Some notes on Machine Learning

March 6, 2015

Contents

1 Data scaling

2 Loss functions

Optimally model parameters are usually defined via loss functions. The parameters are obtained by minimizing the ensuing Lagrangian. Typical loss functions are

- The *least square loss function* is often used for regression problems [10],

$$l^{ls}(y_1, y_2) := (y_1 - y_2)^2 \tag{1}$$

- The l*ogistic/sigmoid function* is often used for binary classification problems ($y_i \in \{-1, 1\}$),

$$l^{c}(y_1, y_2) := -\ln \sigma(y_1 - y_2)^2 \tag{2}$$

If the number of model parameters is large, overfitting may be avoided by including a regularization term to the Lagrangian.

3 Categorical features

Categorial features may be treated with indicator variables (aka dummy, design, categorial, qualitative, or binary variable, or Boolean indicator)

Caution: (Dummy variable trap)

- Herbrich p. 2 [4]
- Indicator variables

4 Similarity metric

5 Collaborative filtering

Collaborative filtering methods are widely used in recommendation engines. For reviews see Refs.[12, 1, 5]

Idea We would like to adapt the infamous collaborative filtering method to recommend for a given item (Artikel) a convenient addon-item (Zusatzartikel). This machinery has been successfully employed by Amazon or in the Netflix challenge [7, 2]. In these approaches the goal is to recommend for a given user suitable items. In the following we will keep the notion of user and item. However, in order to adapt to our usecase we eventually translate according to

$$\begin{aligned} \text{user} &\to \text{Artikel} \\ \text{item} &\to \text{Zusatzartikel} \end{aligned}$$

First, define the rating matrix $R \in \mathbb{R}^{m \times n}$ matrix, by

$$r = \{r_{ui}\}_{1 \le u \le m, 1 \le i \le n}, \tag{3}$$

where we introduce the index convention u, v for users and for items i, j, k.

5.1 Neighborhood-based collaborative filtering

Following Ref. [2], we distinguish between the *user-oriented* and *item-oriented* collaborative filtering approach. In the user-oriented approach we consider the set of users $N(u; i)$ that tend to rate similar to user u ("neighbors"), and that acutally rated item i. Thus r_{vi} is known $\forall v \in N(u; i)$. The *predicted* value of r_{ui} is then obtained as a weighted average of the neighbors' rates,

$$r_{ui} = \frac{\sum_{v \in N(u;i)} s_{uv} r_{vi}}{\sum_{v \in N(u;i)} s_{uv}}. \tag{4}$$

This is essentially a normalized matrix product over the domain of neighbors of user u. Eq. (4) may be interpreted as weighted average of the neighbors rating.

The item-oriented approach is similar, but instead of similar users we work with similar items. In order to estimate r_{ui} in this approach, we introduce the set of neighboring items $N(i; u)$ that other users tend to rate similarly to their

rating of i. Thus, r_{uj} is known $\forall j \in N(i;u)$. Then the *predicted* value of r_{ui} is obtained according to

$$r_{ui} = \frac{\sum_{j \in N(i;u)} s_{ij} r_{uj}}{\sum_{j \in N(i;u)} s_{ij}}. \tag{5}$$

Both, user-based and item-based collaborative filtering are very similar in structure. However, it has been shown that the latter gives better-quality estimates and tends to be more efficient [11].

Note, that the similarity matrix s plays a crucial role because it determines the neigbors and enters as weight in the Matrix multiplication above. Different collaboartive filtering methods may differ in their choice for s (one common choice is the Pearson correlation). Methods also differ in the how they normalize or center data before predicting via Eq. (4) or (5).

5.2 Obtaining the weights

In principle the interpolation weights (i.e., components of the similarity matrix s) could be incorporated by any incarnation of a similarity matrix. However, such an approach has several drawbacks [2]

- The similarity function is arbitrary
- Overfitting may occur if the interpolation weights sum to one

Alternatively, one could incorporate the weights directly in the learning process. Given the set of K neighbors $N(i;u)$ we would like to compute the interpolation weights $\{w_{ij} | j \in N(i;u)\}$ such that we obtain the best prediction rule according to

$$r_{ui} = \sum_{j \in N(i;u)} w_{ij} r_{uj} \tag{6}$$

For the basic idea, consider the hypothetical dense case where all users but u rated both, i and all its neighbors $N(i;u)$. Then we can learn thte interpolation weights by modeling the relationships between item i and its neighbors by a least square problem,

$$\min_w \sum_{v \neq u} \left(r_{vi} - \sum_{j \in N(i;u)} w_{ij} r_{vj} \right)^2, \tag{7}$$

which eventually leads to an eigenvalue problem of the form,

$$Aw = b \tag{8}$$

$$A_{jk} = \sum_{v \neq u} r_{vj} r_{vk} \tag{9}$$

$$b_j = \sum_{v \neq u} r_{vj} r_{vi} \tag{10}$$

with $A \in \mathbb{R}^{K \times K}$ and $b \in \mathbb{R}^K$ (K beeing the number of neighbors).

However, for a sparse rating matrix there are likely very few users who rated i and all it's neighbors $N(i;u)$, and/or A could be singular. In order to overcome this limitation, define the set $U(j,k)$ of users who rated both j and k. The problem may be reformulated by [2]

$$\hat{A}w = \hat{b}, \tag{11}$$

with

$$\hat{A}_{jk} = \frac{|U(j,k)|\bar{A}_{jk} + \beta\, avg}{|U(j,k)| + \beta}, \tag{12}$$

$$\bar{A}_{jk} = \frac{\sum_{v\in U(j,k)} r_{vj} r_{vk}}{|U(j,k)|}, \tag{13}$$

and

$$\hat{b}_j = \frac{|U(j,k)|\bar{b}_j + \beta\, avg}{|U(j,k)| + \beta} \tag{14}$$

$$\bar{b}_j = \frac{\sum_{v\in U(j,k)} r_{vj} r_{vi}}{|U(j,k)|}. \tag{15}$$

Thus A and b are estimated by $\hat{A}$ and $\hat{b}$, where the product $\beta\, avg$ accounts for a shrinkage towards a common mean with shrinkage parameter β and the baseline vale avg. In Ref. [2] two different baseline averages are combined by averaging over diagonal and non-diagonal entries respectively.

5.3 Calculation setup

In summary we have the following calculation procedure:

1. Calculate s_{ij} via the Pearson correlation.
2. Shrink s_{ij} them based on their support. I.e., multiply them by $|U(i,j)|/(|U(i,j)|+\alpha)$ for some small α
3. Pre-compute $\hat{A}$ and $\hat{b}$
4. Compute interpolation weights by inverting (eq. 11).
5. Predict according to eq. (6)

5.4 Additional issues

- Normalization by removing global effects
- Cold start problem

6 Matrix Factorization

Matrix factorization is another popular method for recommendation engines. [6, 10]

7 Cold Start Problem

The cold start problem is referred to as the problem of recommending articles that have not been rated so far. There are several suggestions in the literature to overcome the cold start problem.

- Matrix factorization techniques [13, 6]
- Hybrid methods that combine user ratings, and item/user features [8]
- Boltzmann machines [3]
- Information theoretic approach [9]

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] Robert M. Bell and Yehuda Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 43–52, Washington, DC, USA, 2007. IEEE Computer Society.

[3] Asela Gunawardana and Christopher Meek. A unified approach to building hybrid recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 117–124, New York, NY, USA, 2009. ACM.

[4] Ralf Herbrich, Thore Graepel, and David Stern. Recommender system, January 8 2015. US Patent 20,150,012,378.

[5] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, January 2004.

[6] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[7] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[8] Seung-Taek Park and Wei Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 21–28, New York, NY, USA, 2009. ACM.

[9] Al Mamunur Rashid, George Karypis, and John Riedl. Learning preferences of new users in recommender systems: An information theoretic approach. *SIGKDD Explor. Newsl.*, 10(2):90–100, December 2008.

[10] Steffen Rendle. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[11] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[12] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, January 2009.

[13] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 315–324, New York, NY, USA, 2011. ACM.