently known for their increasing rent levels and associated socio-economic changes, I find that those areas tend to have a disproportionately low rent level compared to other neighborhoods with a similar number of high-income-profile visitors. Finally, I further enhance the indicator group by including the age profile of the users. Here, I find that users with a high-income, low-age profile are especially well-suited as an indicator group: their visits to a neighborhood are an early predictor of drastic future rent increase. The adaptation of the concept of bioindicators to urban residents, monitored through social sensing data, allows for the development of a broad range of applications with relevance for various stakeholders from urban planning and governance to the real estate industry. While I focus on rising rents in neighborhoods in this chapter, my methodology is completely generic and can potentially be applied to a broad range of urban challenges, such as detecting social segregation processes or predicting spatio-temporal crime patterns in cities.

4.2 Related work

The current work builds forth on three distinct domains of related work. The first is a long-standing tradition of studying urban processes and urban change. Why and how does a city, its neighborhoods and its people, adapt and evolve? Second, and related, is the domain of social sensing that looks at these questions through the lens of the many novel data sources and computational methods that have emerged in the last twenty years. Although this new wave of data may overcome shortcomings of conventional urban research (e.g., lack of granular data), social sensing has its own set of challenges when trying to understand, predict and, ultimately, govern the dynamics of cities. It is here that I introduce the third domain, adapted from ecology, and posit the concept of bioindicators as a useful method in the practical application of social sensing.

4.2.1 Urban change and Housing

There is a long tradition of studying urban change; an extensive overview is given in the Urban Regeneration Handbook by Roberts and Sykes (2016) [118] which discusses a wide range of aspects ranging from social and community issues to housing to monitoring and evaluating urban change. Much work within the social sciences [119–121] has specifically focused on the relation to housing. To that end, indicators can range from median household income and educational attainment to housing prices and the number of amenities. Generally speaking, these studies try to understand why and how urban change takes place [122] and which neighborhoods are currently undergoing change [123]. Here a growing literature comes from the field of urban economics which typically analyzes neighborhood change with mechanistic models. For example, Guerrieri et al. (2013) [124] links housing price movements across neighborhoods with gentrification in those neighborhoods. Here the developed model generates an equilibrium that describes segregation based upon income. Couture et al. (2020) [125] analyzes neighborhood change from a similar lense and developed a model that quantifies the impact of gentrification on well-being inequality. Another tradition has focused on understanding and modeling neighborhood change as a process, often using ecological concepts such as the invasion-succession model [13, 14]. Recently, this approach has been reinvigorated

with computational methods adopted from across disciplines that allow for an empirical analysis of neighborhood trajectories through time, such as Self-Organizing Maps [126] and genetic sequencing [127]. However, most studies still rely on census data or other conventional survey instruments and track socio-economic indicators aggregated to the neighborhood level.

4.2.2 Social sensing

Answering the question of *‘who’ moves ‘where’ and ‘when’ in the city* is of great interest for social and economic research and policy. Novel sources of user-generated data such as those automatically collected from social media or mobile phones have proven to provide valuable new insights into the organization of cities [2, 128–130]. This geo-referenced data is unprecedented both in terms of the number of people it covers and in terms of its spatiotemporal resolution, which also enables new approaches to the demographic profiling of people, often summarized under the label ‘social sensing’ [131]. Blumenstock et al. [132] recently demonstrated that feature engineering techniques allow for the inference of different socio-economic characteristics of individuals from their anonymized mobile phone usage patterns. Calabrese et al. [133] inferred the home location of the user to then assign socio-economic census data. Other approaches go beyond only using the spatio-temporal aspects of this data and analyze other variables contained within user-generated data. For example, Facebook Likes can be used to make predictions about the age, gender, and ethnicity of the user [134] and the content of tweets can be used to predict the occupational class of the user [135].

Specifically related to the application in this work, such data can be applied to the study of human mobility patterns at different spatio-temporal scales: from global studies of migration [136] to mobility at national scales [137, 138] and finally at the city level [139–141]. Such datasets have already seen varying applications from epidemiology [142] to spatial interaction [141] and traffic modeling [143]. Finally, multiple data sources (e.g., Foursquare, Twitter, Instagram) can be combined to construct mobility and demographic profiles for social media users [144, 145]. These studies have enabled great methodological advances towards a near real-time monitoring of cities - although a widespread adoption of applications of these methods has still to emerge.

4.2.3 Indicator group

In ecology the concept of bioindicators is often employed to study the quality of environments – and monitor potential changes. Although bioindicators may contain a variety of biological processes, a single species or a collection of species is usually taken. Not every species makes for a good bioindicator: bioindicators have a very low tolerance to changes in a particular factor of interest to the researcher; much lower than other species in the same environment [115]. Different bioindicators can be identified and developed for different types of environments. An overview of the concept, including methodology, benefits and disadvantages, is given by Holt et al. [115] while Siddig et al. [146] provide a thorough overview of how ecologists have selected, used, and evaluated the performance of bioindicators over the last 15 years.

Analyzing only a subset of the population is not an entirely new idea and has seen uptake in the social sciences before [147]. But to the best of my knowledge, the concept of bioindicators or, more generally, indicator groups has not yet been used to study the change of urban neighborhoods. Perhaps the closest adaptation is within the field of geodemographics [148]. Geodemographics aims to segment the population into smaller subgroups that go beyond traditional indicators (e.g., age, education) and more towards 'lifestyle'. In commercial applications this can be used for customer targeting [149, 150] but it can also be used for the better targeting of public policy [151]. Although the use and identification of these segments is not dissimilar from the construction of indicator groups, geodemographic segmentation has been exclusively used in an applied manner (e.g., for customer targeting) rather than the use of a segment as an indicator group in subsequent scientific research.

4.3 Contributions

In this chapter, I make the following contributions:

I develop a novel methodology to monitor neighborhoods based on the characteristics of its visitors, allowing me to predict neighborhoods with expected rent increase in subsequent years. As the core idea, I combine the bioindicator concept, borrowed from ecology, with social sensing data from location-based social networks.

The significance of the developed method is tested on real data sets including Twitter data and census data. The Twitter data allows me to study the spatial behavior of 1.6 million users in the US cities Chicago, New York City, Los Angeles, Boston and Portland over the course of one year (July 2012 - June 2013) and subsequently create a profile for each user based on the socio-economic characteristics of the visited neighborhoods.

I find that the proposed method can be used for the early detection of rising rents in urban neighborhoods. This suggests that the adaptation of the concept of bioindicators to urban residents, monitored through social sensing data, is of great relevance to various stakeholders from urban planning and governance to the real estate industry. The proposed methodology is completely generic and can potentially be applied to a broad range of urban challenges, such as detecting social segregation processes or predicting spatio-temporal crime patterns in cities.

4.4 Method

Figure 4.1 presents the developed monitoring model that allows to quantify the visiting frequency of an indicator group to neighborhoods. An important function of the model is to describe the relationship between the two distinct sets of users on the one side, and neighborhoods on the other side. To that end, I set up a bipartite graph $G = (\top, \perp, E)$ where $\top$ is the set of top nodes, $\perp$ is the set of bottom nodes, and $E \subseteq \top \times \perp$ is the set of links. The top layer consists of a set of nodes $u \in \top$ that represents a set of N users where each user u_n can be described with a unique user id. The bottom layer consists of a set of nodes $l \in \perp$ that represents a set of M neighborhoods where each neighborhood l_m can be described by a set of polygons with vertices described by coordinates. A set of links $v \in E$ represent the visits of users to neighborhoods where $v_{n,m}$ can be described as the number of visits of user u_n to neighborhood l_m . A visit is made when the user was spatially located in the neighborhood. For the definition of $v_{n,m}$, I use a temporal resolution of one day. Therefore, several (potentially) distinct visits of a given user to the same neighborhood during the same day (midnight - midnight) are treated as one single visit.

FIGURE 4.1: Model for monitoring indicator group. 2-mode network model that consists of a set of user nodes in the top layer (indicator users, green), a set of neighborhood nodes in the bottom layer, and a set of links representing the visits of users to neighborhoods. Adapted from [1].

Each node u_n has a set of K node attributes represented by column vector A_n of size K and is denoted as the user profile. The user profile is evaluated based on the visited neighborhoods:

$$A_n = \sum_{m=1}^M w_{n,m} B_m$$

where B_m is a column vector of size K that represents a set of K node attributes of node l_m and is denoted as the residents profile, and $w_{n,m} = v_{n,m} / \sum_{m=1}^M v_{n,m}$ normalizes the number of visits of user u_n .

This allows me to segment users based on their spatial profile and define the indicator group I as a set of L indicator users where each indicator user u_l fulfills the indicator group requirements for all attributes K . Finally, I define for each neighborhood l_m the visitor profile p_m as the number of unique indicator group visitors in a given time period.

4.5 Experiments

4.5.1 Data sets

In total, three different types of datasets were used for the experiments. The first type of dataset consists of geotagged Tweets with a spatial resolution of GPS-based latitude/longitude coordinates that were collected through the Twitter streaming API [152]. These data were used to describe the mobility patterns of millions

of individuals. The second type of dataset consists of socio-economic attributes, specifically income and age distributions, of all neighborhoods in a city with a spatial resolution of census tracts [153]. These data were used to describe the socio-economic characteristics of the places that someone visited. The third type of dataset consists of the median rent of all neighborhoods in a city with a spatial resolution of census tracts [154]. These data were used to describe the housing prices of neighborhoods and were used as proxies for neighborhood change. Following, all three datasets are described more in detail.

Visitors. The availability of new types of data collected from human beings or devices on their behalf opens up the opportunity to monitor the movement of people and therewith the ‘pulse’ of the city at an unprecedented scale near real-time. Here, mobile devices function as sensors that can provide information about the ‘who’, the ‘where’ and the ‘when’ of urban life. For this work, I used a dataset of 92 million geotagged tweets through the Twitter streaming API [152]. The tweets were posted by 1.6 million users across five US cities, namely Chicago, New York City, Los Angeles, Boston and Portland (spatial boundaries as defined by the Metropolitan Statistical Areas), in the time period between July 2012 and June 2013. Each geotagged tweet contains information about the user described by an anonymized user ID, the location from which the tweet was sent as lat/lon coordinates and the time at which the message was created.

Residents. For socio-economic neighborhood attributes, as well as the spatial delineation of neighborhood boundaries, data from the U.S. Census Bureau is used. The data from the Census Bureau provides an extensive snapshot of the characteristics of urban residents, based on a statistically representative sample of the society, and also covers the most relevant socio-economic domains. Specifically, I employ a dataset based on the 5-Year Estimates from the American Community Survey (ACS) that contains the median household income and median age in a neighborhood for the year 2012. For the neighborhood boundaries, I use the census tract definition, which are areas with a population between 2,000 and 8,000 residents.

Housing prices. Collecting accurate, comparable and reliable housing price data over the period of several years on a small spatial scale such as the census tract level is a difficult task as transaction prices and rental values are often not publicly available. For this work, I again employ a dataset based on the ACS 5-Year Estimates, which contains the median contract rent on census tract level for the years 2012 and 2015. I do need to note here that a drawback of ACS data at this spatial scale is its large margins of error on census tract level, a result of a relatively small sample size. In the following, I refer to the census tracts as neighborhoods.

4.5.2 Selection of the indicator group

Challenges. The first challenge is the selection of an appropriate indicator group for the phenomenon of interest. Not all social groups can serve as successful indicator groups for the same phenomenon and, more so, social groups are not discrete, natural units like species.

The second challenge is the derivation of a user profile based on human activity data. Here, recent research made substantial advances towards an accurate and reliable feature extraction, for example, through the inference of different socio-economic characteristics from mobile phone data [132] or social media data such as Facebook Likes [134]. A common difficulty is the access to fine-grained geotagged data at the individual level. For this work, I employ a large dataset of geotagged tweets, which are publicly accessible through the Twitter streaming API. However, when zoomed into the individual and neighborhood level, geotagged tweets become relatively sparse. Combined with the fact that only a small subset of Twitter users opts-in to sharing their location [152, 155], the inference of profiles for a broad set of users, let alone an entire population, is especially challenging.

Finally, I need to monitor the indicator group's activity across the city and find an appropriate measure that can be used as a predictor for the housing prices.

Method. Previous studies have shown that urban change is strongly related to socio-economic attributes of the neighborhood, such as income or age profiles [156]. These studies give me a set of relevant domains for neighborhood change predictions but they lack a specific recipe for selecting indicator groups.

As a proof-of-concept, I define income as the first indicator attribute and select a specific income range as the indicator feature. I select the indicator group according to the income-profile distribution for all users in the dataset. To test the accuracy of the indicator group selection, I define a reference group of similar size but with a different income range. Exemplary, and to showcase the different performances of different data subsets, I define the 10 percent ‘richest’ users as the indicator group, denoted as high-income-profile users, and the 10 percent ‘poorest’ users as the reference group, denoted as low-income-profile users. It is important to note that it is not my goal to provide an explanation for the underlying complex process of rising housing prices [157], but rather to have available a simple but powerful indicator that signals potential future changes in rents before they are manifested in, for instance, census data.

To estimate an income profile, I employ a relatively simple model that characterizes users based on the characteristics of the neighborhoods they visited. This makes theoretical sense as well: for the prediction of neighborhood change, not only the actual income of visitors to a neighborhood is relevant but also their tastes and preferences. These tastes and preferences are proxied by the character of the other neighborhoods they visit. Specifically, to each user I assign the weighted median household income of visited neighborhoods, where the weight is the number of visits to the neighborhood divided by the number of visits to all neighborhoods. I define the number of visits as the number of distinct days in one year a user was geolocated in the neighborhood. Due to the sparseness of geotagged tweets, I used the visit history of a user over the course of an entire year (July 2012 - June 2013). Further, I excluded users with less than 10 visits for this time period.

Finally, I need a measure that reflects how frequently the indicator group is attracted to each neighborhood. To do so, I define the indicator measure as the number of unique indicator visitors in a neighborhood in a given time period. By counting the number of unique indicator visitors, rather than their total visits, I avoid an over-representation of power users. The indicator measure is independent of the time of visit, day in the week, duration of stay, and frequency of visits. It simply reflects how many unique high-income-profile users a neighborhood attracted over the course of one year.

When comparing the indicator measure to housing prices at the neighborhood level, I face the challenge that ACS data at this spatial scale has relatively large margins

of error. It gets even more difficult when analyzing the housing prices over time, as the errors for the same neighborhood but for different years are independent. To resolve this issue and to make correlations visible, I use neighborhood bins instead, which consequently reduces the variability. Bin edges represent thresholds for the number of visitors and are spaced evenly on a log scale, while the bin value represents the median of all included samples within those edges. Neighborhood bins that contain less than 10 neighborhoods are removed. I denote the resulting number of neighborhood bins as N .

Indicator group statistics. Table 4.1 shows the descriptive statistics of the monitoring model based on the indicator group in five US cities. Taking Chicago as an illustrative example, I identified 8,238 users as indicator users, which represents 10 percent of the entire user population. Together they visited 2,032 different neighborhoods. In the period between July 2012 and June 2013, they sent around 1.52M tweets which translates into around 0.48M visits or 0.12M network links. Hence, on average, an indicator user visited around 14 different neighborhoods, and each of those about 4 times per year.

TABLE 4.1: Network statistics for indicator group.

City	# Users	# Neighb.	# Tweets	# Visits	# Links
Chicago	8238	2032	1 516 730	478 507	118 081
NYC	11 284	1554	1 430 679	469 957	115 696
LA	13 638	2730	2 219 601	712 494	201 468
Boston	3706	579	605 596	202 319	37 755
Portland	1546	426	311 468	83 464	9893

Figure 4.2a shows the distribution of the derived income profile for all users in Chicago. The indicator group (green) and the reference group (red) correspond to the 10 percent ‘richest’ and 10 percent ‘poorest’ users, respectively. Figure 4.2b confirms my initial assumption that a higher number of high-income-profile visitors is associated with higher rents, as indicated by Spearman’s rank correlation coefficient (r_s) ($r_s = 0.99$; $p < 10^{-18}$; $N = 22$), and that a higher number of low-income-profile visitors is an indicator for lower rents ($r_s = -0.97$; $p < 10^{-12}$; $N = 20$). Interestingly, towards a higher number of unique visitors the two curves diverge the most, indicating large differences between neighborhoods with high numbers of high-income-profile visitors and neighborhoods with high numbers of

low-income-profile visitors (see also Figure 4.3b and d). This strengthens my initial assumption that defining an indicator group based on the income profile of visited location is an appropriate measure when looking at future changes in housing prices.

FIGURE 4.2: Selecting an indicator group. **a**, Income-profile distribution for users in Chicago. I assign an income profile to each user based on the census median income of the visited neighborhoods. The 10 percent ‘richest’ users represent the indicator group (green) while the 10 percent ‘poorest’ users function as a reference group (red). **b**, For each neighborhood, I compare the number of unique indicator visitors (green) and the number of unique reference visitors (red) in 2012/2013 with the neighborhood’s median rent prices in 2012. Neighborhoods that bring together more indicator visitors show higher rent prices while more reference visitors indicate lower rent prices.

4.5.3 Predicting neighborhood change

Rising housing prices. Figure 4.3 demonstrates the specific suitability of the developed indicator measure for the prediction of rising rents within the different neighborhoods of Chicago. Indeed, as is depicted in Figure 4.3a, the number of high-income-profile visitors between July 2012 and June 2013 is strongly correlated with an increase of the rent values over the *following* years (2012 - 2015) ($r_s = 0.70$; $p < 10^{-3}$; $N = 22$). Conversely, the correlation becomes negative for the reference group ($r_s = -0.59$; $p < 10^{-2}$; $N = 20$). This confirms my main assumption behind the selection of the indicator group: neighborhoods that attract a large number of visitors with a high-income profile also tend to experience rising rents in the subsequent years. Figures 4.3b-e show the spatial aspect of the revealed relations as choropleth maps. Comparing Figure 4.3b with Figure 4.3d reveals a strong difference between the activity spaces of the indicator group and the reference

group. The neighborhoods preferred by the indicator group are located towards the north of the city center, while those preferred by the reference group tend to be located in the western and southern parts of the city center. This difference is not surprising per se. It simply reflects the spatial segregation along socioeconomic lines common in many cities. However, it does again underline that the indicator group is in some ways distinct from the population as a whole. Moreover, comparing Figure 4.3b with Figure 4.3c, visually confirms that neighborhoods with a large number of indicator visitors also experience a high rent increase in the following years. Comparing Figure 4.3d with Figure 4.3e indicates the negative correlation found for the reference group.

The robustness of the indicator group as a predictor of rising housing prices is confirmed for all cities analyzed in this thesis, see Figure 4.4.

In comparison, the average rent for 2012 is far less (and often not significantly) correlated with changes in rents over the following years. I found a negative correlation for Chicago ($r_s = -0.57$) while the correlation values for New York City, Los Angeles, Boston and Portland are not significant ($p > 0.05$). The average rents alone are thus not suitable as a robust predictor of rising housing prices.

Transforming neighborhoods. Neighborhoods that experience large increases in rental prices might already have a relatively high median rent. In simpler terms: an already expensive neighborhood becomes even more expensive. Such rent increases might not always go hand-in-hand with larger changes in the characteristics of a neighborhood. On the other hand, neighborhoods that have relatively affordable rents but see similarly large increases in rent prices in subsequent years, could be more significantly affected, which is why this often holds the concern of urban planners and policy makers. However, finding and identifying these neighborhoods at an early stage is a challenging problem. This is partly caused by the delay in measuring these changes in conventional data sources. While social sensing might provide a solution, another challenge remains: real estate markets themselves might be slow to respond to neighborhood changes and rents could thus increase with a delay. Here I use this delay to my advantage and identify neighborhoods that attract the indicator group with the same frequency as other neighborhoods but have disproportionately low rents. More precisely, I define the Visitor-Rent-Index as $VRI = x/\tilde{x} - 1$, where x is the rent of a neighborhood and $\tilde{x}$ is the median rent

FIGURE 4.3: Comparing the predictions of the indicator group and the reference group. **a**, For each neighborhood bin in Chicago, I compare the number of unique indicator visitors (green) and the number of reference visitors (red) in 2012/2013 with the 2012-2015 change of the census median rent. Neighborhoods with more indicator visitors tend to experience a higher rent increase. **b-c**, Mapping the rent price predictions (a) for the indicator group, and **d-e**, for the reference group.

of all other neighborhoods in the same bin. Negative VRI values thus indicate a disproportionately low rent, and vice versa.

FIGURE 4.4: Robustness of the indicator group across multiple cities. Comparing the rent price predictions of the indicator group and the reference group for New York City (indicator group: $r_s = 0.70$; $p < 10^{-4}$; $N = 25$, reference group: $r_s = 0.46$; $p = 0.03$; $N = 22$), Los Angeles (indicator group: $r_s = 0.73$; $p < 10^{-4}$; $N = 23$), Boston (indicator group: $r_s = 0.57$; $p < 10^{-2}$; $N = 20$) and Portland (indicator group: $r_s = 0.67$; $p < 10^{-2}$; $N = 14$). The correlation values for the reference group in Los Angeles, Boston and Portland are not significant ($p > 0.05$).

The result is shown in Figure 4.5a. It highlights three neighborhoods in different transformation stages: Pilsen, which in 2012 still had relatively low median rent and could be labeled as in an early stage of transformation ($VRI \ll 0$); Logan Square, which in 2012 had moderately higher rents and can be seen as in a medium stage of transformation ($VRI < 0$); and finally Lake View, which is in a later stage of transformation with high rents ($VRI \geq 0$). At this point, I selected these well-known neighborhoods as illustration. Figure 4.5b maps the Visitor-Rent-Index with $VRI < 0$ indicating neighborhoods in 2012 where future transformation is likely. Figure 4.5c supports my hypothesis that indeed the neighborhoods with a low VRI show significantly (95% confidence interval) higher rent increase in the following years compared to neighborhoods with a high VRI. However, more

research needs to be done for a systematic analysis of VRI and its relation to neighborhood transformation.

FIGURE 4.5: Transforming neighborhoods. **a**, For each neighborhood in Chicago, I compare the number of unique indicator visitors in 2012/2013 with the census median rent in 2012; the neighborhood bins (green) generate a baseline (corresponding to Figure 4.2b). Neighborhoods below (above) the baseline show a disproportionately high (low) rent compare to other neighborhoods in the same bin. With Pilsen, Logan Square, and Lake View, I show three neighborhoods in different transformation stages. **b**, Mapping the level of disproportion with the Visitor-Rent-Index (VRI) (on the rise, $VRI < 0$). Here, I removed neighborhoods with less than 10 indicator visitors from the map. **c**, I compare the rent increase (2012-2015) in neighborhoods with a low VRI (first quartile) to neighborhoods with a high VRI (last quartile). Neighborhoods with a low VRI show significantly higher rent increase (notches indicate 95% confidence interval around the medians).

It is here that the advantages of the combination of indicator groups with social sensing start to become clear. By utilizing the finer spatio-temporal granularity of data available from social media data platforms, I can potentially analyze and predict dynamic urban processes that are difficult to capture with conventional census or survey instruments, or can only be captured with significant delay. This potential has long been highlighted in the academic literature but so far it has been difficult to demonstrate the potential in more applied, practical work as such datasets have large issues with bias and representativeness. In addition, once we go down to those finer spatial and temporal scales, big data becomes ‘small’ very fast. The indicator group approach may overcome such limitations because the overall size of an indicator group is not so relevant, as long as it is not too ‘rare’ and is indeed a good indicator for the phenomenon of interest.

Beyond income. Up until now, income was the only user attribute included. However, the great potential of the indicator approach lies in its flexibility. First, I can include multiple user attributes to fine-tune the indicator group identification. For example, I could specify it to ‘high-income low-age single-status’. Second, I can include multiple indicator groups to improve the prediction power (corresponding to species assemblages in ecology). In this sense, selecting the appropriate indicator groups is not unlike other optimization problems. In future work, I can make use of machine learning techniques to further enhance the predictive power and accuracy of the method. This will help for the inference of more detailed socio-economic characteristics of urban dwellers [132] as well as for the systematic analysis of different indicator groups and their combinations.

In this thesis, I illustrate this potential by introducing a second user attribute. By way of example, I adopt the median age of a neighborhood. Importantly, age as a single attribute might not be a good predictor for rising housing prices (consider neighborhoods with a high density of low-age blue collar families). It is only in its combination with income that it increases prediction power where ‘high-income low-age’ people can indicate ‘hipness’ of a neighborhood. I evaluate the user’s age profile in a similar way to its income profile. This time, whenever a user visited a neighborhood I assign the weighted median age instead of the median household income. I then upgrade the indicator group by only selecting the 10 percent ‘youngest’ of the high-income-profile users. In the following I refer to them as high-income-low-age-profile users. Consequently, the upgraded indicator measure counts the number of unique high-income-low-age-profile users a neighborhood attracted between July 2012 and June 2013.

Figure 4.6a shows the distribution of the income-age profile for Chicago users with the upgraded indicator group (green) now representing the 10 percent ‘youngest’ of the ‘richest’ users. Figure 4.6b shows that the number of high-income low-age visitors is a better indicator for high rents than high-income visitors in general, which confirms my initial assumption that age as a second user attribute can significantly improve the prediction power of the indicator group.

FIGURE 4.6: Upgrading the indicator group. **a**, Income-age-profile distribution for users in Chicago. I assign an income-age profile to each user based on the census median income and median age of the visited neighborhoods (darker colors indicate a higher frequency). The 10 percent ‘youngest’ of the ‘richest’ users represent the upgraded indicator group (green) while all ‘richest’ users represent the initial indicator group. **b**, For each neighborhood bin in Chicago, I compare the number of upgraded indicator visitors (green) and the number of initial indicator visitors (red) in 2012/2013 with the 2012-2015 change of the census median rent. Using visitors with a high-income-low-age profile as indicator group yields a stronger correlation than using all high-income-profile visitors (upgraded: $r_s = 0.78$; $p < 10^{-3}$; $N = 15$, initial: $r_s = 0.70$; $p < 10^{-3}$; $N = 22$).

Part III

Conclusion and future directions

Chapter 5

Conclusion and future directions

Human interactions are the fundamental drivers that establish cities as the hubs for innovation and economic growth. Here the way people interact in the city is directly linked to the spatial organization of urban spaces and transportation infrastructures. Importantly, the rapid growth and changing dynamics of cities requires a continuous adaptation of the system. This can lead to a complex feedback loop when the present interactivity informs the planning of new spaces and infrastructures. To that end, this thesis introduced new methods for the quantitative description of interactivity in cities and its relation to the spatial organization of urban spaces and transportation infrastructure.

From the vast palette of mobility models, the thesis focused on spatial interaction models which aim to predict aggregated mobility flows (number of trips) based on the spatial distribution and socio-economic characteristics of locations. Such models take an important stage in times where data privacy often won't allow to develop models that are dependent on individual mobility trajectories. Models that are able to describe the complex mobility patterns of today's cities solely based on aggregated mobility data are therefore of great value for urban planning.

To that end, the first part of the thesis introduced a new machine learning-based method, named Horseshoe-NNGP, for the prediction of urban mobility flows. The proposed approach offers not only the flexibility and accuracy of deep neural networks but also explainable predictions with respect to the relevance of the inputs and the confidence of its predictions. Specifically, the Horseshoe-NNGP is based on

the probabilistic formulation of an infinitely wide deep neural network as a Gaussian process and the integration of automatic variable selection using the Horseshoe prior. The effectiveness of the proposed approach is verified with synthetic and real data sets and has shown superior performance compared to traditional non-linear spatial interaction models in terms of prediction performance, uncertainty estimation, and variable selection. The developed method has shown to be particularly relevant for the prediction of intra-city Origin-Destination matrices and the resulting insights can help urban experts to make decisions and communicate to the public, for example, to predict how a planned urban development in a given neighborhood will affect mobility flows between neighborhoods in the city. Importantly, predictions come with insights into how certain the model is about its predictions as well as insights into which urban measures are relevant for the prediction. These characteristics of the proposed approach are an important advance over existing models since model explainability is very important to ensure acceptability of the model for urban experts as well as the public.

However, with respect to the developed model more research is needed to benchmark the prediction performance and compare to other non-linear models that were recently introduced to spatial interaction modeling such as decision trees and simple neural network models [80]. Furthermore, it could be informative to compare the uncertainty estimation of the chosen *exact* Bayesian inference approach with existing *approximate* Bayesian inference approaches such as Variational Inference [21]. Such comparison could give insights into the effectiveness of the chosen *exact* inference approach that offers a principled way of handling model uncertainty but comes at the cost of stronger modelling constraints. Furthermore, it could be useful to benchmark the chosen variable selection method with simpler methods such as filtering. With respect to the input variables more research is needed for the systematic analysis of variables that could potentially lead to better prediction results such as travel time between origin and destination, number of social media posts at a location, etc. Additionally, an extension of this study across different data sets (telcom, surveys, census, etc.) across different countries with different cultural context would further help to proof the robustness of my approach. With respect to an application in practice more research is also needed to better understand the potential of the developed method for an actual application in urban planning. The biggest advantages of the Horseshoe-NNGP are certainly the combination of accurate predictions, uncertainty estimation, and variable selection. However, the

biggest hurdle for an application in practice might be the complexity of the model. It might be worthwhile to research if the lost performance of a simplified model is worth its higher acceptance in practice.

While I demonstrated the relevance of the Horseshoe-NNGP for the specific problem of predicting urban mobility flows, it could be applied to a broader spectrum of processes where accurate yet explainable predictions are important. Possible examples include the application in the health sector, for example, in intensive care prediction [60] or in the real estate sector, for example in home valuation. It is worth noting that the developed methods need to be made further accessible to practitioners so as to foster the direct usability for policy makers and planners (e.g. through ‘easy-to-use’ software tools). This is however beyond the scope of this thesis. Another important avenue for future research is a closer link to scientific mobility models which are build on a theoretical understanding of the domain. To that end, I see the added value of the developed machine learning models in providing additional information about the explanatory variables which can guide the design process of the scientific models.

The second part of the thesis introduced the concept of indicator groups, adapted from ecology, to the analysis of socio-economic processes in cities. The underlying assumption of the proposed approach is that a change in human mobility patterns often precedes physical neighborhood change by several years. Taking gentrification as an example, the influx of artists into a deprived neighborhood might happen years before the neighborhood becomes hip and attracts the interest of investors. The hypothesis was that by monitoring changing mobility patterns of a carefully selected indicator group one can predict neighborhood change years into the future. Specifically, the developed method uses socio-economic profiles of LBSN users in combination with their activity spaces as predictor of changing urban neighborhoods. It does so by defining an indicator group as a small, specifically defined, subset of the population that is especially sensitive to the neighborhood changes that I am interested in studying. This is akin to the canaries that miners deployed in coal mines: canaries are more sensitive to carbon-monoxide than humans and thus serve a natural early warning system. My approach has shown to be particularly relevant for the early detection of future drastic rent increases in neighborhoods that currently have relatively low rents. This is an area of study that is highly relevant to urban stakeholders, from policymakers to real estate developers.

Traditional approaches can be hindered by a lack of granular data or conversely an overload of data (needle in a haystack). The indicator group approach provides a potential solution. If defined appropriately for the phenomenon of interest, this approach can yield early predictions while simultaneously reducing the amount of data that needs to be collected and analyzed.

However, more research is needed to benchmark the predictive power of my approach. This includes the systematic analysis of unobserved variables that could potentially yield early predictions of rent increases such as quality of schools, crime rate, number of restaurants, etc. An extension of this study across different countries with different cultural context would further help to proof the robustness of my approach. More research is also needed to better understand the underlying mechanisms of increasing rents which would help to distinguish between correlation and causality [158] and shed light on the ‘chicken-egg problem’, i.e. do indicator groups visit neighborhoods because they are increasing in value vs. do neighborhoods increase in value because of more indicator visitors. Here it can be helpful to include e.g. the purpose of visit as a parameter [159] and cluster areas independent of administrative boundaries [160].

An advantage of the indicator group approach is that it is very flexible. Different groups can be identified for different social processes of interest, or they can even be adapted to unique local contexts. While I demonstrated the feasibility of my framework for the specific problem of predicting rising housing prices, it thus can be applied to a broader spectrum of urban processes. Possible examples include crime patterns or increasing social segregation in a city. Nevertheless, further investigation needs to be forthcoming to enable a more systematic identification of suitable indicator groups. The median income and the median age of a neighborhood were two helpful attributes to construct the profile of the indicator group. A natural extension for future work would be to enhance the indicator profile by incorporating additional attributes that should be informed by the existing literature on neighborhood change, e.g. [124, 125], and that could be enhanced with the help of machine learning approaches. Importantly, the set of additional attributes should not be constrained to household types but should also consider neighborhood characteristics such as the amenity landscape within a neighborhood [161]. With the increasing availability of detailed data on how individuals actually make use of urban space, finding a suitable indicator group can then be seen as an optimization

problem. On these premises, the application of machine learning techniques may offer a promising next step towards a novel urban monitoring tool that is based on a comprehensive set of early-warning indicators [162, 163]. Another topic for future work could be the link to the supply side when analyzing housing price movements. In my thesis, I have only focused on the demand side where gentrification and the increased demand for housing in certain neighborhoods was assumed to be the driver for increasing housing prices. However, housing supply elasticities will have a strong effect on housing prices as well [164, 165] and should therefore be modelled as well.

List of Author's Publications

Journal Articles

- **Aike Steentoft**, Ate Poorthuis, Bu-Sung Lee, and Markus Schläpfer, “The canary in the city: indicator groups as predictors of local rent increases,” *EPJ Data Science*.
- **Aike Steentoft**, Bu-Sung Lee, and Markus Schläpfer, “Application of explainable machine learning to spatial interaction modeling” *in preparation*.

Bibliography

- [1] Desislava Hristova, Matthew J Williams, Mirco Musolesi, Pietro Panzarasa, and Cecilia Mascolo. Measuring urban social diversity using interconnected geo-social networks. In *Proceedings of the 25th International Conference on World Wide Web*, pages 21–30. International World Wide Web Conferences Steering Committee, 2016. [xix](#), [4](#), [84](#), [91](#)
- [2] Markus Schläpfer, Luís MA Bettencourt, Sébastien Grauwin, Mathias Raschke, Rob Claxton, Zbigniew Smoreda, Geoffrey B West, and Carlo Ratti. The scaling of human interactions with city size. *Journal of the Royal Society Interface*, 11(98):20130789, 2014. [1](#), [88](#)
- [3] Michael Batty. *The new science of cities*. MIT Press, 2013. [1](#)
- [4] Geoffrey B West. *Scale: the universal laws of growth, innovation, sustainability, and the pace of life in organisms, cities, economies, and companies*. Penguin, 2017. [1](#)
- [5] Kees Christiaanse, Apard Hetey, Vera Schmidt, and Ying Zou. Urban breeding grounds: research and design on Rochor+. *FCL Magazine Special Issue*, 2016. [1](#)
- [6] Edward L Glaeser. Learning in cities. *Journal of Urban Economics*, 46(2): 254–277, 1999. [1](#)
- [7] Hugo Barbosa, Marc Barthelémy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1–74, 2018. [2](#), [54](#), [56](#)
- [8] George Kingsley Zipf. The $p \propto 1/p^2$ hypothesis: on the intercity movement of persons. *American Sociological Review*, 11(6):677–686, 1946. [2](#), [54](#), [56](#)
- [9] Samuel A Stouffer. Intervening opportunities: a theory relating mobility and distance. *American Sociological Review*, 5(6):845–867, 1940. [2](#), [54](#), [58](#)
- [10] Paul Waddell. UrbanSim: Modeling urban development for land use, transportation, and environmental planning. *Journal of the American Planning Association*, 68(3):297–314, 2002. [2](#)

- [11] Paul Salvini and Eric J Miller. Ilute: An operational prototype of a comprehensive microsimulation model of urban systems. *Networks and Spatial Economics*, 5(2):217–234, 2005. [2](#)
- [12] Michael Wegener. Overview of land use transport models. In *Handbook of Transport Geography and Spatial Systems*. Emerald Group Publishing Limited, 2004. [2](#)
- [13] Edgar Malone Hoover and Raymond Vernon. Anatomy of a metropolis. The changing distribution of people and jobs within the New York Metropolitan Region. 1959. [2](#), [87](#)
- [14] Kent P Schwirian. Models of neighborhood change. *Annual Review of Sociology*, 9(1):83–102, 1983. [2](#), [87](#)
- [15] Maxime Lenormand, Aleix Bassolas, and José J Ramasco. Systematic comparison of trip distribution laws and models. *Journal of Transport Geography*, 51:158–169, 2016. [3](#), [54](#), [58](#), [59](#), [60](#)
- [16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016. [3](#), [12](#), [17](#)
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. [3](#)
- [18] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013. [3](#)
- [19] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. [3](#)
- [20] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014. [3](#)
- [21] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011. [3](#), [20](#), [106](#)
- [22] Robert J Hill and Daniel Melser. Hedonic imputation and the price index problem: an application to housing. *Economic Inquiry*, 46(4):593–609, 2008. [3](#), [83](#)
- [23] Alain Bertaud. *Order without design: how markets shape cities*. MIT Press, 2018. [4](#)

- [24] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE, 2018. [5](#)
- [25] Aike A Steentoft, Ate Poorthuis, Bu-Sung Lee, and Markus Schlöpfer. The canary in the city: indicator groups as predictors of local rent increases. *EPJ Data Science*, 7(1):21, 2018. [7](#), [83](#)
- [26] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584, 2017. [11](#)
- [27] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012. [11](#), [26](#)
- [28] Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 1(3), 2016. [12](#)
- [29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. [12](#)
- [30] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013. [12](#), [14](#), [29](#), [31](#)
- [31] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *ICLR*, 2018. [13](#), [23](#), [33](#), [36](#), [38](#), [42](#)
- [32] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003. [15](#), [17](#), [36](#)
- [33] John S Denker and Yann LeCun. Transforming neural-net output levels to probability distributions. In *Advances in neural information processing systems*, pages 853–859, 1991. [19](#)
- [34] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992. [19](#)
- [35] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on computational learning theory*, pages 5–13, 1993. [20](#)
- [36] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015. [20](#), [33](#)

- [37] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015. [20](#)
- [38] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang Bui, and Richard Turner. Black-box α -divergence minimization. 2016. [20](#)
- [39] Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. *arXiv preprint arXiv:1802.02538*, 2018. [20](#)
- [40] Radford M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, Toronto, Ont., Canada, Canada, 1995. AAINN02676. [21](#), [23](#), [28](#), [33](#), [34](#), [35](#), [37](#), [38](#)
- [41] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011. [22](#)
- [42] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012. [22](#)
- [43] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International conference on machine learning*, pages 1683–1691, 2014. [22](#)
- [44] Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate Bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017. [23](#)
- [45] Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*, 1997. [23](#), [28](#)
- [46] Christopher KI Williams. Computing with infinite networks. In *Advances in neural information processing systems*, pages 295–301, 1997. [23](#)
- [47] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018. [23](#)
- [48] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. *arXiv preprint arXiv:1808.05587*, 2018. [24](#)
- [49] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013. [24](#), [44](#), [51](#)

- [50] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003. [24](#)
- [51] Verónica Bolón-Canedo, Noelia Sánchez-Maróño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519, 2013. [24](#), [41](#)
- [52] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. [26](#)
- [53] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015. [26](#)
- [54] Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032, 1988. [27](#), [36](#)
- [55] Juho Piironen, Aki Vehtari, et al. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051, 2017. [27](#), [29](#), [35](#), [39](#)
- [56] David JC MacKay et al. Bayesian nonlinear modeling for the prediction competition. *ASHRAE transactions*, 100(2):1053–1062, 1994. [28](#)
- [57] Carlos M Carvalho, Nicholas G Polson, and James G Scott. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80, 2009. [28](#)
- [58] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*, pages 3288–3298, 2017. [29](#)
- [59] Soumya Ghosh and Finale Doshi-Velez. Model selection in bayesian neural networks via horseshoe priors. *arXiv preprint arXiv:1705.10388*, 2017. [29](#)
- [60] Anna-Lena Popkes, Hiske Overweg, Ari Ercole, Yingzhen Li, José Miguel Hernández-Lobato, Yordan Zaykov, and Cheng Zhang. Interpretable outcome prediction with sparse bayesian neural networks in intensive care. *arXiv preprint arXiv:1905.02599*, 2019. [29](#), [34](#), [107](#)
- [61] Efficient bayesian inference with hamiltonian monte carlo. <https://www.youtube.com/watch?v=pHsuIaPbNbY>. Accessed: 2020-08-01. [29](#)
- [62] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017. [29](#), [30](#)

- [63] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. [31](#)
- [64] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970. [31](#)
- [65] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012. [32](#)
- [66] Arya A Pourzanjani, Richard M Jiang, and Linda R Petzold. Improving the identifiability of neural networks for bayesian inference. In *NIPS Workshop on Bayesian Deep Learning*, volume 4, page 29, 2017. [35](#)
- [67] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010. [35](#), [36](#)
- [68] Juho Piironen and Aki Vehtari. Comparison of bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735, 2017. [35](#)
- [69] Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009. [38](#)
- [70] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the NIPS 2003 feature selection challenge. In *Advances in neural information processing systems*, pages 545–552, 2005. [41](#), [44](#), [49](#)
- [71] Isabelle Guyon. Design of experiments of the NIPS 2003 variable selection benchmark. In *NIPS 2003 workshop on feature extraction and feature selection*, volume 253, 2003. [42](#), [43](#), [44](#)
- [72] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of machine learning research*, 5(Jan):101–141, 2004. [42](#)
- [73] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011. [42](#)
- [74] Madelon sklearn. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html. Accessed: 2020-06-01. [43](#)
- [75] Michael Betancourt and Mark Girolami. Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79(30):2–4, 2015. [46](#)
- [76] Hamiltonian Monte Carlo in Stan. https://mc-stan.org/docs/2_18/reference-manual/hmc-algorithm-parameters.html. Accessed: 2020-06-01. [47](#)

- [77] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011. [47](#)
- [78] Filippo Simini, Marta C González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *Nature*, 484(7392):96, 2012. [54](#), [59](#)
- [79] Nastaran Pourebrahim, Selima Sultana, Amirreza Niakanlahiji, and Jean-Claude Thill. Trip distribution modeling with twitter data. *Computers, Environment and Urban Systems*, 77:101354, 2019. [54](#), [61](#)
- [80] Caleb Robinson and Bistra Dilkina. A machine learning approach to modeling human migration. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–8, 2018. [54](#), [60](#), [106](#)
- [81] Di Chai, Leye Wang, and Qiang Yang. Bike flow prediction with multi-graph convolutional networks. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 397–400, 2018.
- [82] Florian Toqué, Etienne Côme, Mohamed Khalil El Mahrssi, and Latifa Oukhellou. Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pages 1071–1076. IEEE, 2016.
- [83] Frans Tillema, Kasper M Van Zuilekom, and Martin FAM Van Maarseveen. Comparison of neural networks and gravity models in trip distribution. *Computer-Aided Civil and Infrastructure Engineering*, 21(2):104–119, 2006. [54](#)
- [84] Alan Geoffrey Wilson. A family of spatial interaction models, and associated developments. *Environment and Planning A*, 3(1):1–32, 1971. [56](#)
- [85] Xinhai Li, Huidong Tian, Dejian Lai, and Zhibin Zhang. Validation of the gravity model in predicting the global spread of influenza. *International journal of environmental research and public health*, 8(8):3134–3143, 2011. [56](#)
- [86] Ira S Lowry. A model of metropolis. Technical report, RAND CORP SANTA MONICA CALIF, 1964. [57](#)
- [87] Ira S Lowry. A short course in model design. *Journal of the American Institute of Planners*, 31(2):158–166, 1965. [57](#)
- [88] David L Huff. A probabilistic analysis of shopping center trade areas. *Land economics*, 39(1):81–90, 1963. [57](#)
- [89] James E Anderson and Eric Van Wincoop. Gravity with gravitas: A solution to the border puzzle. *American economic review*, 93(1):170–192, 2003. [58](#)

- [90] Ferdinando Monte, Stephen J Redding, and Esteban Rossi-Hansberg. Commuting, migration, and local employment elasticities. *American Economic Review*, 108(12):3855–90, 2018. [58](#)
- [91] Damiaan Persyn and Wouter Torfs. A gravity equation for commuting with an application to estimating regional border effects in belgium. *Journal of Economic Geography*, 16(1):155–175, 2016. [58](#)
- [92] Sumit Agarwal, J Bradford Jensen, and Ferdinando Monte. Consumer mobility and the local structure of consumption industries. Technical report, National Bureau of Economic Research, 2017. [58](#)
- [93] James P LeSage and R Kelley Pace. Spatial econometric modeling of origin-destination flows. *Journal of Regional Science*, 48(5):941–967, 2008. [58](#)
- [94] Lianfa Zhang, Jianquan Cheng, and Cheng Jin. Spatial interaction modeling of od flow data: Comparing geographically weighted negative binomial regression (gwnbr) and ols (gwolsr). *ISPRS International Journal of Geo-Information*, 8(5):220, 2019. [58](#)
- [95] Michael Batty and S Mackie. The calibration of gravity, entropy, and related models of spatial interaction. *Environment and Planning A*, 4(2):205–233, 1972. [58](#)
- [96] Alan Geoffrey Wilson. Urban and regional models in geography and planning. 1974. [58](#), [59](#)
- [97] Sven Erlander and Neil F Stewart. *The gravity model in transportation analysis: theory and extensions*, volume 3. Vsp, 1990. [58](#)
- [98] Peter Davis. Spatial competition in retail markets: movie theaters. *The RAND Journal of Economics*, 37(4):964–982, 2006. [58](#)
- [99] Matthias De Beule, Dirk Van den Poel, and Nico Van de Weghe. An extended huff-model for robustly benchmarking and predicting retail network performance. *Applied Geography*, 46:80–89, 2014. [58](#)
- [100] A Paolo Masucci, Joan Serras, Anders Johansson, and Michael Batty. Gravity versus radiation models: On the importance of scale and heterogeneity in commuting flows. *Physical Review E*, 88(2):022812, 2013. [58](#), [59](#)
- [101] Xiao Liang, Jichang Zhao, Li Dong, and Ke Xu. Unraveling the origin of exponential law in intra-urban human mobility. *Scientific reports*, 3:2983, 2013. [59](#)
- [102] Xiao-Yong Yan, Chen Zhao, Ying Fan, Zengru Di, and Wen-Xu Wang. Universal predictability of mobility patterns in cities. *Journal of The Royal Society Interface*, 11(100):20140834, 2014. [58](#)

- [103] Yingxiang Yang, Carlos Herrera, Nathan Eagle, and Marta C González. Limits of predictability in commuting flows in the absence of data for calibration. *Scientific reports*, 4:5662, 2014. 59
- [104] Mussone Lorenzo and Matteucci Matteo. Od matrices network estimation from link counts by neural networks. *Journal of Transportation Systems Engineering and Information Technology*, 13(4):84–92, 2013. 60
- [105] Gabriel Spadon, Andre CPLF de Carvalho, Jose F Rodrigues-Jr, and Luiz GA Alves. Reconstructing commuters network using machine learning and urban indicators. *Scientific reports*, 9(1):1–13, 2019. 61
- [106] Gevorg Yeghikyan, Felix L Opolka, Mirco Nanni, Bruno Lepri, and Pietro Lio. Learning mobility flows from urban features with spatial interaction models and neural networks. *arXiv preprint arXiv:2004.11924*, 2020. 62
- [107] United States Census Bureau. Longitudinal Employer-Household Dynamics. <https://lehd.ces.census.gov/data/>, . [Online; accessed 08-Apr-2021]. 64
- [108] Chicago Metropolitan Agency for Planning. Chicago Travel Survey. <https://www.cmap.illinois.gov/data/transportation/travel-survey>. [Online; accessed 08-Apr-2021]. 64
- [109] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008. 64
- [110] Census tract shapefiles. <https://www.census.gov/cgi-bin/geo/shapefiles/index.php?year=2019&layergroup=Census+Tracts>. Accessed: 2020-06-01. 65
- [111] Tlc trip record data. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>. Accessed: 2020-06-01. 65
- [112] Lehd employment data. <https://lehd.ces.census.gov/data/lodes>. Accessed: 2020-07-05. 67
- [113] Openstreetmap data. <https://www.openstreetmap.org/>. Accessed: 2020-08-18. 67
- [114] Poi data. <https://overpass-turbo.eu/>. Accessed: 2020-07-06. 67
- [115] Emily A Holt and Scott W Miller. Bioindicators: using organisms to measure environmental impacts. *Nature Education Knowledge*, 3(10):8, 2011. 84, 89
- [116] Loretta Lees, Elvin K Wyly, and Tom Slater. The gentrification reader, 2010. 85
- [117] Alex Pentland and Tracy Heibeck. Honest signals: how they shape our world, 2010. 85

- [118] Peter Roberts and Hugh Sykes. *Urban regeneration: a handbook*. Sage, 1999. [87](#)
- [119] Sharon Zukin. Loft living: culture and capital in urban change, 1989. [87](#)
- [120] Ruth Lazarus Glass. London: aspects of change, 1964.
- [121] Neil Smith. Gentrification and uneven development. *Economic geography*, 58(2):139–155, 1982. [87](#)
- [122] Rowland Atkinson. Measuring gentrification and displacement in greater london. *Urban studies*, 37(1):149–165, 2000. [87](#)
- [123] Daniel J Hammel and Elvin K Wyly. A model for identifying gentrified areas with census data. *Urban Geography*, 17(3):248–268, 1996. [87](#)
- [124] Veronica Guerrieri, Daniel Hartley, and Erik Hurst. Endogenous gentrification and housing price dynamics. *Journal of Public Economics*, 100:45–60, 2013. [87](#), [108](#)
- [125] Victor Couture, Cecile Gaubert, Jessie Handbury, and Erik Hurst. Income growth and the distributional effects of urban spatial sorting. Technical report, National Bureau of Economic Research, 2019. [87](#), [108](#)
- [126] Elizabeth C Delmelle. Differentiating pathways of neighborhood change in 50 us metropolitan areas. *Environment and Planning A*, page 0308518X17722564, 2017. [88](#)
- [127] Elizabeth C Delmelle. Mapping the dna of urban neighborhoods: Clustering longitudinal sequences of neighborhood socioeconomic change. *Annals of the American Association of Geographers*, 106(1):36–56, 2016. [88](#)
- [128] Carlo Ratti, Dennis Frenchman, Riccardo Maria Pulselli, and Sarah Williams. Mobile landscapes: using location data from cell phones for urban analysis. *Environment and Planning B: Planning and Design*, 33(5):727–748, 2006. [88](#)
- [129] Taylor Shelton, Ate Poorthuis, and Matthew Zook. Social media and the city: Rethinking urban socio-spatial inequality using user-generated geographic information. *Landscape and Urban Planning*, 142:198–211, 2015.
- [130] Chen Zhong, Markus Schlöpfer, Stefan Müller Arisona, Michael Batty, Carlo Ratti, and Gerhard Schmitt. Revealing centrality in the spatial structure of cities from human activity patterns. *Urban Studies*, 54(2):437–455, 2017. [88](#)
- [131] Yu Liu, Xi Liu, Song Gao, Li Gong, Chaogui Kang, Ye Zhi, Guanghua Chi, and Li Shi. Social sensing: A new approach to understanding our socio-economic environments. *Annals of the Association of American Geographers*, 105(3):512–530, 2015. [88](#)

- [132] Joshua Blumenstock, Gabriel Cadamuro, and Robert On. Predicting poverty and wealth from mobile phone metadata. *Science*, 350(6264):1073–1076, 2015. [88](#), [93](#), [101](#)
- [133] Francesco Calabrese, Mi Diao, Giusy Di Lorenzo, Joseph Ferreira, and Carlo Ratti. Understanding individual mobility patterns from urban sensing data: A mobile phone trace example. *Transportation research part C: emerging technologies*, 26:301–313, 2013. [88](#)
- [134] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013. [88](#), [93](#)
- [135] Daniel Preoțiuc-Pietro, Vasileios Lamos, and Nikolaos Aletras. An analysis of the user occupational class through twitter content. The Association for Computational Linguistics, 2015. [88](#)
- [136] Bartosz Hawelka, Izabela Sitko, Euro Beinart, Stanislav Sobolevsky, Pavlos Kazakopoulos, and Carlo Ratti. Geo-located twitter as proxy for global mobility patterns. *Cartography and Geographic Information Science*, 41(3):260–271, 2014. [88](#)
- [137] Raja Jurdak, Kun Zhao, Jiajun Liu, Maurice AbouJaoude, Mark Cameron, and David Newth. Understanding human mobility from twitter. *PloS one*, 10(7):e0131469, 2015. [88](#)
- [138] Xin Lu, Erik Wetter, Nita Bharti, Andrew J Tatem, and Linus Bengtsson. Approaching the limit of predictability in human mobility. *Scientific reports*, 3:srep02923, 2013. [88](#)
- [139] Anastasios Noulas, Salvatore Scellato, Renaud Lambiotte, Massimiliano Pontil, and Cecilia Mascolo. A tale of many cities: universal patterns in human urban mobility. *PloS one*, 7(5):e37027, 2012. [88](#)
- [140] Sibren Isaacman, Richard Becker, Ramón Cáceres, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 239–252. Acm, 2012.
- [141] Feixiong Luo, Guofeng Cao, Kevin Mulligan, and Xiang Li. Explore spatiotemporal and demographic characteristics of human mobility via twitter: A case study of chicago. *Applied Geography*, 70:11–25, 2016. [88](#)
- [142] Michele Tizzoni, Paolo Bajardi, Adeline Decuyper, Guillaume Kon Kam King, Christian M Schneider, Vincent Blondel, Zbigniew Smoreda, Marta C González, and Vittoria Colizza. On the use of human mobility proxies for modeling epidemics. *PLoS computational biology*, 10(7):e1003716, 2014. [88](#)

- [143] Taha H Rashidi, Alireza Abbasi, Mojtaba Maghrebi, Samiul Hasan, and Travis S Waller. Exploring the capacity of social media data for modelling travel behaviour: Opportunities and challenges. *Transportation Research Part C: Emerging Technologies*, 75:197–211, 2017. [88](#)
- [144] Aleksandr Farseev, Liqiang Nie, Mohammad Akbari, and Tat-Seng Chua. Harvesting multiple sources for user profile learning: a big data study. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 235–242. ACM, 2015. [88](#)
- [145] Samiul Hasan and Satish V Ukkusuri. Location contexts of user check-ins to model urban geo life-style patterns. *PloS one*, 10(5):e0124819, 2015. [88](#)
- [146] Ahmed AH Siddig, Aaron M Ellison, Alison Ochs, Claudia Villar-Leeman, and Matthew K Lau. How do ecologists select and use indicator species to monitor ecological change? insights from 14 years of publication in ecological indicators. *Ecological Indicators*, 60:223–230, 2016. [89](#)
- [147] Yury Kryvasheyev, Haohui Chen, Esteban Moro, Pascal Van Hentenryck, and Manuel Cebrian. Performance of social network sensors during hurricane sandy. *PLoS one*, 10(2):e0117288, 2015. [89](#)
- [148] Alexander D Singleton and Seth E Spielman. The past, present, and future of geodemographic research in the united states and united kingdom. *The Professional Geographer*, 66(4):558–567, 2014. [89](#)
- [149] Mark Birkin. Customer targeting, geodemographics and lifestyle approaches. *GIS for business and service planning*, pages 104–149, 1995. [89](#)
- [150] Downloads.Esri.Com. Tapestry segmentation: Methodology, June 2016. URL http://downloads.esri.com/esri_content_doc/dbl/us/J9941_Tapestry_Segmentation_Methodology_2016.pdf. [89](#)
- [151] Jakob Petersen, Maurizio Gibin, Paul Longley, Pablo Mateos, Philip Atkinson, and David Ashby. Geodemographics as a tool for targeting neighbourhoods in public health campaigns. *Journal of Geographical Systems*, 13(2): 173–192, 2011. [89](#)
- [152] Ate Poorthuis and Matthew Zook. Making big data small: Strategies to expand urban and geographical research using social media. *Journal of Urban Technology*, pages 1–21, 2017. [91](#), [92](#), [93](#)
- [153] United States Census Bureau. ACS Median Houshold Income. <https://data.census.gov/cedsci/advanced?t=Income%20and%20Poverty&g=0500000US36005.140000&y=2012>, . [Online; accessed 16-Apr-2021]. [92](#)
- [154] United States Census Bureau. ACS Median Rent. <https://data.census.gov/cedsci/advanced?t=Housing&g=0500000US36005.140000&y=2012>, . [Online; accessed 16-Apr-2021]. [92](#)

- [155] Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H Chi. Tweets from justin bieber’s heart: the dynamics of the location field in user profiles. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 237–246. ACM, 2011. [93](#)
- [156] Karl E Case and Robert J Shiller. Forecasting prices and excess returns in the housing market. *Real Estate Economics*, 18(3):253–273, 1990. [93](#)
- [157] Edward L. Glaeser, Joseph Gyourko, and Raven Saks. Why have housing prices gone up? Working Paper 11129, National Bureau of Economic Research, February 2005. URL <http://www.nber.org/papers/w11129>. [94](#)
- [158] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The parable of google flu: traps in big data analysis. *Science*, 343(6176):1203–1205, 2014. [108](#)
- [159] Lorenzo Gabrielli, Salvatore Rinzivillo, Francesco Ronzano, and Daniel Villatoro. From tweets to semantic trajectories: mining anomalous urban mobility patterns. In *Citizen in sensor networks*, pages 26–35. Springer, 2014. [108](#)
- [160] Vanessa Frias-Martinez and Enrique Frias-Martinez. Spectral clustering for sensing urban land use using twitter activity. *Engineering Applications of Artificial Intelligence*, 35:237–245, 2014. [108](#)
- [161] Kristian Behrens, Brahim Boualam, Julien Martin, and Florian Mayneris. Gentrification and pioneer businesses. 2018. [108](#)
- [162] Marten Scheffer, Jordi Bascompte, William A Brock, Victor Brovkin, Stephen R Carpenter, Vasilis Dakos, Hermann Held, Egbert H Van Nes, Max Rietkerk, and George Sugihara. Early-warning signals for critical transitions. *Nature*, 461(7260):53–59, 2009. [109](#)
- [163] Stephen R Carpenter, Jonathan J Cole, Michael L Pace, Ryan Batt, WA Brock, Timmothy Cline, Jim Coloso, James R Hodgson, Jim F Kitchell, David A Seekell, et al. Early warnings of regime shifts: a whole-ecosystem experiment. *Science*, 332(6033):1079–1082, 2011. [109](#)
- [164] Albert Saiz. The geographic determinants of housing supply. *The Quarterly Journal of Economics*, 125(3):1253–1296, 2010. [109](#)
- [165] Joseph Gyourko, Albert Saiz, and Anita Summers. A new measure of the local regulatory environment for housing markets: The wharton residential land use regulatory index. *Urban Studies*, 45(3):693–729, 2008. [109](#)