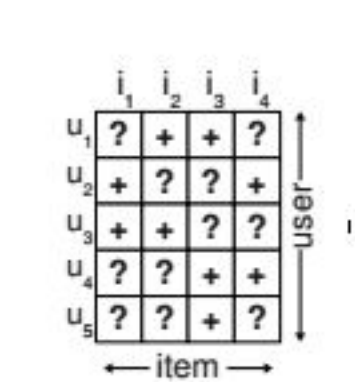


关于矩阵分解

矩阵分解活跃在推荐领域，基于SVD的推荐系统也是矩阵分解的一种。给定一个用户评分表，通常这个是个很大的矩阵，m行n列，m代表用户的个数，n代表项目的个数。并且这个矩阵在实际情况中是非常稀疏的，用户只能评价少部分的项目，因而矩阵中会存在很多?,用户并没有对对应的项目打分或者是评价过，所以我们很难了解用户对相应项目的偏好情况。



而我们推荐矩阵分解就是希望能通过用户已有的评分来预测用户对未打分或者评价项目的评价情况，而通过矩阵分解则能挖掘用户的潜在因子和项目的潜在因子，来估计缺失值。

$$Y_{m \times n} = U_{m \times k} * V_{n \times k}^T$$

矩阵Um,k的行向量表示用户u的k维的潜在因子，表达用户的内部特性，矩阵Vn,k的行向量表示项目i的k维的潜在因子，表示项目的内部特性。利用矩阵U和V可以估计用户u对项目i的评分为：

$$y_{u,i} = u_u * v_i^T$$

对于任意矩阵，一定存在矩阵U和V使得Y=U*VT么？

但是一般情况下不一定能非常完美的进行矩阵分解，所以我们可以利用最小化偏差来不断训练参数，这里的参数theta = (U,V);

$$e_{u,i} = y_{u,i} - \hat{y}_{u,i}$$

$\hat{y}_{u,i}$ 为利用矩阵U、V矩阵预测用户u对项目i的打分情况。

Basic MF:

Basic MF的目标函数为误差平方和作为优化目标:

$$SSE = \frac{1}{2} \sum_{u,i} e_{u,i}^2 = \frac{1}{2} \sum_{u,i} (y_{u,i} - \hat{y}_{u,i})^2$$

$$\hat{y}_{u,i} = \langle u_u, v_i \rangle = u_u * v_i^T = \sum_k u_{u,k} * v_{i,k}$$

我们可以一般采用梯度下降来优化参数theta,为了减少计算的复杂度，我们这里采用的是随机梯度下降。

$$\frac{\partial SSE}{\partial u_{u,k}} = -e_{u,i} * v_{k,i}$$

$$\frac{\partial SSE}{\partial v_{k,i}} = -e_{u,i} * u_{u,k}$$

采用负梯度进行梯度下降

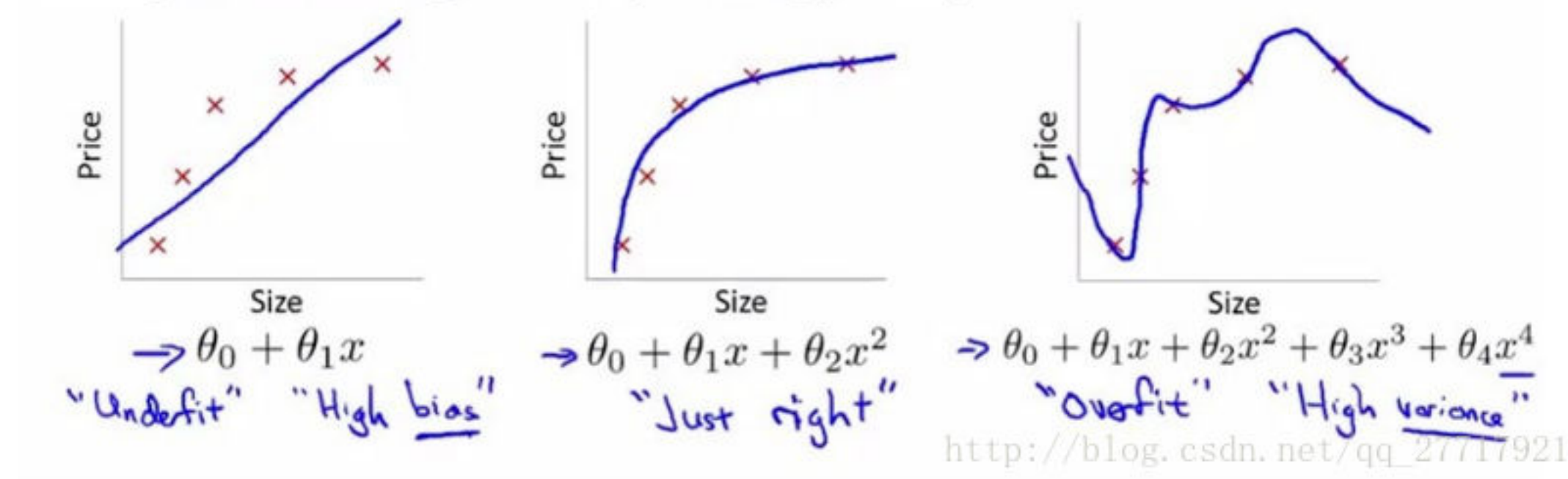
$$u_{u,k} = u_{u,k} - \eta * (-e_{u,i} * v_{k,i}) = u_{u,k} + \eta * e_{u,i} * v_{k,i}$$

$$v_{k,i} = v_{k,i} - \eta(-e_{u,i} * u_{u,k}) = v_{u,k} + \eta * e_{u,i} * u_{u,k}$$

迭代更新U、V矩阵，直到趋于收敛。

Regularized MF

Example: Linear regression (housing prices)



如果待分解的矩阵Y非常的稀疏，我们在不断减少平方误差的过程中就很可能出现的过拟合的现象，为了使训练出来的U、V矩阵更好的拟合现有的数据而导致在缺失上的数据效果不好就可能造成过拟合现象。因此我们应该在一定程度上限制参数，因为U、V上的每一个元素都是待更新的参数，我们要在优化目标上加入正则项，也就是惩罚项。

$$SSE = \frac{1}{2} \sum_{u,i} e_{u,i}^2 + \frac{1}{2} \lambda \sum_u |u_u|^2 + \frac{1}{2} \lambda \sum_i |v_i|^2$$

梯度为:

$$\frac{\partial SSE}{\partial u_{u,k}} = -e_{u,i} * v_{k,i} + \lambda u_{u,k}$$

$$\frac{\partial SSE}{\partial v_{k,i}} = -e_{u,i} * u_{u,k} + \lambda * v_{k,i}$$

梯度更新式为:

$$u_{u,k} = u_{u,k} - \eta(-e_{u,i} * v_{k,i}) = u_{u,k} + \eta * e_{u,i} * v_{k,i}$$

$$v_{k,i} = v_{k,i} - \eta * (-e_{u,i} * u_{u,k} + \lambda * v_{k,i})$$

迭代更新U、V矩阵，直到趋于收敛。

Regularized MF + 偏置项

$$y_{u,i} = u_u * v_i^T$$

我们可能无法表达用户u和项目i的关系，可能除了受用户和项目的潜在因素影响，可能还会受到其他因素的影响，所以我们需要用偏置项来表示这些其他因素对预测的影响。

$$\hat{y}_{u,i} = u_u * v_i^T + b_u + b_i + \mu$$

误差平方和以及加上损失项的目标函数

$$SSE = \frac{1}{2} \sum_{u,i} e_{u,i}^2 + \frac{1}{2} \lambda \sum_u |u_u|^2 + \frac{1}{2} \lambda \sum_i |v_i|^2 + \frac{1}{2} \lambda \sum_u |b_u|^2 + \frac{1}{2} \lambda \sum_i |b_i|^2$$

这时我们的theta 除了包含U,V矩阵还包含bu、bi向量，U、V的更新式和Regularized MF 保持一致，而bu,bi的更新式为:

$$b_u = b_u + \eta(e_{u,i} - \lambda b_u)$$

$$b_i = b_i + \eta(e_{u,i} - \lambda b_i)$$

不断迭代更新，直到参数收敛，我们就可以利用 $\hat{y}_{u,i}$ 来估计用户u对项目i的打分情况。