SDML HW2 Task2 Report

B04902016 曾奕青

B04902103 蔡昀達

B04902105 戴培倫

November 20, 2018

1 Preprocess

day D-2				day D-1					day D		
1	2	3	4	5	6	7	8	9	10	11	12

Table 1: user A food sequence

Method 1 將每個 user 的資料當成一個很長的 sequence。

set sequence length = 8, stride = 1

sequence for training would be:

[1, 2, 3, 4, 5, 6, 7, 8]

[2, 3, 4, 5, 6, 7, 8, 9]

[3, 4, 5, 6, 7, 8, 9, 10]

[4, 5, 6, 7, 8, 9, 10, 11]

[5, 6, 7, 8, 9, 10, 11, 12]

Method 2 使 sequence 爲前面的 sequence + 今日的一個食物。

set sequence length = 8

sequence for training would be:

[3, 4, 5, 6, 7, 8, 9, 10]

[3, 4, 5, 6, 7, 8, 9, 11]

[3, 4, 5, 6, 7, 8, 9, 12]

2 Spotlight

Make use of spotlight implicit recommender model

https://github.com/maciejkula/spotlight

Model Basic LSTM

For a sequence [1,2,3,4,5,6], optimizes the loss of [1] predicts [2], [1,2] predicts [3], [1,2,3] predicts [4]...

Loss function: adaptive hinge loss

Negative samples: random sequence

Experiments

- Single model
  - public : 0.17561
  - private : 0.17492
- Ensemble models using different sequence length
  - public : 0.20014
  - private : 0.19710
- Use method 2 preprocessing (Single model)
  - public : 0.20665
  - private : 0.20667
- Change negative sampling method using food less eaten
  –> Failed

3 LSTM

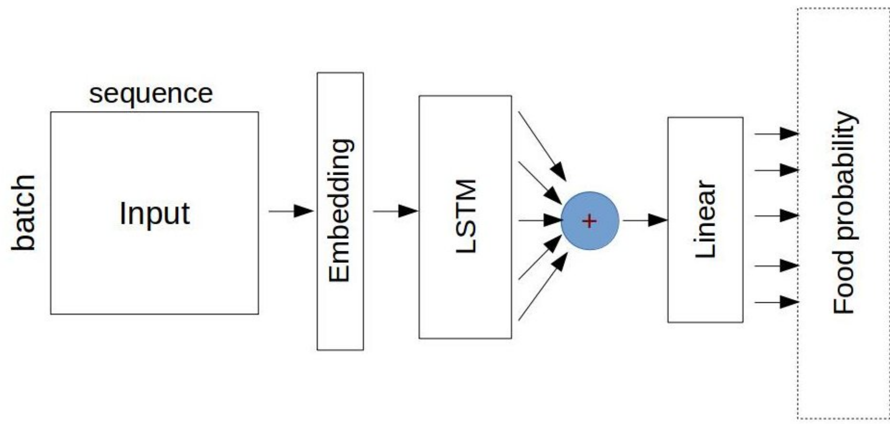


Figure 1: Model Structure

Uses method 2 data preprecessing.

Training Procedure

1. Split training / testing data.
2. Train on training data.
3. Find best epoch E (by MAP of testing data)
4. Train E epochs on all dataset
5. Train one epoch on testing data to fine tune

Experiments

- Sum all timesteps after LSTM
  - Validation score during training: 0.248
  - After training E epochs on all dataset
    - public : 0.28026

    * private : 0.27887
  - After fine-tuning:
    * public : 0.29071
    * private : 0.28954

- Maximum of all timesteps after LSTM

score \ seqlen	20	40	60	80	100	120
public	0.26710	0.28375	0.28551	0.28839	**0.29056**	0.28885
private	0.26202	0.28141	0.28453	0.28678	**0.29017**	0.28955

- Add global user features

  From the results of heuristic models [section 4], it's clear to notice that food count is an important role for prediction. Therefore I used a simple VAE to encode 5533 dimension food count to a 64 dimension vector, and concatenate it with LSTM output, hoping it can capture long term features. However, it seemed to do no help to this model.

  - public : 0.28959
  - private : 0.28913

4 Heuristic Models

- Eaten food count
  - public : 0.28731
  - private : 0.28751
- Eaten food & popular food
  - public : 0.27402
  - private : 0.27435
- Recently eaten food (30 days)
  - public : 0.31428

  - private : 0.31454

- Food count with weight decay through time ($1/((days\ ago)^{1.08})$)
  - sequence length: 30 days
    * public : 0.33702
    * private : 0.33389
  - choose best sequence length for each user (by validating on last 5 days)

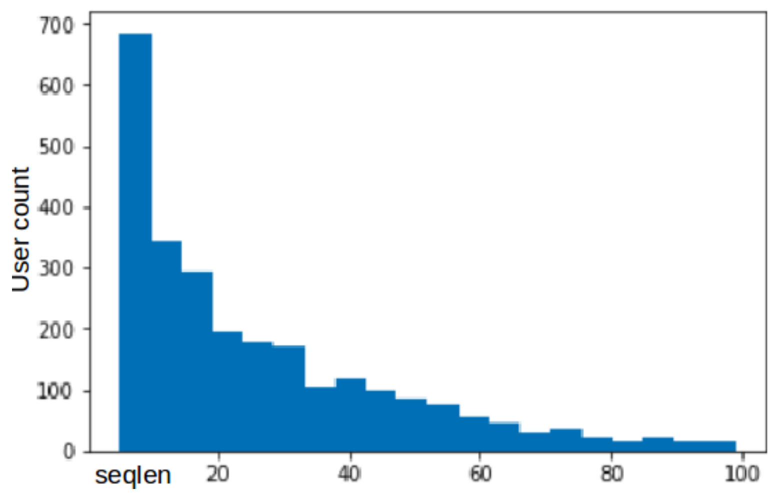


Figure 2: sequence length

Reaches 0.349 on validation set, however it seems to overfit.

    * public : 0.32917
    * private : 0.32869

- Filter out similar foods

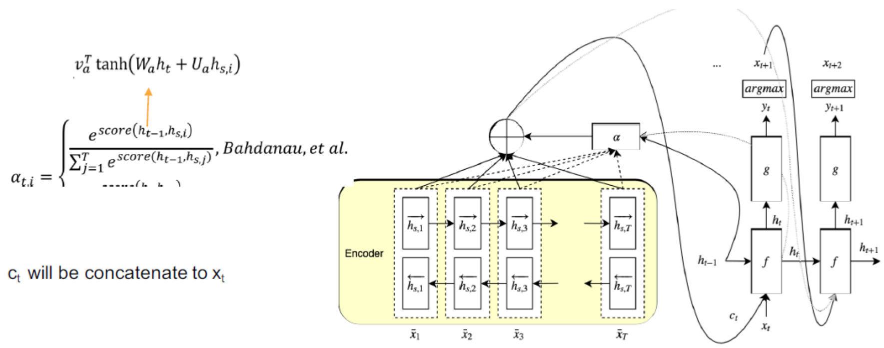
  Considering that users might not eat too much similar food in the same day, I made use of the food category ("_cat" and "_subcat" postfixes).

  However, whatever cosine similarity threshold and similar_food_count tolerence leads to terrible validation score.

5 Seq2seq

- public : 0.30803
- private : 0.30650

Use Bahdanau attention



6 best model

Results of models in this section are shown in Table 2

	private	public
model 1	0.34602	0.35096
model 1 + RNN	0.34490	0.35047
model 2	0.28620	0.28058
model 1 + model 2	**0.34954**	0.35268
model 1 + model 2	0.34927	**0.35316**

Table 2: kaggle score

6.1 model 1

對所有吃過的食物計算分數，對每個日期給予不同權重，越久以前吃的權重越低。因為這樣肯定只能 fit 某一天的結果，所以在決定參數時肯定是不能拿 validation 太高的參數在為初始參數，決定初始參數以後再根據 testing data 做微調。

6.2 model 2

因爲 model 1 基本上就是 overfit 在 testing data 的 RNN，所以跟 RNN model 做 ensemble 基本上沒有用，在 Table 2也可以看到和 RNN 做 ensemble 結果甚至還會變差。因此，我把 model 1 跟 mDAE[1] 做 ensemble，ensemble 方法爲只取兩 model 的前 20 名，依據名次給予各個食物不同分數，最後分數總合越高的食物排在越前面。不過在 Table 2可以看到 mDAE 的 performance 和 model 1 其實有一段差距，所以必須給兩個 model 不同的權重，最後可達到 0.35 的 kaggle score。

References

[1] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. *Variational Autoencoders for Collaborative Filtering*. In WWW. ACM, 689–698.