Deep and Confident Prediction For Time Series at Uber

(2017)

Introduction

- Uncertainty estimation implementation for deep prediction models
- Quantifies the prediction uncertainty from 3 sources:
  - Model uncertainty
  - Inherent noise
  - Model misspecification
- Anomaly detection

-> 즉, point estimation만 하는 것이 아니라 uncertainty estimation도 하고 이를 이용하여 예측뿐만 아니라 이상치탐지도 할 수 있는 모델링

Time Series Prediction

- 딥러닝모델을 이용한 시계열예측은 다양하게 제안되고 있다.
  - cnn, rnn, transformer, hybrid 등등
- 시계열예측도 그 목적에 따라 방향성이 다른데 이 논문은 uncertainty estimation에 초점을 두었다고 할 수 있다.

- 딥러닝을 이용한 시계열 예측의 일반적인 장점
  - End-to-end
  - 복잡한 시계열 특징을 잡아낼 수 있다.
  - Exogenous 변수들을 함께 이용하기 편하다.
  - 다변량시계열 변수들끼리의 nonlinear한 관계를 잘 잡는다.
  - Feature extraction이 좋다.

Bayesian Neural Network

- 딥러닝에서 parameter들은 deterministic한데 이를 bayesian으로 접근
- 그런데 복잡한 non-linearity와 non-conjugacy로 인해 posterior inference 가 쉽지 않다.

- 이 논문에 사용한 방법은 Monte Carlo Dropout
  - 그 때 당시 상당히 획기적인 방법이었고
  - 지금은 다른 방법들이 더 나왔으며 현재 MC dropout 방법론은 덜 인기

Prediction Uncertainty

- Goal : evaluate the uncertainty of the model prediction
  - 아래의 식에서 예측치 $\hat{y}$뿐만 아니라
  - Prediction standard error η도 구하는 것

$$[\hat{y} - z_{\frac{\alpha}{2}}\eta, \hat{y} + z_{\frac{\alpha}{2}}\eta]$$

Prediction Uncertainty

- Neural network as function $f^W(\cdot)$
  - W의 prior와 regression setting

$$W \sim N(0, I)$$

$$y|W \sim N(f^W(x), \sigma^2)$$

Prediction Uncertainty

- Prediction distribution in Bayesian

$$p(y^*|x^*) = \int_w p(y^*|f^w(x^*))p(W|X,Y)dW$$

- Variance of the prediction distribution quantifies the prediction uncertainty!

$$Var(y^*|x^*) = Var\left[E[y^*|W,x^*]\right] + E\left[Var[y^*|W,x^*]\right]$$
$$= Var[f^w(x^*)] + \sigma^2$$

Model uncertainty

Inherent noise

Prediction Uncertainty

- 그런데 앞선 식에서 가정하는 것은 new data y^*가 동일한 procedure에서 generate되었다는 것이다.
- 하지만 시계열에서 항상 그렇다고 할 수는 없다.
  - 예를 들어, 매출예측을 하는데 갑자기 코로나 확진자의 증가로 인해 평소의 일반적인 매출과 다른 양상의 매출이 발생할 수 있다.

- 따라서 저자는 이런 추가적인 uncertainty를 **model misspecification**으로 정의하고 이를 estimation하기 위한 접근법도 추가로 진행한다.

-> uncertainty = model uncertainty + model misspecification + noise
이제 하나씩 이들을 측정하는 방법에 대해 알아보자!

Model Uncertainty

- 먼저 model uncertainty에 대해 MC dropout방법론을 이용하였다.

- MC dropout
  1. 훈련된 모델에 new data x^*를 넣을 때, randomly dropout each hidden unit with certain probability p
  2. 이를 B번 반복한다. 그러면 우리는 $\{\hat{y}^*_{(1)}, \hat{y}^*_{(2)}, \dots, \hat{y}^*_{(B)}\}$
  3. 이를 이용하여 아래처럼 model uncertainty를 approximate한다.

$$\widehat{Var}[f^w(x^*)] = \frac{1}{B}\sum_{b=1}\left(\hat{y}^*_{(b)} - \overline{\hat{y}^*}\right)^2$$

Model Misspecification

- New data가 새로운 patterns를 보일 때, 그 uncertainty를 잡고 싶다!
  - Encoder-Decoder의 구조를 이용
- 저자들은 Encoder-Decoder를 훈련시킨 뒤에 encoder를 통해 representative features를 뽑아내서 사용하였다. -> 여기에 dropout을 통해 model misspecification을 고려
- Encoder를 통해 구한 features를 다른 예측모델(MLP)에 넣어서 예측을 진행한다.
- 이때, encode와 예측모델에 모두 **dropout를 적용**하여 model misspecification도 고려해준다. 아래의 식에 misspecification도 같이 들어가는 것!

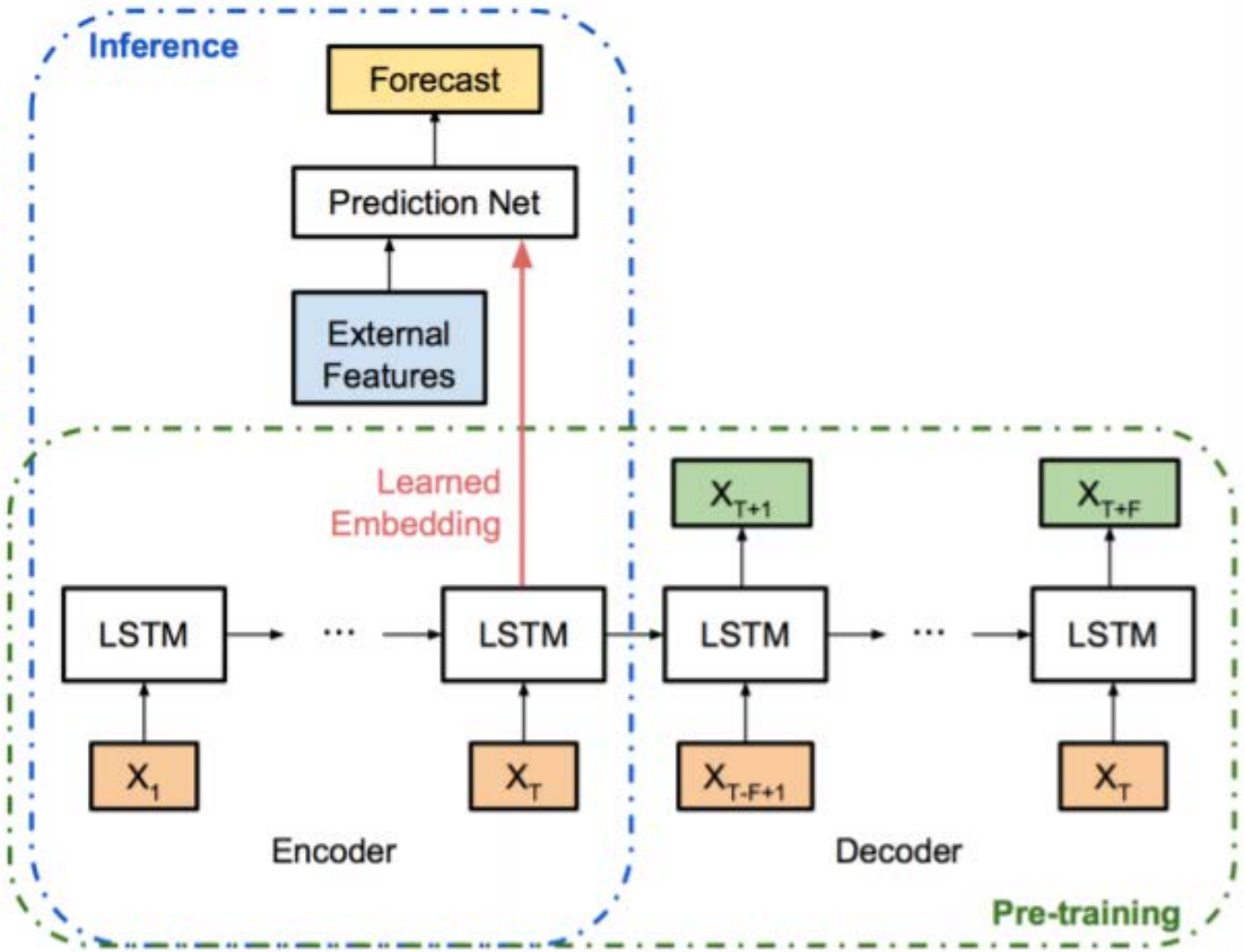
$$\widehat{Var}[f^{w}(x^{*})] = \frac{1}{B}\sum_{b=1}\left(\hat{y}^{*}_{(b)} - \overline{\hat{y}^{*}}\right)^{2}$$

Inherent noise

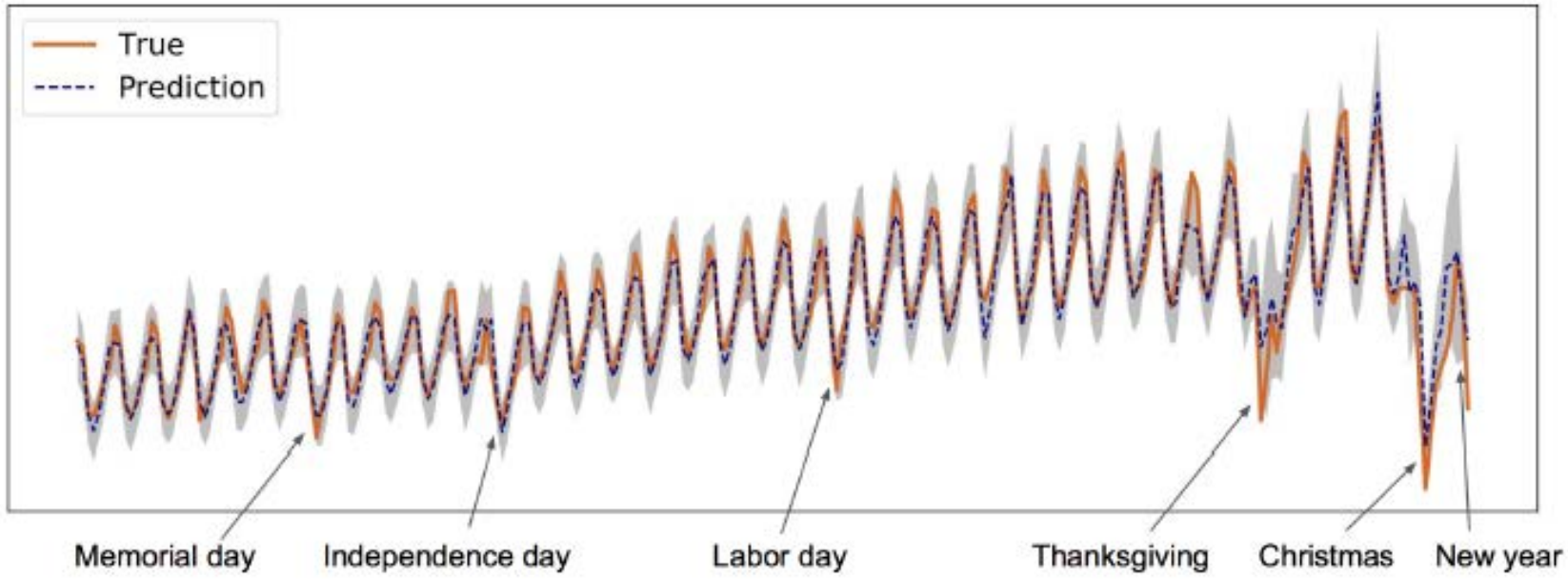
- Residual sum of squares evaluated on an independent held-out validation set 으로 noise를 estimate한다.
- Train data가 커질수록 σ^2에 대한 unbiased estimator가 된다고 한다.

$$\hat{\sigma}^2 = \frac{1}{V}\sum_{v=1}(y_v - f^w(x_v))^2$$

Model Design

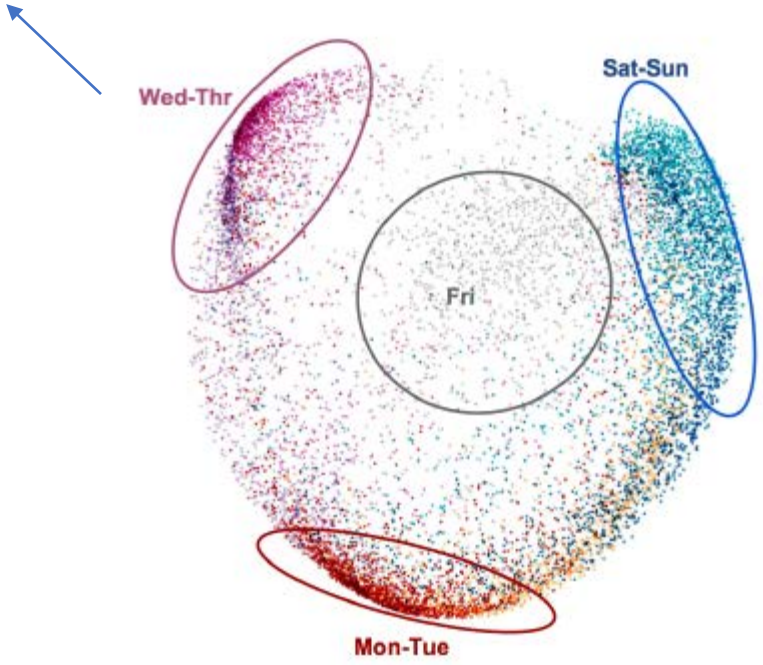


Model Design

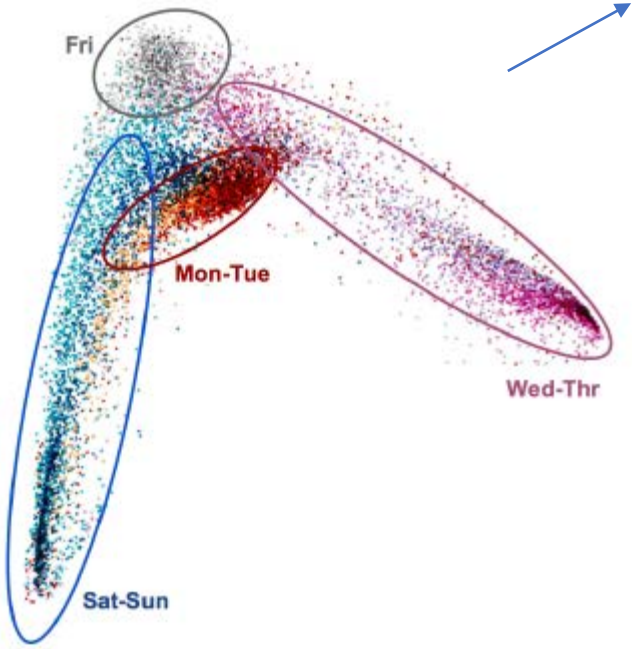


Conclusion

1번째 LSTM layer를
통과한 embedding



2번째 LSTM layer를
통과한 embedding



Conclusion

- 예측력이 좋다 (TABLE 1)
- Predictive Intervals도 3가지 uncertainty를 하는 것이 제일 좋다 (TABLE 2)

TABLE 1. SMAPE OF FOUR DIFFERENT PREDICTION MODELS, EVALUATED ON THE TEST DATA.

City	Last-Day	QRF	LSTM	Our Model
Atlanta	15.9	13.2	11.0	7.3
Boston	13.6	15.4	10.0	8.2
Chicago	16.0	12.7	9.5	6.1
Los Angeles	12.3	10.9	8.5	4.7
New York City	11.5	10.9	8.7	6.1
San Francisco	10.7	11.8	7.3	4.5
Toronto	15.2	11.7	10.0	5.3
Washington D.C.	13.0	13.3	8.2	5.2
Average	**13.5**	**12.5**	**9.2**	**5.9**

TABLE 2. EMPIRICAL COVERAGE OF 95% PREDICTIVE INTERVALS, EVALUATED ON THE TEST DATA.

City	PredNet	Enc+Pred	Enc+Pred+Noise
Atlanta	78.33%	91.25%	94.30%
Boston	85.93%	95.82%	99.24%
Chicago	71.86%	80.23%	90.49%
Los Angeles	76.43%	92.40%	94.30%
New York City	76.43%	85.55%	95.44%
San Francisco	78.33%	95.06%	96.20%
Toronto	80.23%	90.87%	94.68%
Washington D.C.	78.33%	93.54%	96.96%
Average	**78.23%**	**90.59%**	**95.20%**

Conclusion

- End-to-end로 uncertainty까지 알 수 있다는 점이 매력적!
- prediction + uncertainty + anomaly detection : 일타삼피 ㄷㄷ
- Encoder를 통해 feature extraction을 하고 이를 통해 따로 prediction model을 만든 것이 신기했다.
  - Feature extraction을 더 잘할 수 있는 방법을 생각해볼까
  - 아니면 decode도 추가로 이용?!
- 근데 훈련을 물론 예측에 걸리는 시간도 다소 오래 걸릴 것 같다.
  - 근데 Uber가 썼다니까 ㅎㅎ
- MC dropout보다 advanced한 방법론을 적용해볼까