PSOCC

一、 算法思想

1、 构建 user-item 矩阵；

2、 SVD 得到 laten Factor，来作为 item 的特征；

3、 粒子群优化 PSOCC（PSO with Constriction Coefficient）算法学习 rank model；

4、 用 model 在 test 数据集上测试，得到 TopN 的度量结果

二、 算法说明

1、 main.m

程序入口文件，初始化 PSOCC 参数，训练、测试

2、 psocc.m

PSOCC 算法，其实就是在速度上有所体现：

$$\chi = \frac{2k}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, where \varphi = c_1 + c_2, \varphi > 4$$

$$v_{id}(t) = \chi[v_{id}(t-1) + c_1\varphi_1(p_{id} - x_{id}(t-1)) + c_2\varphi_2(p_{gd} - x_{id}(t-1))]$$
$$x_{id}(t) = x_{id}(t-1) + v_{id}(t)$$

```
function [theta, fit] = psocc(opinion, feature, label)
```

输入：

opinion：参数

feature：item 的特征，一行一个样本

label：1 和-1,1 代表用户 action 过（action：听歌）

输出：

theta：结果模型参数，这里用的线性模型，参考 Logistic Regression。

fit：最优的 fitness 值

3、 mapfitness.m

MAP 作为 fitness function。每个用户的 AP 最大，最好的 MAP 也是最大的。

```
function fit = mapfitness(theta, feature, label)
```

输入：

theta：模型参数，也就是每个粒子的 position 值

feature：item 特征

label：同上。

4、 psopredict.m

预测函数，sigmoid function。1/(1+exp(-v))

5、 logistic_regression

除了用 PSOCC 选练模型，也可以用 LR、SVM 等，所以也写了个 LR，可以对比结果用。

其实我发现 LR 模型效率更高、效果更好。这也是 LR 模型在工业界应用如此广泛的原因。

6、 test_psocc.m

PSOCC 的测试用例，可以测试用

三、 数据集

1、 Last.fm-1K user 的数据集，为了试验方便，选择听歌最多的 500 个 user，选了被听过最多次数的 10000 个专辑；每个用户根据时间分出其训练集和测试集。与论文中描述的一样。

2、 /data/ train_set.dat，/data/ test_set.dat，
训练集，第一列用户 id，第二列专辑 id，第三列听的次数；

3、 /data/m_train.dat
user-item 矩阵，svd 就是分解这个矩阵，当然矩阵中的元素只有训练集中的用户听专辑的次数，不能包含测试集中的。

4、 /data/result.txt
第一行是 PSOCC 的 TopN
第二行是 LR 的 TopN

四、 使用说明

运行 main.m

五、 并行

1、 粒子群优化效率比较低，matlab 中可以用 parfor 并行。

六、 联系

Pagelee.sd@gmail.com