Basics, Un/Supervised, Overfitting, Train/Dev/Test

Thursday, September 28, 2023 11:12 AM

What we need to do data mining:

- Data with labels
- Clear objective to be optimized
- Enough data and data is good (meaningful, significant, etc)

Supervised Learning:

- Directly model relationships between inputs and outputs

Unsupervised Learning:

- Find patterns, relationships, structure in data

Overfitting: model performs well on training data but poorly on test data (poor generalization)

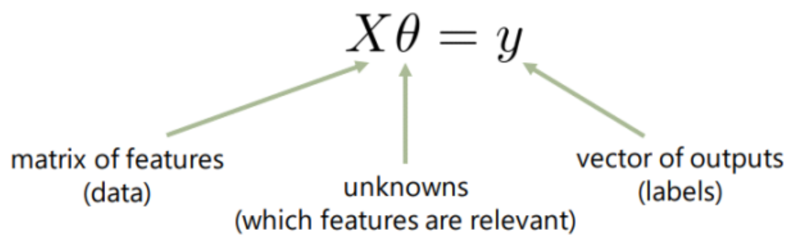
Train, Validation, Test:

- Train: fit models
- Validation: choose best model
- Test: get model performance

Regression, MSE, R^2

Thursday, September 28, 2023 11:01 AM

Linear regression:



$$\theta = \left(X^T X\right)^{-1} X^T y$$

Nonlinear transformations:

We can still perform linear regression on non linear transformations by:

$$\text{rating} = \theta_0 + \theta_1 \times \text{ABV} + \theta_2 \times \text{ABV}^2 + \theta_3 \times \text{ABV}^3$$

MSE: Mean Squared Error

$$= \frac{1}{N} \sum_{i=1}^{N} (y_i - X_i \cdot \theta)^2$$

R^2:

$$R^2 = 1 - FVU(f) = 1 - \frac{MSE(f)}{Var(y)}$$

R^2 = 0 → Trivial predictor

R^2 = 1 → Perfect predictor

Regularization, Gradient Descent

Thursday, October 5, 2023 11:52 AM

Regularization: balance accuracy with complexity

$$\arg\min_\theta = \frac{1}{N}\|y - X\theta\|_2^2 + \lambda\|\theta\|_2^2$$

= MSE + lambda * complexity

Gradient Descent: Optimize model by taking steps towards optimum

1. Initialize θ at random
2. While (not converged) do
   $\theta := \theta - \alpha f'(\theta)$

where $f(\theta)$ is the regularized model

Classification

Thursday, September 28, 2023 11:28 AM

Naïve Bayes: Associate a probability between labels and data

For features which are conditionally independent

Conditional Independence: $a \perp b \leftrightarrow p(a,b|c) = p(a|c) * p(b|c)$

$$p(\text{label} \mid \text{data} = \{\text{feature}_1 \dots \text{feature}_k\}) = p(\text{label}) * \prod p(feature_i \mid label)$$

Compare $p(label|data)$ vs $p(\neg label|data)$

Logistic Regression:

Sigmoid function: $\sigma(t) = \frac{1}{1+e^{-t}}$

We can use the sigmoid function to get a probability from the linear regressor

Fitting:

$$p(y|X) = \sigma(X\Theta), \qquad L(y|X) = \prod \delta(y_i = 1)\sigma(X_i\Theta) * \prod \delta(y = 0)(1 - \sigma(X_i\Theta))$$

Take logarithm to convert to sum of logs, then use gradient ascent to optimize for error

Log-likelihood:

$$l_\theta(y|X) = \sum_i -\log(1 + e^{-X_i \cdot \theta}) + \sum_{y_i=0} -X_i \cdot \theta - \lambda\|\theta\|_2^2$$

Derivative:

$$\frac{\partial l}{\partial \theta_k} = \sum_i X_{ik}(1 - \sigma(X_i \cdot \theta)) + \sum_{y_i=0} -X_{ik} - 2\lambda\theta_k$$

Support Vector Machines:

Classifier Metrics,

Thursday, October 12, 2023 11:10 AM

True Positives: positive label , positive prediction
False Positives: negative label, positive prediction
True Negative: negative label, negative prediction
False Negative: positive label, negative prediction

True Positive Rate: TP / (TP + FN)
True Negative Rate: TN / (FP + TN)

Balanced Error Rate: (TPR + TNR) / 2

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

"fraction of retrieved documents that are relevant"

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

"fraction of relevant documents that were retrieved"

Scores: each classifier can score the confidence for each prediction

- Precision@k: precision only considering the top k confidence predictions

Recommenders

Tuesday, October 17, 2023 11:29 AM

Terms:
- I_u: set of items associated with user u
- U_i: set of users associated with item I

Matrix: $R_{u,i}$
- $I_u =$

Euclidean Distance:

$$|i,j| = |U_i \backslash U_j| + |U_j \backslash U_j| \text{ where } a \backslash b \text{ means a and not b}$$

Jaccard Similarity (binary similarity):

$$\text{Jaccard}(U_i, U_j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|}$$

Cosine Similarity (Real value similarity):

$$\frac{A \cdot B}{|A||B|}$$

Rating prediction: rating for item I is weighted combination of other ratings where weight is similarity

$$r(u,i) = \frac{1}{Z}\sum_{j \in I_u \backslash \{i\}} r_{u,j} * sim(i,j) \text{ where } Z = \sum_{j \in I_u \{i\}} sim(i,j)$$

Latent Models

Thursday, October 19, 2023 11:13 AM

Simple model:

- Constant
- How much does the user tend to rate things above average
- Does the item dent to receive higher ratings than others

$r(u,i) = \alpha + \beta_i + \beta_u$

Adding dimensionality: add features describing the item and user

$R = ratings$

Singular value decomposition: $R = eig(RR^T) * \sqrt{eig(RR^T)} * eig(R^T R)$

Can be written $r(u,u) = \alpha + \beta_u + \beta_i + \gamma_u + \gamma_i$

Logistic Regression:

- Perform regression and fill missing values with 0
- Problems:
  - Data very imbalanced
  - Usually can't deal with all zeros
  - Negatives are not really negative (zeros will negatively impact rating but really means no interaction)

Instance reweighting: try to figure out which negative or positives are important

Example: $\text{argmin}(\gamma) \sum c(u,i)\left(p_{u,i} - \gamma_u * \gamma_i\right)^2 + \lambda\Omega(\gamma)$

Where c is the weighting on the positive or negatives

Optimize relative scores: Bayesian Personalized Ranking

- Predict if negative items are liked less than positive items
- $p\left(R_i > R_j\right) = \sigma(\gamma_u * \gamma_i - \gamma_u * \gamma_j)$

Evaluating Recommender Systems

Thursday, October 26, 2023 11:10 AM

Problems with MSE:

- Applies the same penalty to wrong guesses
- Should some guesses be worse than others?
- Emphasize high rated things because they are recommended to users
- Assumes errors are normally distributed, what if they are bimodal?
- Most popular items will dominate MSE, less popular items will not get fair consideration

Precision and Recall @K

$$P@K(u) = \frac{|\{i \in I_u | rank_u(i) \leq K\}|}{K}$$

$$R@K(U) = \frac{1}{|U|}\sum \frac{|\{i \in I_u | rank_u(i) \leq K\}|}{|I_u|}$$

Area Under ROC Curve: does ranker tend to give positive items higher ranks than negative items

$$AUC(u) = \frac{1}{|I|}\sum_{i \in I_u}\sum_{j \notin I_u} \delta(rank(i) < rank(j))$$

$$AUC(U) = \frac{1}{|U|}\sum AUC(u)$$

Rewards algorithm for ordering items relatively, not necessarily getting the rating accurately

Mean Reciprocal Rank

$MRR(U) = \frac{1}{|U|}\sum \frac{1}{rank_u(i)}$ withholding the rankings for u foreach u

User Free Recommenders

Thursday, October 26, 2023 11:49 AM

What if we don't have access to the user's data?

- Instead of inputting the user's identity, input the user's history

Sparse Linear Methods

$$f(u,i) = \sum R_{u,j} W_{i,j}$$

Factored Item Similarity Models

$$f(u,i) = \alpha + \beta_u + \beta_i + \frac{1}{|I_u\{i\}|} \sum_{j \in I_u \setminus \{i\}} \gamma_j' \cdot \gamma_i$$

Deep Learning, Autoencoders

Tuesday, October 31, 2023 11:04 AM

Idea: want to expose non-linear relationships among features, generalize the relationship between user and items

$f(u,i) = \alpha + \beta_u + \beta_i + f(\gamma_u, \gamma_i)$

Neural Collaborative Filtering

- Use NN to learn the relationship between γ_u and γ_i

Autoencoders

- Learn a low dimensional representation of the input vectors
- Model is trained to encode these vectors
- At test time, find un-consumed items that have the highest score according to the decoder

CNNs

RNNs

Extending Latent Models

Tuesday, October 31, 2023 11:29 AM

1) Features about users and items
   $f(u,i) = \alpha + \beta_u + \beta_i + (\gamma_u + \sum \rho_a) * \gamma_i$

   Where ρ_a represents user features (age, location, etc.)

2) Implicit feedback, describe user actions through vector
   $f(u,i) = \alpha + \beta_u + \beta_i + \left(\gamma_u + \frac{1}{|\rho_a|} \sum \rho_a\right) * \gamma_i$

3) Temporal dynamics

Processing Text Data

Thursday, November 2, 2023 11:56 AM

Bag of Words Model: Fixed dimension representation of text

- Count how many times each word appears in the text
- Ignores syntax entirely
- Can remove capitalization & punctuation
- Stemming: Merge word inflections
- Discard extremely rare words or only consider the top N words

N-grams: store combinations of words to keep some grammar and meaning structure

- n-gram: sequence of n words