Content-based recommenders

Friday, 26 March 2021 07:07

Based on dependencies/correlations between item and user features
Similar to typical ML models

$$\underbrace{y}_{\text{response}} = f(\underbrace{x}_{\text{item and user features}} \mid \underbrace{\theta}_{\text{model parameters}})$$

response: e.g. interaction binary (0,1) or rating

Example (linear model)

$$y = \theta_0 + \theta_1 u_1 + \theta_2 u_2 + \theta_3 v_1 + \theta_4 v_2 + \theta_5 v_3$$

u_1, u_2: user features
v_1, v_2, v_3: item features

A sample trained model may look like that:

$$\text{n\_bought} = 4 + 0.5\ \text{age} + 0.01\ \text{avg\_income} + (-1)\ \text{price} + 0.2\ \text{quality} + (-5)\ \text{is\_used}$$

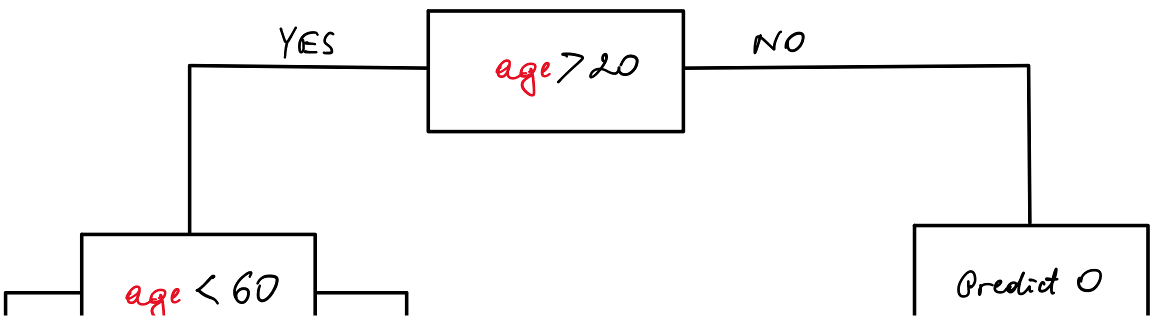
and a sample realization

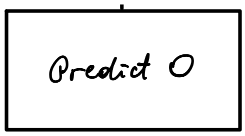
$$1 = 4 + 0.5 \cdot 30 + 0.01 \cdot 1200 - 30 + 0.2 \cdot 10 - 5 \cdot 1$$

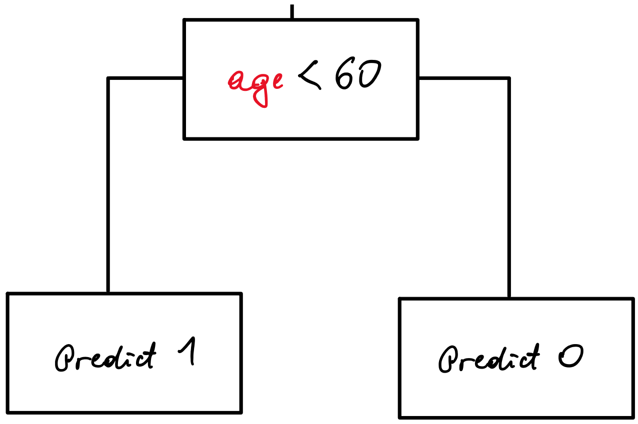
An ML model learns the weights θ to accurately predict the responses y observed historically based on features (explanatory variables) X

Remark Notice that in the linear model the user characteristics may have impact on the model error (e.g. RMSE) in predicting actual values, but will not have any impact on the ranking

Example (decision tree)







A nonlinear model is able to return personalized offers

Recommendations for active user u

Explicit feedback

Position	Item	Predicted rating
1	Rocky	4.9
2	Interstellar	4.7
3	Shrek	4.1
4	Star Wars	3.6
⋮	⋮	⋮

Implicit feedback

Position	Item	Probability / score
1	Rocky	0.95
2	Interstellar	0.91
3	Shrek	0.85
4	Star Wars	0.72
⋮	⋮	⋮

Typically items the user has not interacted with are evaluated

Explicit feedback

Any regression ML model can be used as a recommender in the explicit feedback case

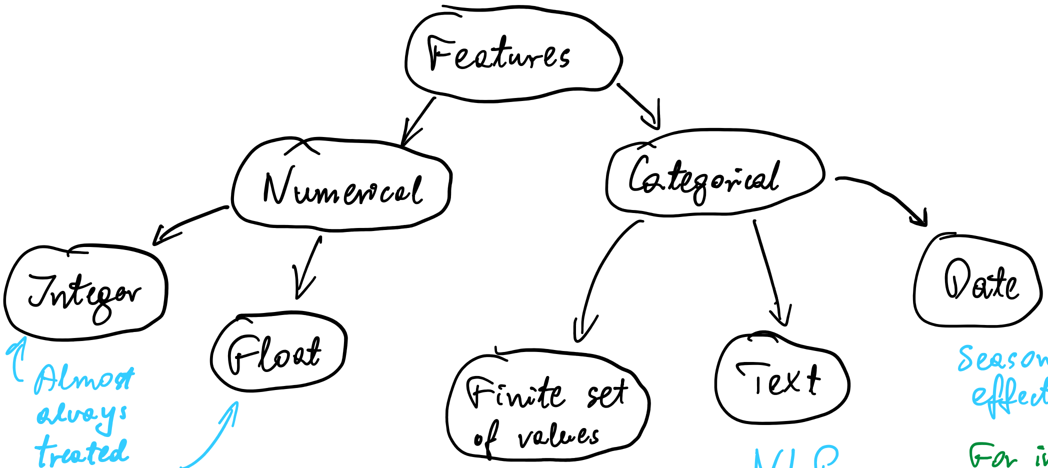
Implicit feedback

Any classification ML model returning probabilities can be used as a recommender in the implicit feedback case

Non-personalized : one linear model for all users

Personalized :
- one non-linear model for all users
- one model per cluster of users
- one model per user

Types of features



Almost always treated identically

For instance:
- length of a movie
- box-office result
- number of beds in a hotel room

Typically small

For instance:
- movie genres
- hotel room types
- ids

NLP

For instance:
- movie description
- movie title

Seasonal effects

For instance:
- arrival date

Categorical finite sets of values - one-hot encoding

One-hot encoding transforms a single column with N possible values into N columns with binary values

movie	genre
movie 1	sci-fi
movie 2	drama
movie 3	comedy
movie 4	sci-fi
⋮	⋮

→

movie	sci-fi	drama	comedy
movie 1	1	0	0
movie 2	0	1	0
movie 3	0	0	1
movie 4	1	0	0
⋮			

NLP

- one-hot
- n-grams
- embeddings (vectorization)

Dates

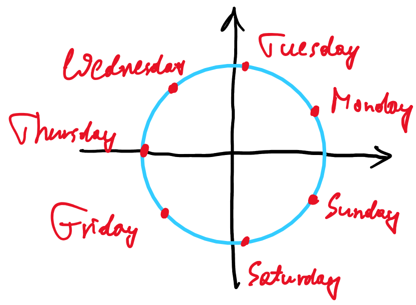
Examples:

One-hot encoded day of week

One-hot encoded month

	Monday	Tuesday	Wednesday	...
Monday →	1	0	0	...
Tuesday →	0	1	0	...

Day of week projected on a circle

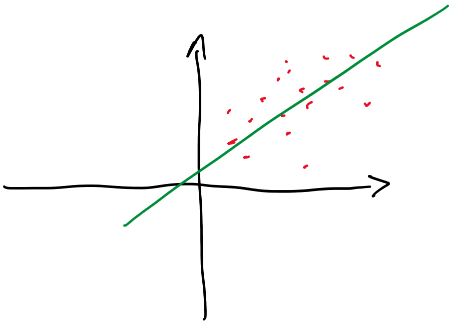


Thursday $\rightarrow [\underset{x}{-1}, \underset{y}{0}]$

Models

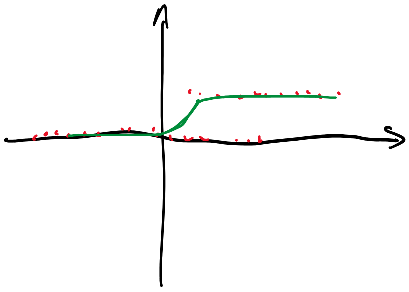
Linear

$$\hat{y} = f(x \mid \theta) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$



Logistic

$$\hat{y} = f(x \mid \theta) = \frac{1}{1 + e^{-\theta_0 - \theta_1 x_1 - \ldots - \theta_n x_n}}$$



$x_1, x_2, \ldots, x_n$ – numerical variables

$\theta_1, \theta_2, \ldots, \theta_n$ – trained parameters

$$\hat{y} = \begin{cases} \text{real ratings} & : \text{explicit feedback} \\ \text{real binary interactions} & : \text{implicit feedback} \end{cases}$$

Other very popular models

- SVR
- XGBoost
- Random Forest (RF)
- Decision Tree
- Naive Bayes
- Artificial Neural Networks (ANN)

Tuning hyperparameters

Many models have tunable parameters (hyperparameters)

$$\hat{y} = f(x \mid \theta, \tau)$$

- set τ
- train θ on the training set
- evaluate on the validation set

$\Big\}$ Iterate for many τ

- choose the best τ
- evaluate the model on the test set

TF-IDF Term Frequency - Inverse Document Frequency

Based on relative frequencies of feature values for a given user vs all users

Term frequency (TF) - how many times a given term appears in a given document (in our case for a given user)

Inverse document frequency (IDF) - natural logarithm of the number of documents (users) divided by the number of documents (users) with a given term

Example

user	concatenated genres
1	sci-fi, drama, sci-fi, sci-fi
2	comedy, comedy, drama
3	sci-fi, action, comedy
4	comedy, sci-fi, sci-fi

$tf(1, \text{sci-fi}) = 3$ $tf(1, \text{drama}) = 1$

$tf(2, \text{comedy}) = 2$ $tf(2, \text{drama}) = 1$

$tf(3, \text{sci-fi}) = 1$ $tf(3, \text{action}) = 1$ $tf(3, \text{comedy}) = 1$

$tf(4, \text{comedy}) = 1$ $tf(4, \text{sci-fi}) = 2$

$idf(\text{sci-fi}) = \ln\frac{4}{3}$ $idf(\text{drama}) = \ln\frac{4}{2} = \ln 2$

$idf(\text{comedy}) = \ln\frac{4}{3}$ $idf(\text{action}) = \ln\frac{4}{1} = \ln 4$

$tf\text{-}idf(1, \text{sci-fi}) = 3 \cdot \ln\frac{4}{3}$ $tf\text{-}idf(1, \text{drama}) = 1 \cdot \ln 2$

$tf\text{-}idf(2, \text{comedy}) = 2 \cdot \ln\frac{4}{3}$ $tf\text{-}idf(2, \text{comedy}) = 1 \cdot \ln\frac{4}{3}$

$tf\text{-}idf(3, \text{sci-fi}) = 1 \cdot \ln\frac{4}{3}$ $tf\text{-}idf(3, \text{action}) = 1 \cdot \ln 4$ $tf\text{-}idf(3, \text{comedy}) = 1 \cdot \ln\frac{4}{3}$

$tf\text{-}idf(4, \text{comedy}) = 1 \ln\frac{4}{3}$ $tf\text{-}idf(4, \text{sci-fi}) = 2 \cdot \ln\frac{4}{3}$

To get an item score take its features tf-idf average for a given user

$\boxed{\text{To get an item score take its features tf-idf average for a given user}}$

Example:

movie : sci-fi , action

user : 1

$$\text{score} = \frac{\text{tf-idf}(1, \text{sci-fi}) + \text{tf-idf}(1, \text{action})}{2}$$

$$= \frac{3 \cdot \ln\frac{4}{3} + 0}{2} = \frac{3}{2} \ln \frac{4}{3}$$