

推荐系统

example: user 看过一些 movie 并且 rating 过, recommender

System 的目标是 predict 那些 user 未看的 movie 的 rating

并将 rating 最高的那些推荐给 user.

基于内容的推荐

前提: 已知各个 movie 的 feature 值, 那些目标就是求出每个用户的 params, 就是一个 Linear Regression 的 model

然后就可以对一个新的 $x^{(i)}$ predict 其值

$$\text{cost func: } \min_{\theta^{(1)} \dots \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_m} \sum_{i: r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_m} \sum_{i=1}^n \theta_{i,j}^2$$

然后求 Gradient, 使用 optimizer 来进行优化。

协同过滤

前提: 我们已知各个用户的 params θ , $x^{(i)}$ 不知道

需要去优化 $x^{(i)}$ 向量的值

$$\text{cost func: } \min_{x^{(1)} \dots x^{(n_m)}} \frac{1}{2} \sum_{j=1}^{n_m} \sum_{i: r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n x_{i,k}^2$$



再使用 gradient 来优化其最小值。

collaborative filtering:

将之间的两个 cost function 整合起来，然后优化。

NOTE: $X^{(i)}$ 与 $\Theta^{(j)}$ 均是度量，随机初始化 X 与 Θ

推荐: $X^{(i)}$ 是用户 loves 的, $X^{(j)}$ 要满足

$\|X^{(i)} - X^{(j)}\|$ 最小, 那可以推荐 $X^{(j)}$

Low Rank Matrix Factorization: 低秩矩阵分解

和协同过滤算法是同一个意思

均值归一化:

对每个 $X^{(i)}$ 求均值 (已知评分的) 将每个评分减去 mean

然后用这个新的 X 去训练。

预测时用 $X (\Theta^{(j)})^T X^{(i)}$ 预测时加上 $\mu^{(i)}$

