A Hybrid Approach to Recommending Recipes with Textual Information

Yinghao Sun, Yuhui Huang | {sunmo, yhuang77} @stanford.edu

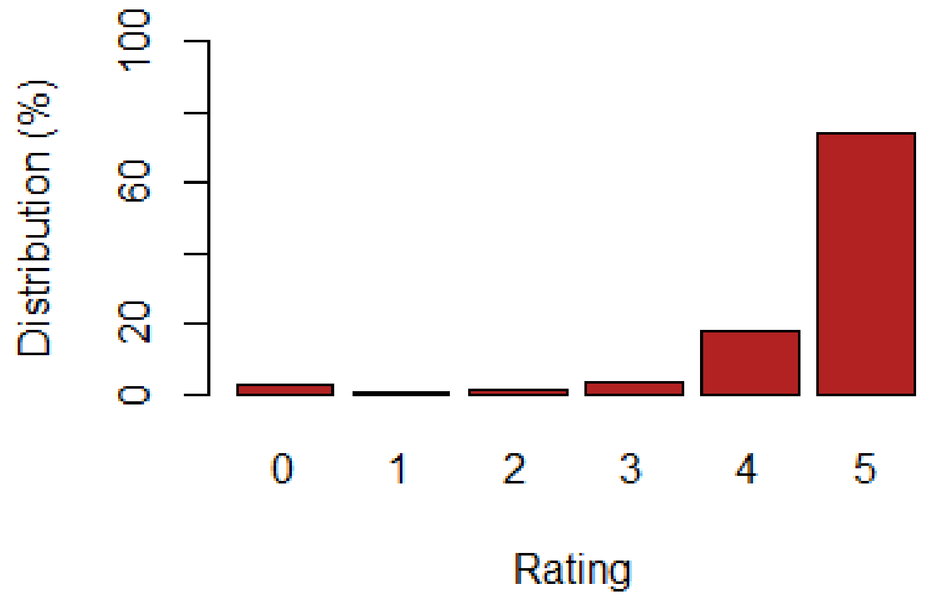
Motivation

The project aims to predict rating by a user on a recipe based on : 1) previous ratings; 2) recipe attributes. We hope to combine collaborative and content-based filtering by **augmenting recipe features with latent factors extracted from matrix factorization** (input) to feed into any reasonable supervised learning algorithms to predict the **ratings** (output).

Data and Features

Data (*source from Kaggle*[1])**:**

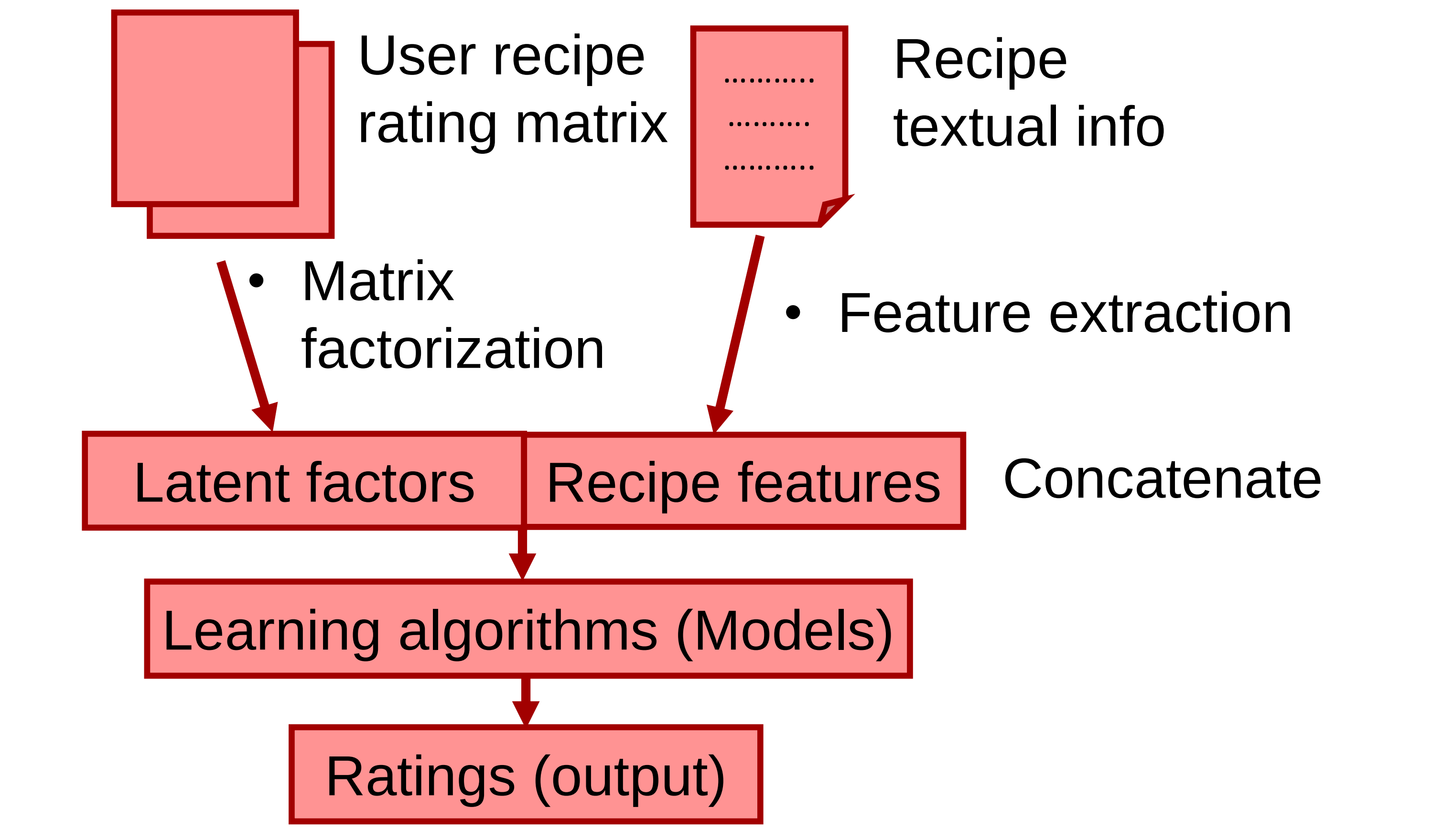
- over 180,000 recipes with attributes (tags, ingredients, descriptions, etc.)
- over 700,000 ratings covering years of user interactions on Food.com



Features:

Raw Features	Transformed Features
Known user-recipe ratings	User/recipe latent factors via: • Matrix factorization
Tags, ingredients, descriptions	Dense features via: • tf-idf + truncated SVD • GloVe • Latent Dirichlet Allocation
Cooking steps	A vector indicating occurrence of 58 cooking techniques

Approach



User recipe rating matrix

Recipe textual info

• Matrix factorization

• Feature extraction

Latent factors | Recipe features

Concatenate

Learning algorithms (Models)

Ratings (output)

Models

- **Matrix factorization-based collaborative filtering:** define predicted rating: $r_{u,i} = \mu + b_u + b_i + q_i^T p_u$ minimize loss function:

$$\sum_{(u,i)\in I} (\hat{r}_{u,i} - r_{u,i})^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2)$$

- **LASSO and Elastic Net**: minimize loss function

$$||y - X\beta||_2^2 + \lambda((1-\sigma)||\beta||_2^2 + \sigma||\beta||_1)$$

- **Approximate radial basis function (RBF) kernel regression**: Nystrom low-rank approximation of the RBF kernel matrix, then minimize

$$\psi(\hat{y}, y) = \max\{0, (\hat{y} - y)^2 - \epsilon\}$$

- **k-Nearest Neighbors (k-NN)**: predictions are generated based on the averaged rating from k nearest neighbors determined by KD-Tree
- **Random Forest**: fit a number of decision trees based on a number of sub-samples of the training set
- **XGBoosting**: combine weak learners (typically shallow decision trees) into a single strong learner in an iterative fashion

Results

Best Model: RBF kernel regression on latent factor + LDA extraction of textual info

Test Set Performance: RMSE = 1.3020, MAE = 0.8517

Table below shows performance of various combinations of features, sampling strategies, and learning algorithms' (parameters tuned via 3-fold **cross-validation on the training set**). Textual feature extractions are also tuned for better and comparable performance.

Algorithm/Sampling	Features [validation RMSE (training RMSE)]				
	latent factors only	latent/tf-idf+SVD	latent/GloVe	latent/LDA	latent/LDA/tech
ElasticNet/8% random	1.2618 (0.8111)	1.2619 (0.8167)	1.2625 (0.8123)	1.2626 (0.8105)	1.2637 (0.8082)
ElasticNet/8% balanced	1.3096 (1.2279)	1.3137 (1.2234)	1.3147 (1.2107)	1.3121 (1.2129)	1.3137 (1.2151)
Approx Kernel Reg/8% random	1.2677 (0.8016)	1.2626 (0.8015)	1.2582 (0.8064)	1.2578 (0.8039)	1.2588 (0.8072)
Approx Kernel Reg/8% balanced	1.3160 (1.2140)	1.3169 (1.2117)	1.3205 (1.2102)	1.3294 (1.2072)	1.3255 (1.2182)
k-NN/8% random	1.3469 (0.0)	1.3466 (0.0)	1.3553 (0.0)	1.3407 (0.0)	1.3505 (0.0)
k-NN/8% balanced	1.2927 (0.0)	1.2689 (0.0)	1.2923 (0.0)	1.2746 (0.0)	1.2752 (0.0)
RandomForest/8% random	1.2700 (0.7680)	1.2812 (0.7516)	1.2737 (0.7579)	1.2714 (0.7637)	1.2729 (0.7556)
RandomForest/8% balanced	1.3031 (1.2059)	1.3312 (1.2076)	1.3209 (1.2030)	1.3061 (1.2074)	1.3187 (1.2008)
XGBoosting/8% random	1.2707 (0.6519)	1.2804 (0.6581)	1.2773 (0.6486)	1.2781 (0.6490)	1.2680 (0.6577)
XGBoosting/8% balanced	1.9952 (1.7010)	2.0157 (1.6959)	1.9952 (1.7011)	2.0132 (1.7025)	1.9927 (1.6979)
XGBoosting/43% random	1.2641 (0.6835)	1.2668 (0.6993)	1.2591 (0.6996)	1.2660 (0.6920)	1.2634 (0.6856)
Matrix factorization/no sampling	1.2763 (0.7966)	-	-	-	-

latent/tf-idf+SVD: latent factors plus tf-idf truncated SVD features from tags, ingredients, descriptions; **latent/GloVe**:textual features were based on GloVe embeddings; **latent/LDA**: textual features from LDA; **latent/LDA/tech**: includes techniques from cooking steps; also note that we were able to increase sample size for xgboosting since it has an efficient implementation, but there is no need to increase for balanced sampling since it did not help with prediction.

Discussion

- RBF kernel regression slightly outperform matrix factorization on latent factors. It also seems to benefit from the addition of extracted textual features, especially with methods GloVe and LDA. This is expected because RBF kernel regression is able to recover complicated non-linear feature interactions.
- Most models have notable gaps between training RMSE and validation RMSE, indicating potential overfitting.
- Loss examples show that the predefined training set lacks rating data for some users in the test set, and that the distributions of user ratings are drastically different between training and test set for some particular users.

Future work: Apply stronger regularization and tune parameters more extensively to further reduce overfitting.

References: [1] MAJUMDER, B. P., LI, S., NI, J., AND MCAULEY, J. Generating personalized recipes from historical user preferences. arXiv preprint arXiv:1909.00105 (2019).