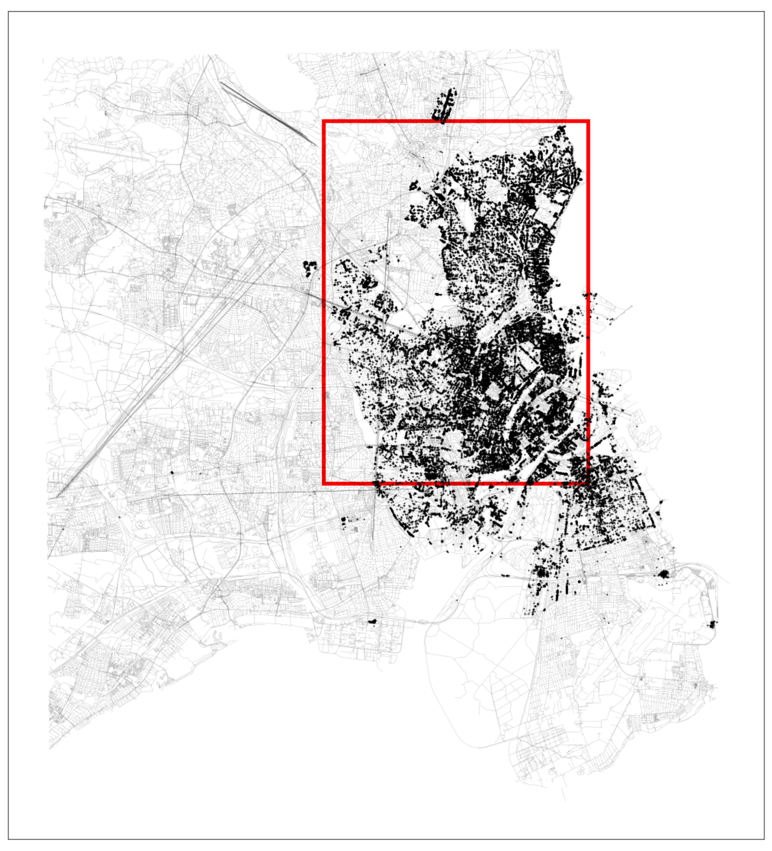
Density estimation of Green Mobility data using Normalizing Flows

Kristian Frellesen Nielsen (s164008) and Theis Ferré Hjortkjær (s163700)
Supervisors: Filipe Rodrigues[1] and Daniele Gammelli[1]
1 Machine Learning for Smart Mobility, DTU Management – Technical University of Denmark

Motivation

Green Mobility (GM) is a leading car sharing company in Denmark. Everyday, hundreds of their cars are picked up to transport people from A to B. Gaining insight into how their cars are used is of course of great interest as this enables GM to make decisions with a greater impact.

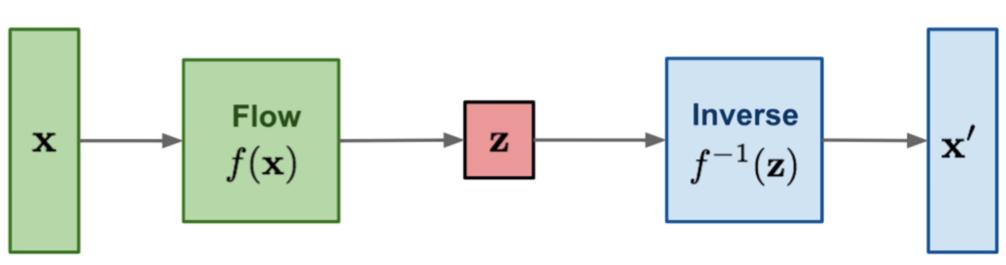
In this project, we will investigate how Normalizing Flows[1] can be used to model the distribution of GM cars being picked up in the Greater Copenhagen area. By using Normalizing Flows we wish to obtain a complex distribution, that with simpler methods would not be possible. To evaluate our models, we will compare it to our baseline which is a simple Gaussian Mixture Model with 10 components.



Green mobility positional data over Greater Copenhagen Area. Red rectangle indicates area of which the distributions are plotted over.

Method

Normalizing flows (NF) is a family of generative models within unsupervised learning methods of machine learning. NFs produces tractable distributions where sampling and density estimation are performed in an efficient way[2]. NFs are constructed by sequentially applying invertible transformations on a prior distribution from where it learns a complex data distribution.



Normalizing Flow architecture [3]

RealNVP

The flow defined in the RealNVP paper[4] is computed with two equations on an input x of dimension D. The equations together make the Affine Coupling layer

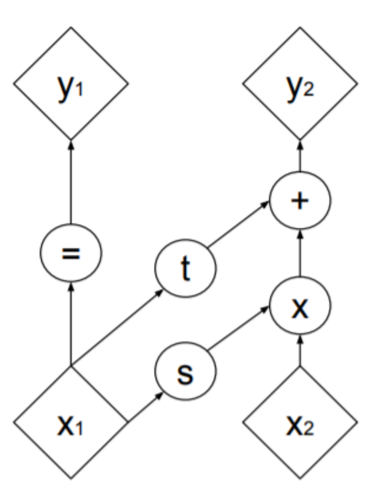
$$y_{1:d} = x_{1:d}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp\big(s(x_{1:d})\big) + t(x_{1:d})$$

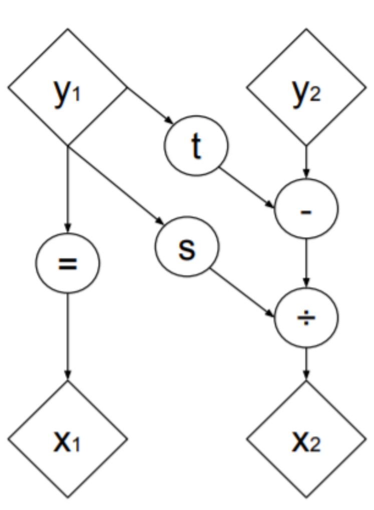
Here s and t are arbitrary complex functions modelled using deep neural networks. The forward and inverse flow are depicted on the figures to the right.
The Jacobian of y w.r.t. to x transposed (x^T) is

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & diag\,(\exp[s(x_{1:d})]) \end{bmatrix}$$

and is a triangular matrix leading to an efficient computation of the determinant, given by the sum of the diagonal entries.



Forward flow



Inverse flow

Model specifications

Prediction task: Find a distribution $p_X(x)$ for $x \in X$ by minimizing the negative log-likelihood (NLL)

$$-\log\big(p_X(x)\big) = -\log(p_Z(f(x)) - \log\left(\left|\det\left(\frac{\partial f(x)}{\partial x^T}\right)\right|\right)$$

where the partial derivative is the Jacobian.

Input representation: Dataset with two features. The two features are the longitude and latitude of the start positions of the vehicles. Two of the NF models will be conditioned on additional features.

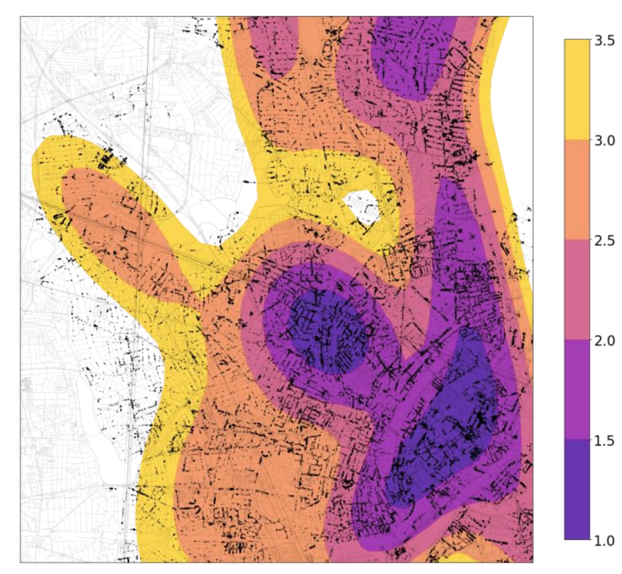
Models: All normalizing flow models are compared against a Gaussian Mixture Model (GMM) with 10 components as baseline.

Model	# of layers	# of nodes	# of features	Batch Norm?
Vanilla NF	6	256	2	No
Time conditioned NF	10	256	4	Yes
Drop-off conditioned NF	6	256	4	No

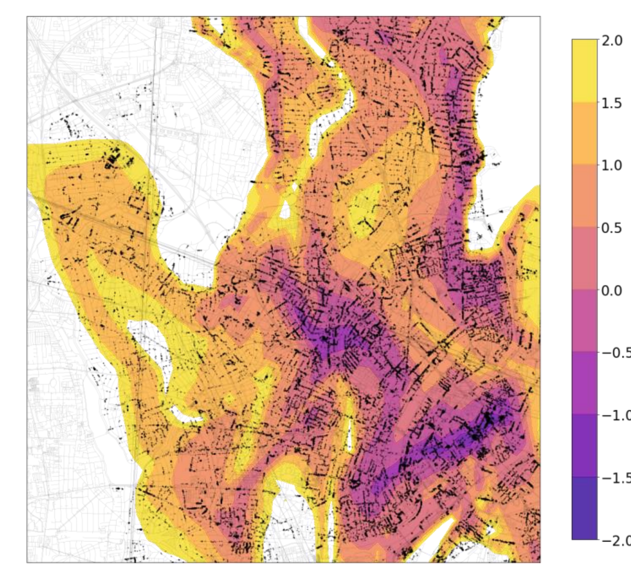
Results

Distributions of the four models are shown by NLL contours of the zoomed-in area outlined in the motivation. The conditioned models each depict 3 plots, to illustrate how the distribution is changed, given a certain condition.

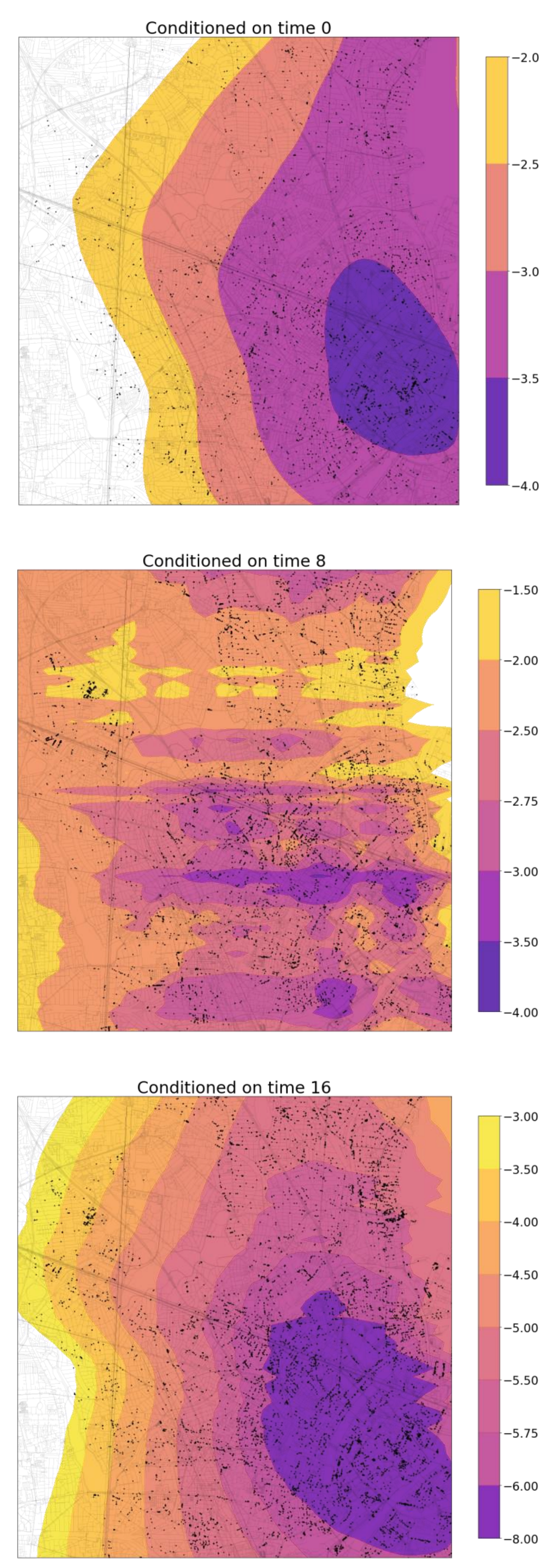
Baseline (GMM)



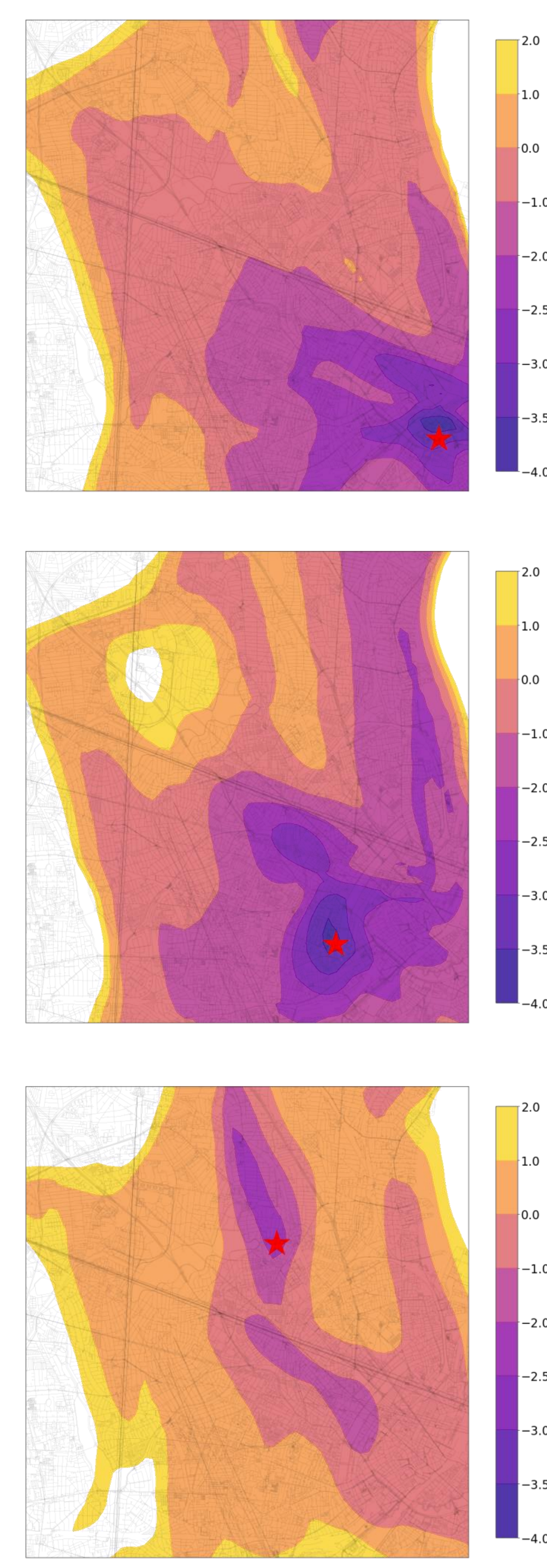
Vanilla NF



Time conditioned NF



Drop-off conditioned NF



Model	Summed NLL ($\times 10^5$)
GMM (10 components)	2.58
Vanilla NF	1.38
Time conditioned NF	−1.94
Drop-off conditioned NF	−5.85

Conclusion

All three normalizing flow models using RealNVP each learned a more complex distribution than the GMM baseline model, also confirmed by the summed NLL. The drop-off conditioned NF model performed the best compared to the other models in terms of the summed NLL value. This indicates that drop-off location provides more information than knowing at what time the car was picked up.

For future work, it would be interesting to see how different Flow models would perform and how adding additional features could change the topology of the distribution pickup locations conditioned on the features.

References

1. Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.
2. Kobyzev, I., Prince, S., & Brubaker, M. A. (2019). Normalizing flows: Introduction and ideas. *arXiv preprint arXiv:1908.09257*.
3. Flow-based Deep Generative Models (accessed 28-11-20)
4. Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.