Oct 20th

Recommender Systems

(user, user) similarity measure

(item, item) similarity measure

PROS
no training
intuitive

CONS
users rate differently [BIAS]
rating change over time

DISCOVER BEST FEATURES IN AUTOMATED WAY

CONTENT BASED - assume you have feat. for movies - want to learn feats for users

COLLABORATIVE FILTERING

Content BASED

feature to movie similarity.

Note: if we had user-to-feature × feature-to-movie = user to movie = r_{ij}

$$p^{(j)} = \arg\min_{p} \frac{1}{\|M^{(j)}\|} \sum_{i \in M_j} (p^T Q^{(i)} - r_{ij})^2 + \underbrace{\lambda \|p\|^2}_{\text{regularization} \to \text{penalty on the size of the param } p}$$

COLLABORATIVE FILTERING.

can we learn P, Q $R = PQ$ *CANT USE SVD SINCE ITS SPARSE

$$\min_{p,q} \sum_{i,j \in R} (r_{ij} - p_i^T q_j)^2 + \lambda (\|p\|_F^2 + \|q\|_F^2)$$

1. Start w/ random Q.
2. Get P
3. improve Q
4. Repeat 2/3

Like em Algo for GMM

LINEAR REGRESSION

given n samples predict how y changes as a function of x [find $y = h(x)$]

Check $h(x_i)$ to y_i to get a metric of accuracy.

$$\mathcal{L}(h) = \sum_i d(h(x_i); y)$$ COST/LOSS FUNC.

if it went through every point we would overfit and do poorly on unseen data.

CONSTRAIN THE SPACE OF THE FUNC WE LOOK FOR

GOAL: find h that maximizes probability of having observed y.

MINIMIZE $\mathcal{L}(h) = \sum_i d(h(x_i), y_i)$

MAXIMIZE $\mathcal{L}(h) = P(y|h)$

MAKE ASSUMPTIONS:

assume linear func plus noise generated data $\vec{y} = h_\beta(X) + \vec{\epsilon}$ Note: h is linear in β.

assume it is not deterministic [SO THERE IS NOISE] $\vec{\epsilon} \sim N(0, \sigma^2)$

LEAST SQUARES.

$$\beta_{LS} = \arg\min_{\beta} \sum_i d(h_\beta(x_i), y_i) = \arg\min_{\beta} \|\vec{y} - \beta X\|_2^2 = B_{LS}$$

MAXIMUM LIKELIHOOD

since $\epsilon \sim N(0, \sigma^2)$ and $y = X\beta + \epsilon$ then $y \sim N(X\beta, \sigma^2)$

$$\beta_{LS} = (X^T X)^{-1} X^T y$$

β_{LS} UNBIASED ESTIMATOR OF TRUE β. THAT IS $E[\beta_{LS}] = \beta$