Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

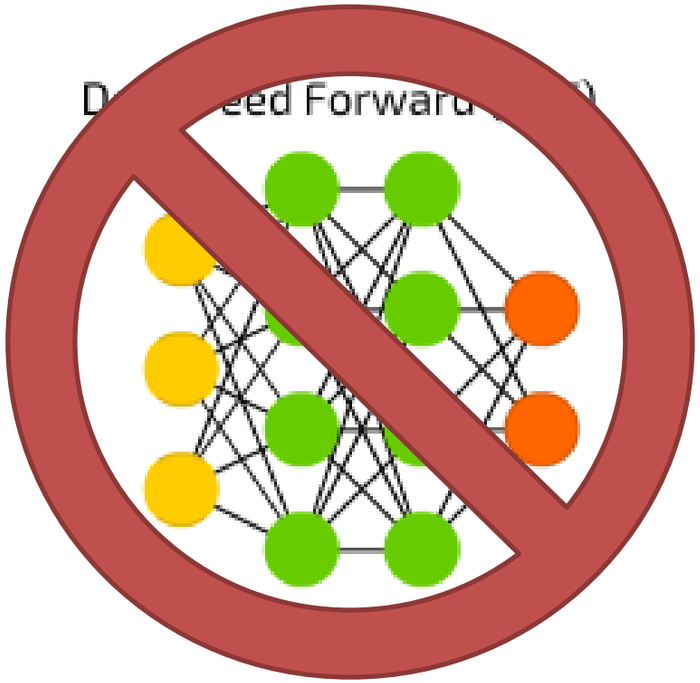
Sistemas Urbanos Inteligentes

Redes Neuronales y Datos Tabulados

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación

Ahora que conocemos los ensambles, ¿dónde están las debilidades de las redes en datos tabulados?

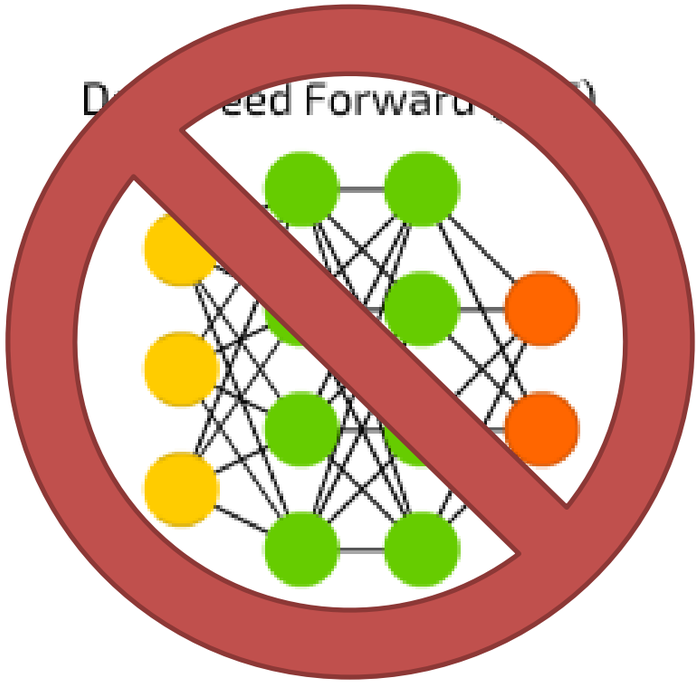
1. Discordancia en tipos de dato
2. Distribución y correlación en los datos



Ahora que conocemos los ensambles, ¿dónde están las debilidades de las redes en datos tabulados?

1. Discordancia en tipos de dato

   - Datos tabulados mezclan generalmente variables categóricas y numéricas.
   - Fuentes son heterogéneas, por lo que pueden tener distintas unidades y escalas de medición.
   - Variables son *sparse*, es decir, toman pocos posibles valores, incluso cuando son numéricas. Peor aún, puede haber datos faltantes.



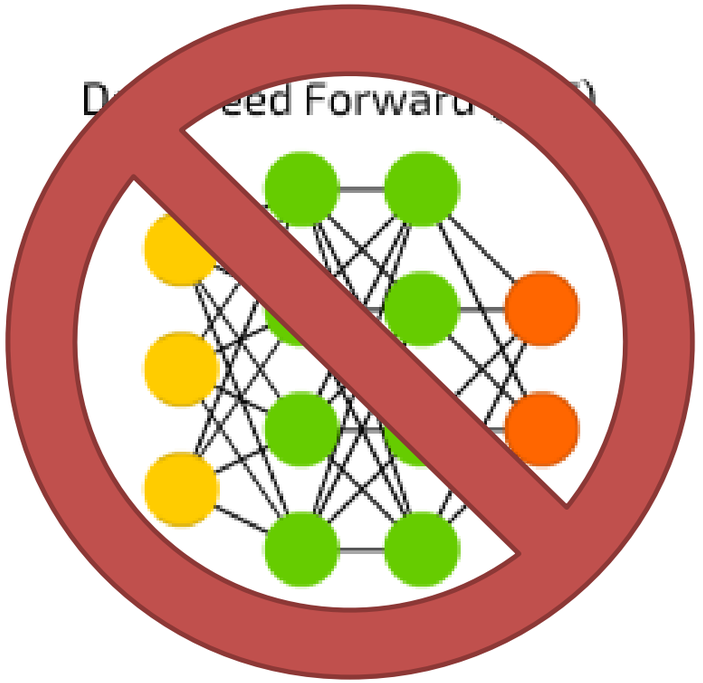
Solución tradicional sería a través de modelación

Ahora que conocemos los ensambles, ¿dónde están las debilidades de las redes en datos tabulados?

2. Distribución y correlación en los datos

- Columnas en datos tabulados presentan generalmente una alta correlación, por lo que basta un subconjunto pequeño de variables para poder hacer la predicción.

- Existe habitualmente un alto desbalance en las clases a predecir.



Solución tradicional sería a través la ingeniería de datos/características y del muestreo

Centrémonos inicialmente en el punto 1. Discordancia en tipos de dato

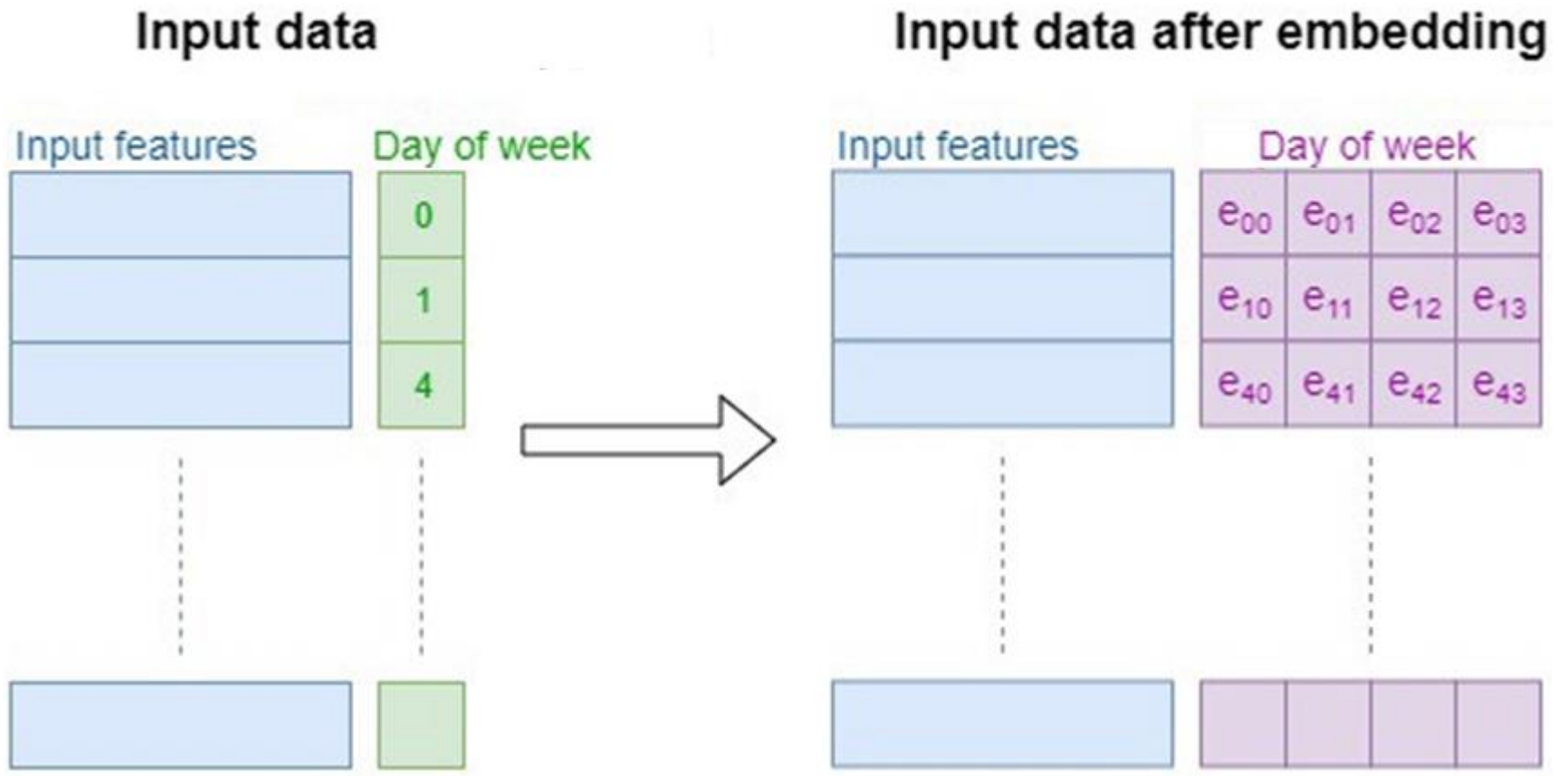
- Si bien la modelación es una herramienta fundamental, en este curso siempre intentaremos tomar inicialmente el enfoque del aprendizaje.
- Al analizar los problemas, notamos que “todo se reduce” a la incapacidad de las redes de trabajar con datos que no sean numéricos (continuos), como los categóricos.
- En vez de adaptar las redes para que funcionen con datos categóricos, buscaremos aprender transformaciones que muevan los datos categóricos a un espacio numérico.
- Esto tiene la ventaja que es posible capturar información semántica en este nuevo espacio (si es aprendida de forma *end-to-end*).
- Para hacer esto, utilizaremos el concepto de *embedding*.

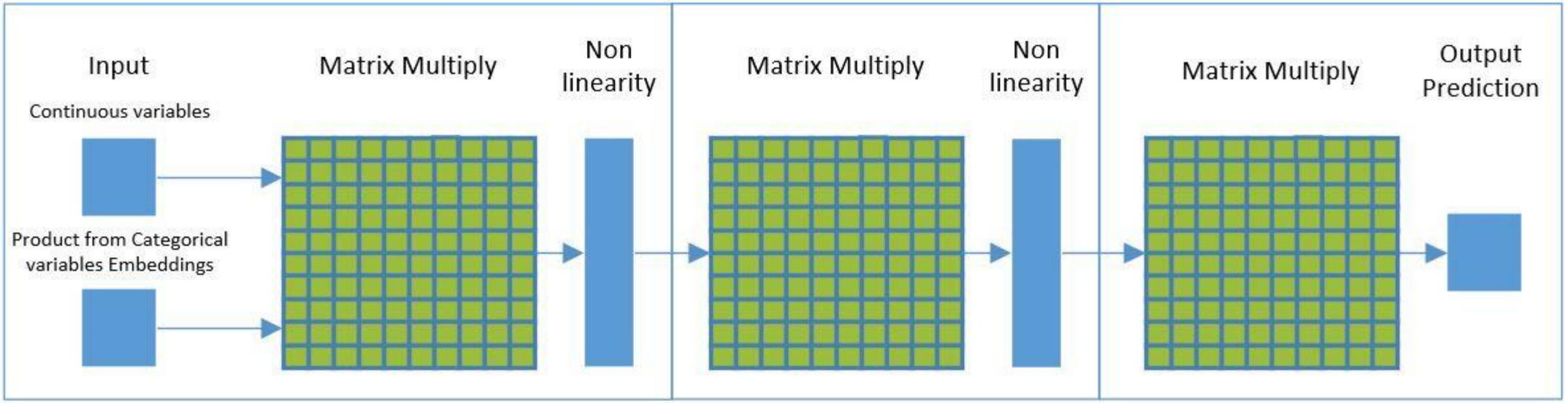
¿Qué es un *embedding*?

- En esencia, un *embedding* es un esquema para transformar datos que viven en espacios no numéricos, a un espacio vectorial continuo, generalmente de menor dimensionalidad (de ahí el nombre).

- Un *embedding* puedes puede ser diseñado o aprendido para una tarea.

- En ML, no solo se usan para transformar datos categóricos, sino también para grafos, texto y otros.


Input data
Input features
Day of week
0
1
4
Input data after embedding
Input features
Day of week
e00 e01 e02 e03
e10 e11 e12 e13
e40 e41 e42 e43


Input
Continuous variables
Product from Categorical variables Embeddings
Matrix Multiply
Non linearity
Matrix Multiply
Non linearity
Matrix Multiply
Output Prediction

¿Como integramos este concepto de *embedding* en una red neuronal?

- En realidad, las redes neuronales lo que hacen es aprender *embeddings*: transformaciones de datos de un espacio a otro.
- En algunos casos, como el *autoencoder*, esa transformación busca reducir la dimensionalidad, pero no es algo obligatorio.
- Hasta ahora, solo hemos hecho transformaciones entre espacios vectoriales continuos.
- ¿Qué cambios debemos hacer entonces en las redes para aprender transformaciones de espacios categóricos a continuos?

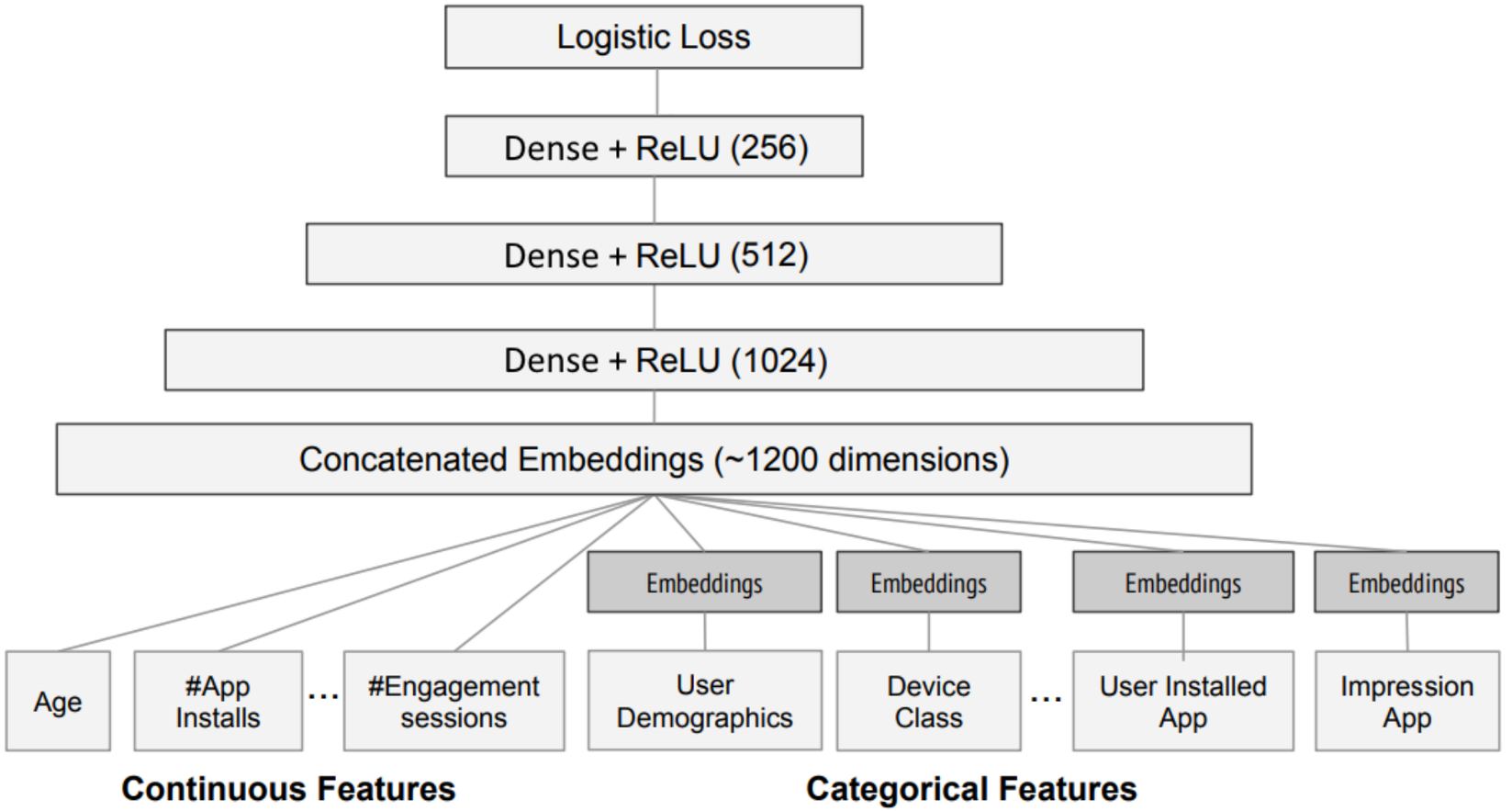
Si la culpa es de los datos, bueno, que entonces cambien los datos…

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Al utilizar una codificación *one-hot*, es posible utilizar productos matriciales para generar un vector asociado a cada código (esencialmente, una indexación)

Finalmente, al tener productos matriciales involucrados, es posible aprender los coeficientes de las matrices, al empaquetarlo todo dentro de una red neuronal



Veamos ahora un caso de estudio

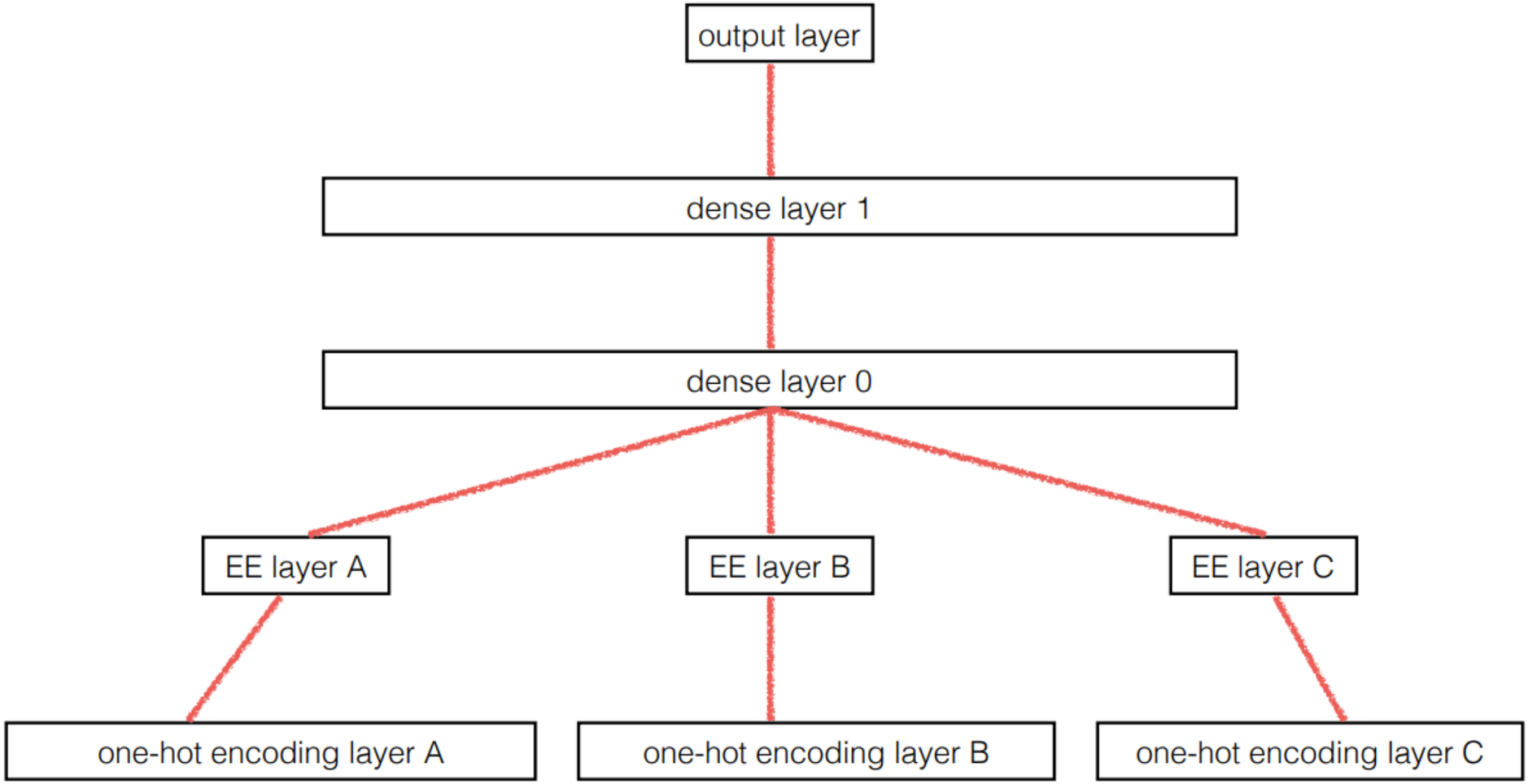
Entity Embeddings of Categorical Variables

C. Guo, F. Berkhahn (2016)

Objetivo es predecir (log) ventas diarias de cada tienda de la cadena Rossman

- 2.5 años de datos para 1115 tiendas.
- 1.017.210 registros, divididos en 90% para entrenamiento y 10% para test.
- No se utilizó *feature engineering*.
- Compara el rendimiento de redes neuronales basadas en *embeddings* categóricos, vs otros modelos de ML: XGBoost, Random Forest, KNN.

feature	data type	number of values	EE dimension
store	nominal	1115	10
day of week	ordinal	7	6
day	ordinal	31	10
month	ordinal	12	6
year	ordinal	3 (2013-2015)	2
promotion	binary	2	1
state	nominal	12	6


output layer
dense layer 1
dense layer 0
EE layer A
EE layer B
EE layer C
one-hot encoding layer A
one-hot encoding layer B
one-hot encoding layer C

xgboost	
max_depth	10
eta	0.02
objective	reg:linear
colsample_bytree	0.7
subsample	0.7
num_round	3000

random forest	
n_estimators	200
max_depth	35
min_samples_split	2
min_samples_leaf	1

KNN	
n_neighbors	10
weights	distance
p	1

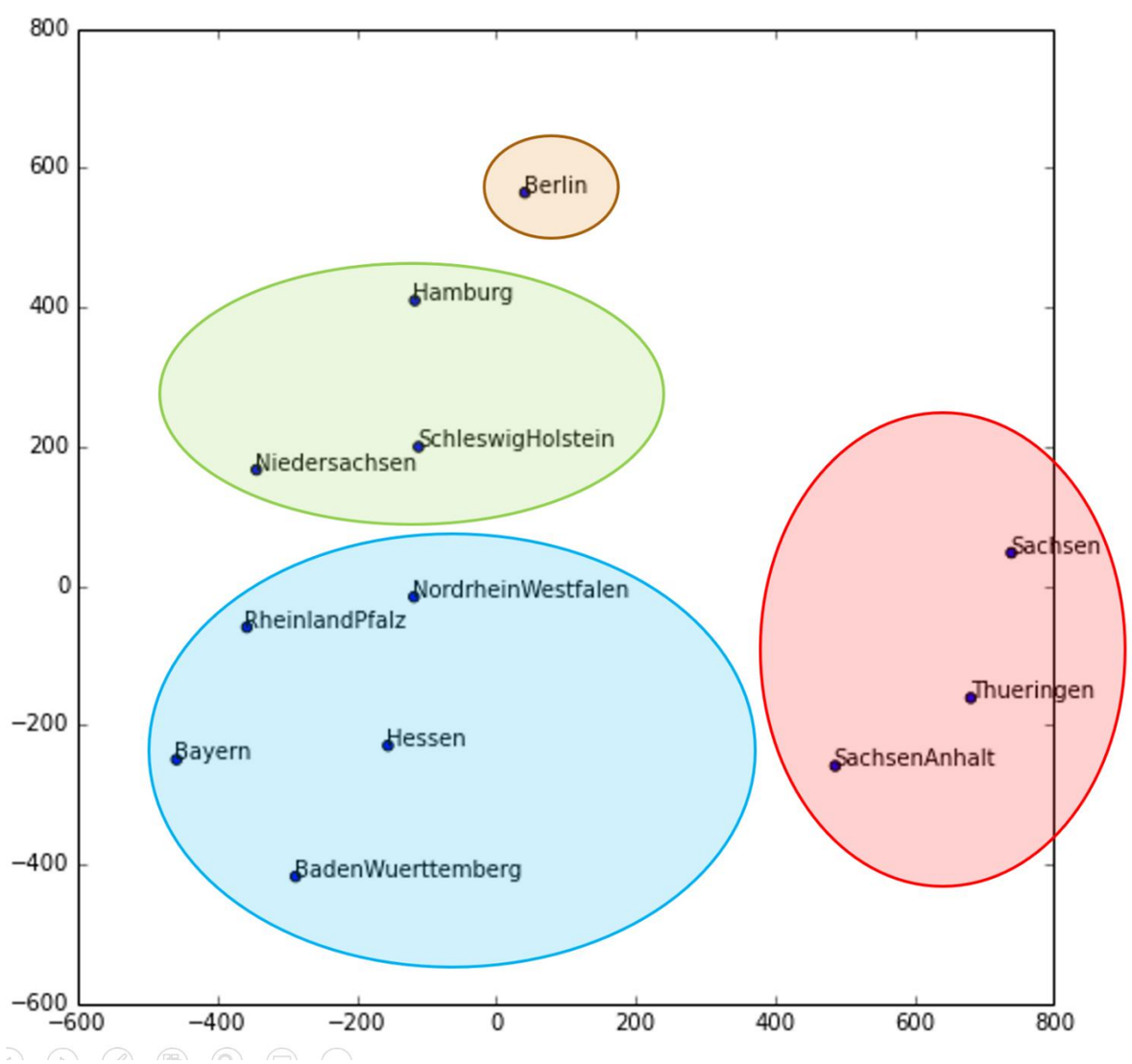
$$MAPE = \left\langle \left| \frac{Sales - Sales_{predict}}{Sales} \right| \right\rangle$$

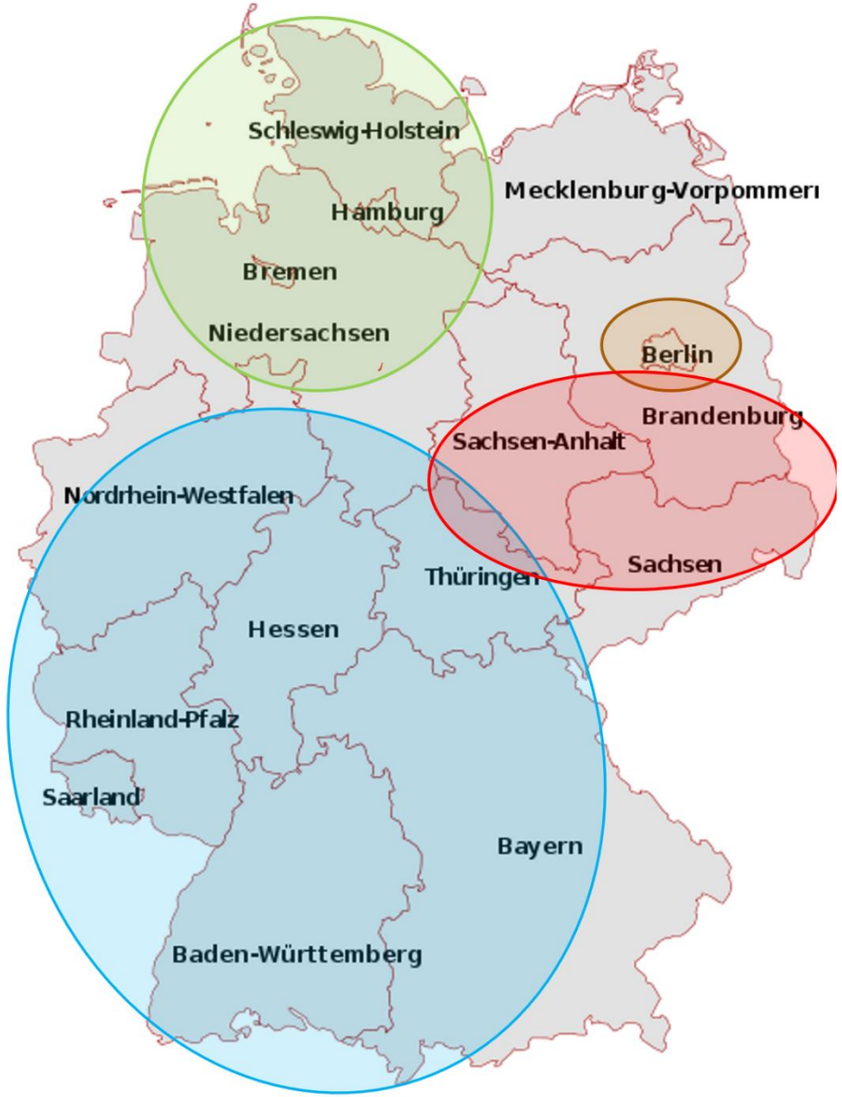
method	MAPE	MAPE (with EE)
KNN	0.315	0.099
random forest	0.167	0.089
gradient boosted trees	0.122	0.071
neural network	0.070	0.070

TABLE III. Comparison of different methods on the Kaggle Rossmann dataset with 10% shuffled data used for testing and 200,000 random samples from the remaining 90% for training.

method	MAPE	MAPE (with EE)
KNN	0.290	0.116
random forest	0.158	0.108
gradient boosted trees	0.152	0.115
neural network	0.101	0.093

TABLE IV. Same as Table IV except the data is not shuffled and the test data is the latest 10% of the data. This result shows the models generalization ability based on what they have learned from the training data.


800
600
400
200
0
−200
−400
−600
−600
−400
−200
0
200
400
600
800
Berlin
Hamburg
SchleswigHolstein
Niedersachsen
Sachsen
NordrheinWestfalen
RheinlandPfalz
Thueringen
Hessen
Bayern
SachsenAnhalt
BadenWuerttemberg


Schleswig-Holstein
Mecklenburg-Vorpommern
Hamburg
Bremen
Niedersachsen
Berlin
Brandenburg
Sachsen-Anhalt
Nordrhein-Westfalen
Sachsen
Thüringen
Hessen
Rheinland-Pfalz
Saarland
Bayern
Baden-Württemberg

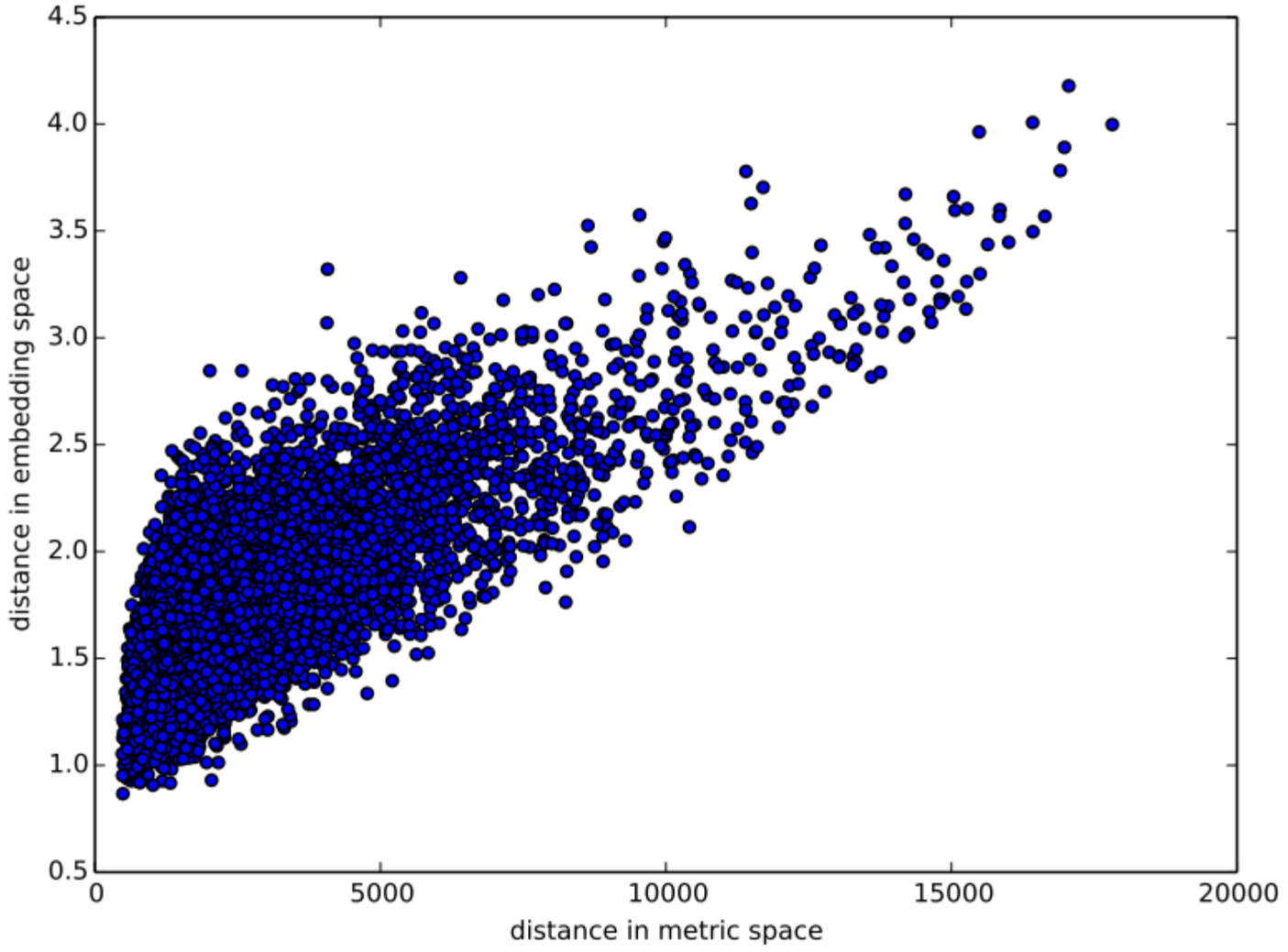
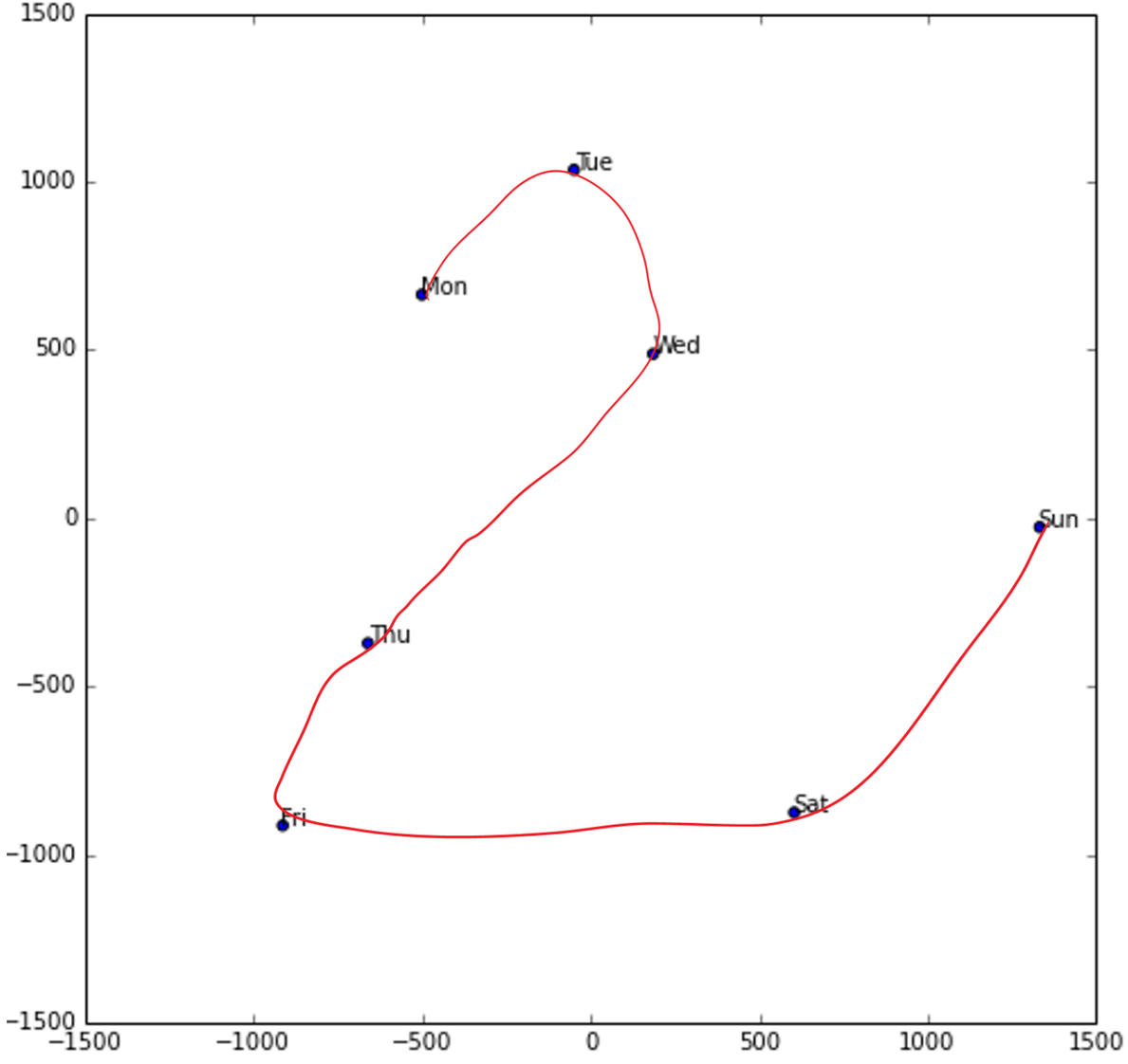
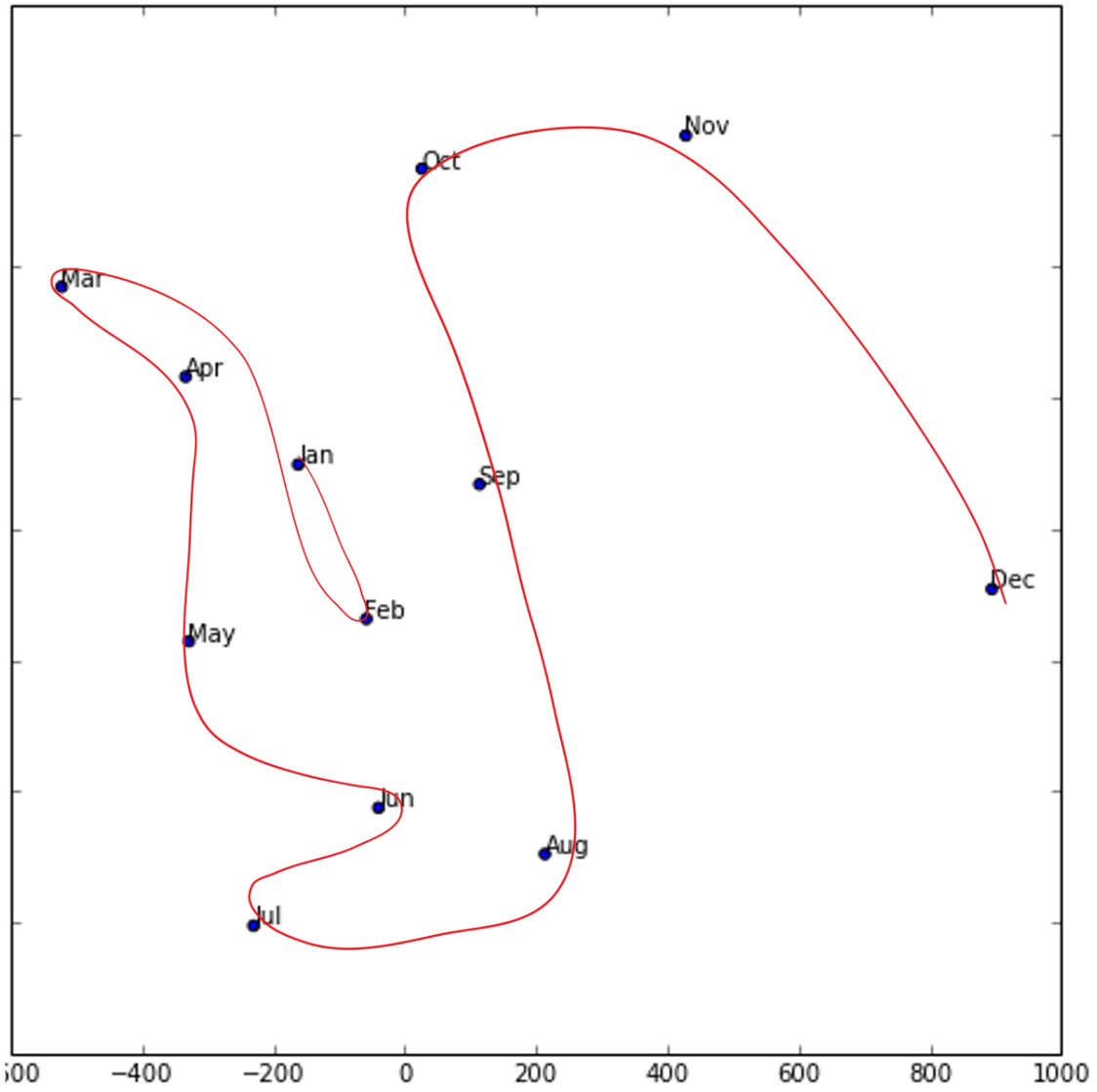


FIG. 2. Distance in the store embedding space versus distance in the metric space for 10000 random pair of stores.


Tue
Mon
Wed
Sun
Thu
Fri
Sat


Nov
Oct
Mar
Apr
Jan
Sep
Dec
Feb
May
Jun
Aug
Jul

Centrémonos ahora en el punto 2. Distribución y correlación en los datos

- Los problemas con la alta correlación entre variables y el alto desbalance de las clases son transversal a todas las técnicas de Deep Learning.

- Debido a esto, las técnicas para enfrentarlos las cubriremos en otros capítulos, al revisar los modelos donde originalmente fueron aplicadas (regularización, mecanismo de atención, y otros).

- Sin embargo, aunque hayan sido propuestas en otro contexto, siguen siendo aplicables a datos tabulados. Dos ejemplos son:

  - TabNet: Attentive Interpretable Tabular Learning (https://arxiv.org/pdf/1908.07442.pdf)

  - Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data (https://arxiv.org/abs/1909.06312)

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Sistemas Urbanos Inteligentes

Redes Neuronales y Datos Tabulados

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación