

京东个性化推荐实践

王志勇，杨骥，司建敏

众所周知，在用户对自己需求相对明确时，通过关键词搜索能很快的找到自己需要的商品。但是在很多情况下，用户并不明确自己需要的商品，或者他们需求的商品很难用关键词来描述，同时用户希望网站的反馈更符合其个人的兴趣和喜好，在这种场景下，个性化推荐系统会比搜索引擎更能满足用户对商品发现的需求。

正如 Amazon、Netflix 等国际知名互联网公司一样，个性化推荐系统在京东的业务发展中起到了非常重要的作用。为了让各位读者能够深入了解京东个性化推荐的全貌，本文介绍了个性化推荐的系统架构、推荐算法框架，以及离线数据挖掘、在线排序等核心模块的实现细节。

一. 个性化推荐系统架构

传统推荐系统的实现主要依靠商品之间的“共现”关系（例如频繁模式挖掘[9]等）进行推荐，从效果上看所有用户得到的推荐结果都是相同的；个性化推荐系统和前者有很大不同——从商品召回、数据索引到最终结果的排序，很多关键环节都考虑了用户的个性化信息，最终每个用户看到的推荐结果都是不同的，并且反映了用户的购物兴趣和购物习惯[3][4]。

个性化推荐系统的目的是在合适的场景，合适的时间，通过合适的渠道把合适的内容推荐给合适的用户。通常线下的导购或者销售只销售有限的几类商品，一般也不会进行客户分析，因此很难满足到店用户的个性化需求。京东作为国内最大的 B2C 购物平台，有足够的计算资源和技术能力对海量数据进行分析，为每一个用户建立购物行为和消费心理模型，并以此为基础构建个性化推荐系统，最大程度的提高用户的购物体验。

京东的个性化推荐不但是一个业务驱动的系统，而且也是一个技术和知识驱动的系统，为了实现业务需求，我们需要对海量的用户和商品进行分析和建模，实质上就是数据挖掘（Data Mining）和机器学习（Machine Learning）技术在海量数据上的应用。同时，个性化推荐系统还要做到高并发实时计算，并能支持算法和模型的快速迭代和效果验证。此外，个性化推荐系统还要综合平衡系统和业务需求，以及用户对商品多样性、惊喜度和可解释性的需要。为了兼顾各方面的需求，我们建立起了一个完备的个性化推荐系统框架，下面就对其进行详细介绍。

1.1 个性化推荐基础架构

如图 1 所示，个性化推荐架构核心模块主要由网关，分流器，推荐器，召回引擎以及外围的商品画像，用户画像服务和配置管理模块构成。网关(web gateway)主要负责参数解析和接口服务转换；分流器(recroot)主要负责和配置管理模块协同，支持推荐场景和实验分流，同时支持不同的缓存策略；推荐器(recommender)是推荐在线算法逻辑的核心，主要负责对具体推荐场景下的召回对象进行排序(Ranking)；召回引擎(recall engine)主要负责召回离线模型挖掘出的用户感兴趣的商品，活动，店铺等对象，核心模块是搜索引擎和 KV 引擎(类 Redis 和 HBase)。

用户在京东购物过程中，首先页面会请求个性化推荐服务，并将用户购物相关场景数据

传递给网关；网关接下来将请求数据转化后传递给分流器；分流器会根据用户所在的具体场景、对应场景下的分流策略和相关配置参数选择合适的推荐器；被选择的推荐器将综合用户画像和商品画像的相关数据，对召回引擎召回的用户感兴趣对象进行排序(Ranking)。其中，用户画像服务能在秒级时间内感知用户的实时行为和兴趣点，是我们个性化推荐系统中的亮点。

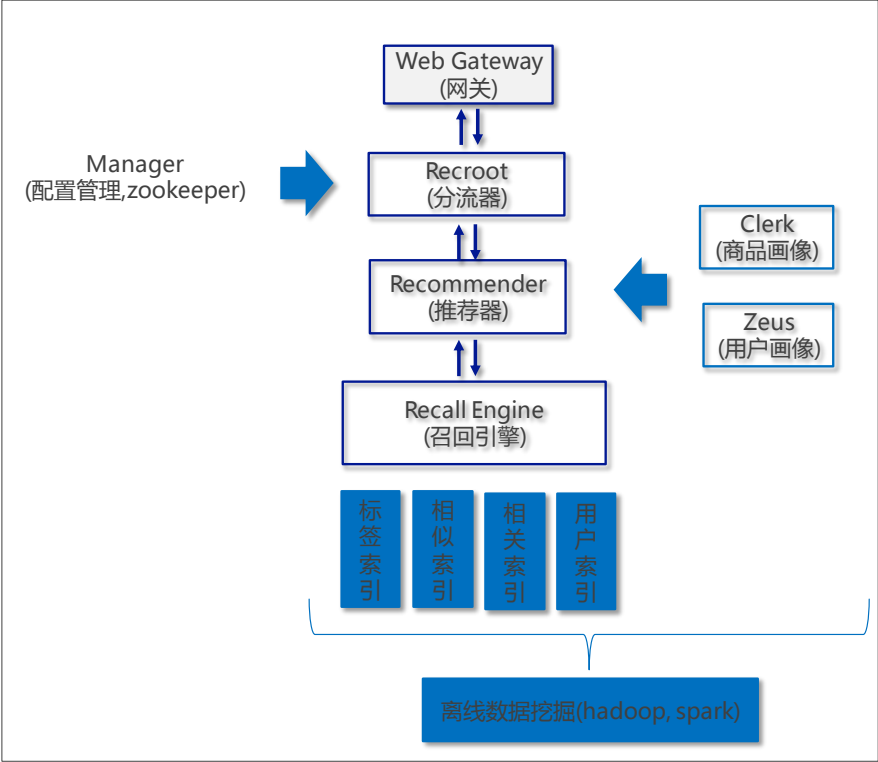


图 1.个性化推荐基础架构

1.2 个性化推荐 A/B 测试机制

个性化推荐不是一个静态的系统，随着用户数的增加和用户购物习惯的变化，我们的个性化推荐系统也需要不断演进和优化。作为优秀的推荐产品，都需要走持续优化之路。为此，京东个性化推荐架构也提供了一个支持算法快速迭代的 A/B 测试机制——首先，系统支持业务实验流量分配的准确性，实时性以及分配方式的多样性；其次，如图 2 左侧所示系统提供离线实验效果的 side by side 的评测和调试；此外，如图 2 右侧所示在线实验效果多指标维度的数据报表分析，给算法工程师提供方便的算法评测环境。

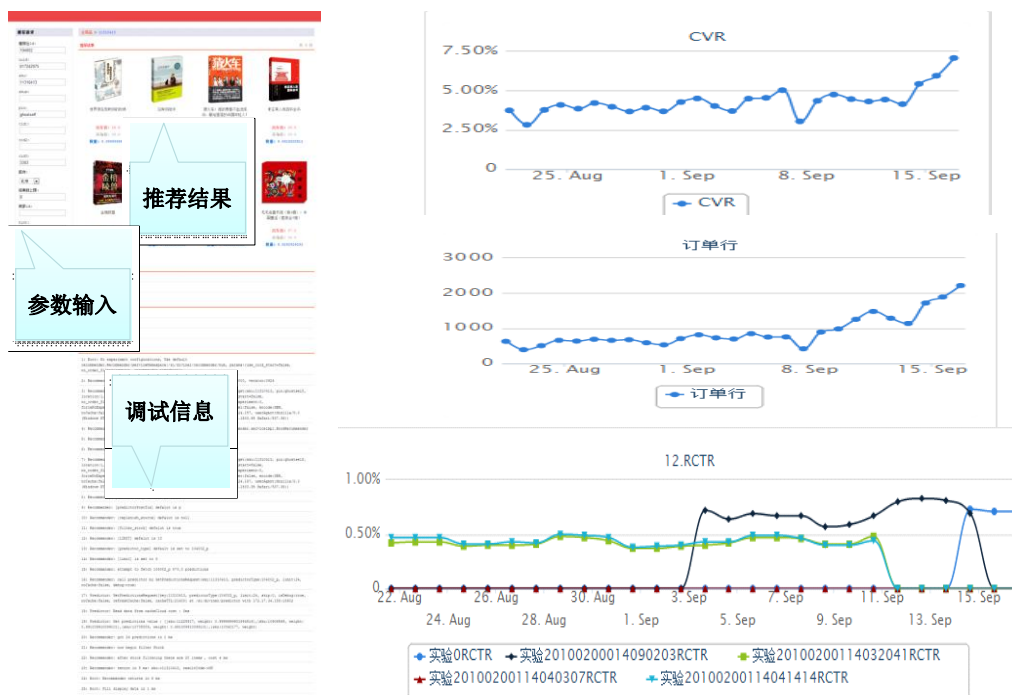


图 2.A/B 测试机制

上述基础框架不但支持高并发访问，同时在海量数据存储和在线高速访问等核心模块上也有符合自身业务特点的设计。

二. 个性化推荐算法框架

本章将介绍个性化推荐算法框架。如图 3 所示，个性化推荐算法框架主要由离线数据挖掘，和在线排序两个核心功能模块组成。由于每个模块所涉及的核心技术比较繁多并且每一个技术所涉及的理论和实践经验就可以单独成文，因此有篇幅所限本文后续章节将着重简单介绍京东个性化推荐实践中的重点技术。

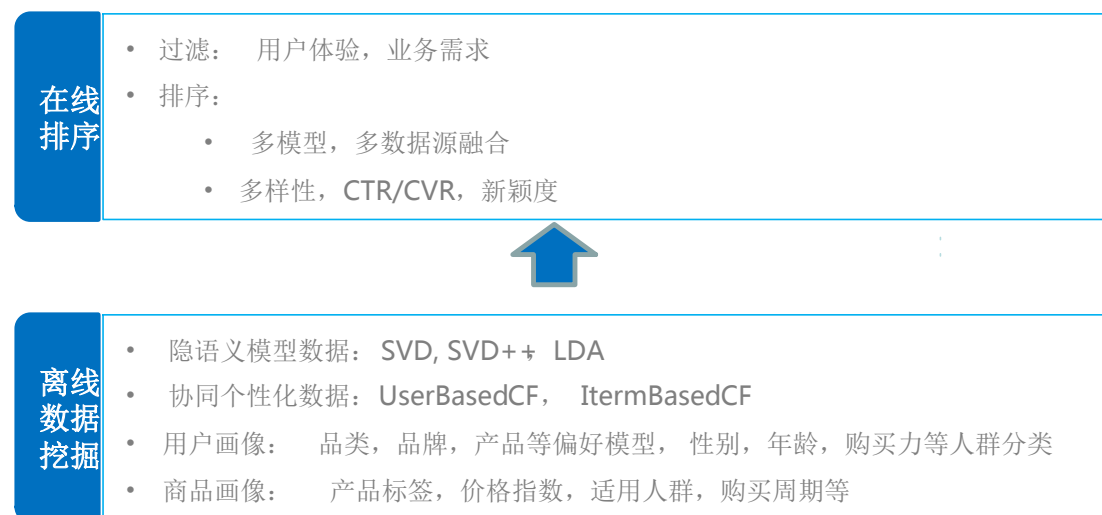


图 3.个性化推荐算法框架

2.1 推荐离线数据挖掘

民间有句俗语“巧妇难为无米之炊”，如果将这句话类比于个性化推荐系统，那么召回的“商品”就是我们的“米”，而且召回商品的“质量”（例如多样性、价格范围、好评度等）

也决定了用户体验的满意程度。

在分析用户的行为之前，我们会对京东用户的海量行为日志进行预处理，目的是将充满大量噪声、缺失值以及冗余的原始数据转换成规范的、结构化的高质量数据，满足接下来的离线数据挖掘任务的需要。数据预处理主要包括“数据清洗”、“数据转换和集成”、“用户行为标注”、“数据归一”以及“恶意用户识别”等工作：

- 1) 数据清洗：目的是对日志文件中的缺失值、错误数据以及前后记录格式不一致的数据进行处理。
- 2) 数据转换和集成：数据转换是指对数据进行平滑、特征计算、数据统计等；数据集成是指将多个数据源的数据汇总到一起。
- 3) 用户行为标注：行为标注在数据预处理环节是最重要的一环，目的是甄别出用户每一个浏览/加入购物车/关注等动作对商品的感兴趣程度，而用户的兴趣挖掘是个性化推荐系统的基石。我们会按照时间尺度的大小分别对用户的进行兴趣识别——在较大的时间尺度上（比如以“天”为单位），会看用户这次浏览行为是否为“已购买商品的查单行为（查看已购买商品所在分类的其他商品，或者是已购买商品本身）”，或者是到了购买周期，需要重新购买该品类的商品等情形；较小时间尺度上，比如用户 session 级别，我们会研究用户的浏览路径，确切说在商品页是否触发了某些“事件”，比如用户是否查看了商品的评论、规格参数、售后保障等，或者是否触发了页面底端的某些推荐位等；另外在用户的浏览路径中也会突然出现多个不同品类的商品，这些行为是用户的真实兴趣或者是被某些其他事件的吸引，也会进行标注。
- 4) 数据归一：数据中所有字段或者属性之间必须有一个统一的计量单位或者相同的范围。归一化的目的是在进一步处理之前给所有的字段/属性一个相同的权重，这在数据挖掘或者机器学习任务中非常重要。
- 5) 恶意用户识别：目的是对恶意用户下单/浏览行为进行降权或者去除，使数据的分布能准确表达京东用户的兴趣或者行为模式。

在数据预处理的基础之上就可进行离线数据挖掘了，目前我们的离线数据挖掘分为四个方面：1) 隐语义模型数据计算；2) 协同个性化数据计算；3) 用户画像；4) 商品画像。下面分别对这几部分进行介绍：

- 1) 协同个性化数据：基于协同过滤（CF）的方法不需要计算用户或者商品的多维特征，只要提取出用户的历史行为（浏览/关注/下单等），构造出用户-商品的交互矩阵，然后利用用户、商品之间的交互作用给用户带来个性化推荐[5][6]。我们使用两种协同个性化数据，即基于用户的协同过滤（UserBasedCF）和基于商品的协同过滤（ItemBasedCF）。

UserBasedCF 认为两个用户如果在历史行为上对一些商品有相似的偏好，那么他们现在偏好仍然相似，对这两个用户进行协同推荐是合理的。ItemBasedCF 认为某个用户历史上喜欢过某些产品，那么这个用户现在仍对这些产品的相似产品感兴趣。

- 2) 隐语义模型：主要思想是通过隐含特征来联系用户和商品。实践中我们会使用预处理好的数据对用户的每次行为进行一个打分。在京东的个性化推荐系统中，我们主要采用了 SVD 和 SVD++[7]两种隐语义模型：

SVD 模型，即奇异值分解，主要思想是给定 m 个用户和 n 个商品，则评分矩阵 $R \in \mathbb{R}^{m \times n}$ ，因为用户不可能浏览所有的商品，所以整个评分矩阵是非常稀疏的，所以需要先将 R 进行补全，得到矩阵 $R' \in \mathbb{R}^{m \times n}$ ，然后对矩阵进行奇异值分解得到[1]

$$R' = U^T S V$$

其中 $U \in \mathbb{R}^{k \times m}$ 和 $V \in \mathbb{R}^{k \times n}$ 分别代表用户和商品的隐因子模型， $S \in \mathbb{R}^{k \times k}$ 是 R' 的 k 个特征值。这样用户 i 对商品 j 的评分预测值就为

$$\widehat{rating}(i, j) = U_i^T S V_j$$

通常会利用已有的相似商品数据对矩阵进行填充和补全，然后再进行矩阵分解。

SVD++模型是 Yehuda Koren 提出的[2]，模型为：

$$r_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j \right)$$

其中 μ 是所有评分的均值， b_u 是用户 u 评分的偏差， b_i 是商品 i 得到评分的偏差， q_i 是商品的隐因子， p_u 是用户的隐因子， $N(u)$ 是用户所有有过行为的商品集合， y_j 是 $N(u)$ 中商品的隐因子。利用随机梯度下降（SGD）法对参数进行迭代求解。

除了上面介绍的 SVD 和 SVD++，我们还使用了同样属于因子模型领域的 LDA 方法对商品进行了主题聚合，从效果上来看，在图书品类下的商品聚合效果很好。

3) 用户画像：目的是在分析行为数据的基础上，精准分析出用户的长期购物兴趣和购物习惯，从而抽象出用户的信息全貌。用通俗的话来讲，用户画像就是给用户打上一个或者多个标签，比如“年龄”、“职业”、“地域”、“性别”等等。

结合京东的业务特点，目前个性化推荐系统使用的用户画像模型能够准确刻画出用户的品牌、品类以及产品偏好；能够描述用户的性别、年龄和购买力等群体分类信息；提供用户是否有小孩以及小孩的性别等家庭成员信息。

目前我们已经将用户画像技术应用于京东个性化推荐系统当中，同时更为精细的用户标签挖掘工作也在持续进行当中。

4) 商品画像：用来刻画商品本身与用户相关的属性，不同于常见的统计指标刻画（比如 PV，UV，销量排名等）。我们使用的商品画像主要有产品标签、价格指数、适用人群和购买周期等。

2.2 推荐在线排序

2.2.1 模型融合

个性化推荐系统中的在线排序，解决的是模型融合问题。因为单一模型效果有限，所以在实践中我们会对多个模型数据的结果进行融合，如在线相似、在线相关、离线相似、离线相关、协同过滤、SVD 和 SVD++等，这样可以大大提高召回商品的多样性和新颖性。但是召回商品的数量较多时，究竟哪些才是用户最感兴趣的以及如何对这些商品进行排序才能让用户有更加满意的体验，这就是模型融合需要解决的问题。

投票法是多模型融合的比较常用的方法，融合时直接将多个模型的商品权重进行相加，得到商品最后的权重。投票法不足之处在于其把每个模型同等对待，更合理的方式应该将“好”的模型的权重加大，“差”的模型权重减小。为了研究模型质量，需要对模型的效率进行分析。例如从点击率与转化率的角度进行分析，从图 4 和图 5 中可以看出模型间的差异比较大，接下来我们就可以根据效率分析的结果来调节每个模型的权重。

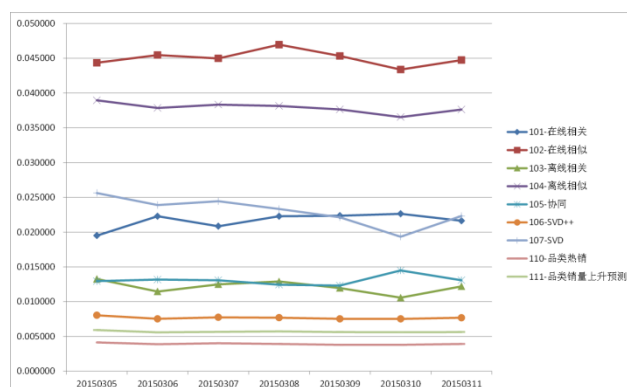


图 4 模型效率-CTR

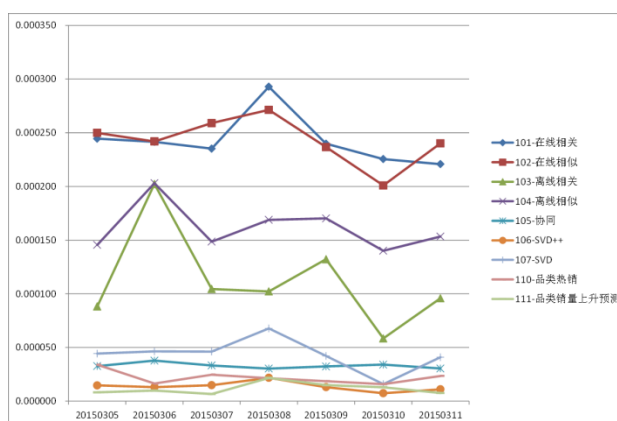


图 5 模型效率-CVR

2.2.2 Learning to Rank 实践

虽然我们可以通过模型效率的分析调整模型的权重，但这种方式毕竟效果有限。此外，推荐结果的排序还与用户的实时兴趣、用户画像中的标签属性、商品自身的特性有关，如果用人工的方式考虑各个维度，工作量十分巨大且排序效果无法保证。

我们排序时使用的特征 $\{f_1, f_2, f_3, \dots, f_n\}$ ，其对应的权重为 $\{w_1, w_2, w_3, \dots, w_n\}$ ，然后根据这些特征以及特征的权重对商品进行打分。我们对商品 I 的打分可用公式（1）来表示。

$$score(I, w) = w_1 * f_{I1} + w_2 * f_{I2} + w_3 * f_{I3} + \dots + w_n * f_{In} \quad (1)$$

通过用户的历史行为数据，可以构建出样本的排序为 $score(I_1, w) > score(I_2, w) > \dots > score(I_k, w)$ ，这样问题就转化为权重 $\{w_1, w_2, w_3, \dots, w_n\}$ 的机器学习问题，即排序学习问题[8][10]。

京东个性化推荐系统采用的是基于序对的排序学习（pairwise）方式，即考虑两个样本之间的偏序关系，若 $I_1 > I_2$ ，则认为 $(I_1 > I_2)$ 为正样本，否则为负样本，这样将其转化为二分类的问题，有两两之间的偏序关系可以得到整体的排序。

整个排序的流程如图 6 所示，首先是要通过用户历史的行为数据将当时的历史场景进行恢复，然后构建训练样本、对模型进行训练，在离线评测达到的最优时再进行线上的实验。下面将简要的介绍每个流程。

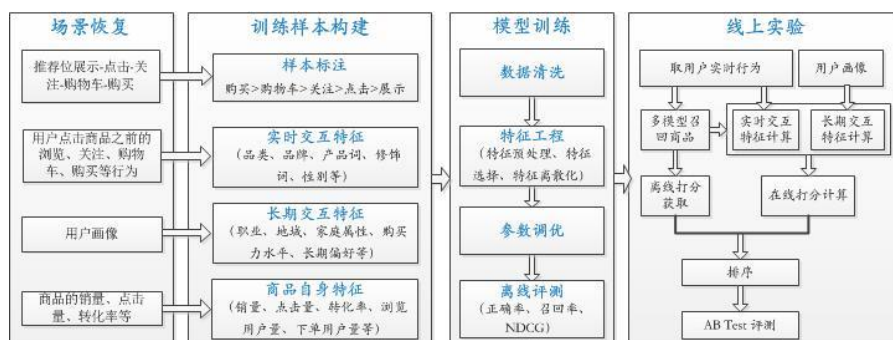


图 6 排序学习流程

场景恢复: 这里所谓的场景是指推荐商品展示给用户那个时间点之前的场景以及用户对推荐商品的行为序列，包括某个时间点之前某个时间段内的所有历史行为数据、用户画像、商品的历史数据、用户点击、关注、加入购物车和购买推荐商品的行为序列。

训练样本构建：这部分包括样本的标注和样本的特征计算。通过恢复出来的用户点击、关注、加入购物车和购买推荐商品的行为序列来对样本进行标注。用户的行为包括购买、加入购物车、关注、点击和未点击等行为，这些行为表示着用户对商品的喜欢程度，通过它就可以对样本进行标注。点击的商品大于前面未点击的商品，关注的商品大于点击的商品，加入购物车的商品大于关注的商品，购买的商品大于加入购物车的商品。如图 4 所示，若展示了 4 个商品 A、B、C、D，用户点击了 B，购买了 C，则可以生成(B,A,+), (C,A,+), (C,B,+), (C,D,+) 四个样本对。标注样本之后就需要计算这些样本的特征，包括商品与用户实时行为特征、用户画像（长期行为）的交互特征和商品自身的特征。

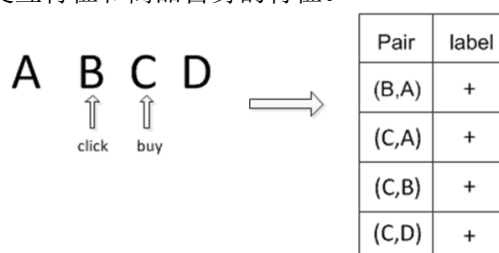


图 7 样本标注

算法简介

这里采用改进的 RankSvm 算法 RankSvmIR^[11]。传统的 Pairwise 算法有一个较大的不足之处在于样本对之间没有任何区别，即点击与未点击的样本对和购买与未购买形成的样本对之间同等对待。很明显在这里是不合理的，因为点击与未点击的样本对要多于购买形成的样本对，购买形成的样本对更能说明用户喜好的差别，应该占有更大的权重。RankSvmIR 的损失函数如公式（2）所示

$$\min_w L(w) = \sum_{i=1}^l \tau_{k(i)} q_i \left[1 - z_i \langle w, x_i^{(1)} - x_i^{(2)} \rangle \right]_+ + \lambda \|w\|^2 \quad (2)$$

其中， $k(i)$ 表示样本对 i 的类型， $\tau_{k(i)}$ 表示样本对类型 $k(i)$ 的权重， q_i 表示不同的样本对 i 对应的 query 的权重。因为在解决推荐系统的排序问题时，query 为用户的实时行为特征和用户画像特征，每个 query 形成的样本对不会特别多，通常不会造成样本对的偏置问题，这也是文档排序学习与推荐排序学习不同的地方。在这里我们只是考虑样本对之间的类型差别而不需要考虑不同 query 之间的样本对数量的差别。我们在开源软件 spark¹平台上实现了基于随机梯度下降的分布式优化求解。

模型训练：比较常见的机器学习管道（machine learning pipeline）的形式，包括样本的清洗、特征预处理、特征选择、特征离散化、参数调优和离线评测等内容。为了调整超参数，我们将数据时间拆分，80%的数据为训练样本，20%的数据作为测试样本，通过交叉验证的方法给出较优的值。

线上实现：为了验证排序学习的有效性，我们进行了 AB Test 测试，即使用相同的召回策略，一种是人工排序的方式，一种使用排序学习的模型。以“我的订单猜你喜欢”推荐位为例，线上实现的流程如图 8 所示。如图 9 和图 10 所示，从线上 RCTR 和 RCVR 指标可以看出排序学习算法比人工排序有着明显的优势。

¹ <http://spark.apache.org>

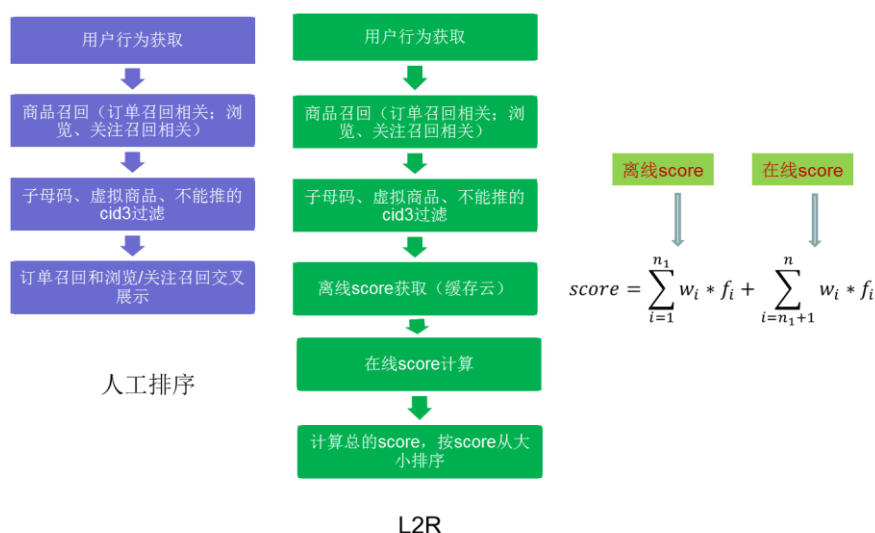


图 8.人工排序与排序学习的线上实验流程

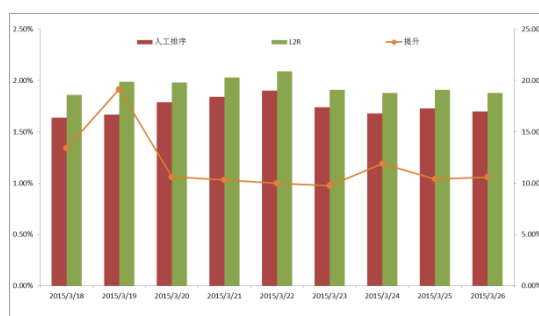


图 9. RCTR

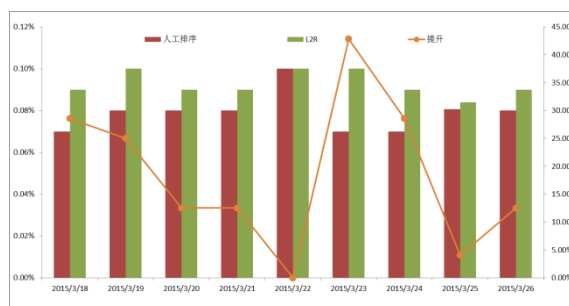


图 10. RCVR

三. 总结

本文介绍了京东个性化推荐系统的基础架构，算法框架以及实践过程的核心技术。如文中提到的，京东个性化推荐系统每天都在进行优化和演进，例如梯度提升分解机、深度神经网络等技术都会陆续在我们的系统中上线，我们相信自己的个性化推荐系统会给用户带来更多的惊喜。

参考文献

[1]. 项亮,推荐系统实践.人民邮电出版社, 2012 年 6 月

- [2] Y. Koren, Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, ACM Press, 2008, pp. 426-434
- [3] Rokach L, Shapira B, Kantor P B. Recommender systems handbook. New York: Springer, 2011.
- [4] Jannach, Dietmar, Recommender systems: an introduction. Cambridge University Press, 2010.
- [5] G. Adomavicius and A. Tuzhilin, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17 (6)
- [6] B. Sarwar, Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of World Wide Web Conference, 2001.
- [7] Bell R M, Koren Y. Lessons from the Netflix prize challenge. ACM SIGKDD Explorations Newsletter, 2007, 9(2): 75-79.
- [8] Z. Cao and T. Liu, Learning to rank: From pairwise approach to listwise approach. In Proceedings of the 24th ICML, 2007.
- [9] Tan P N, Steinbach M, Kumar V. Introduction to data mining. Boston: Pearson Addison Wesley, 2006.
- [10] C. Burges, From RankNet to LambdaRank to LambdaMART: An Overview. MSFT Technical Report, 2010.
- [11] Cao Y, Xu J, Liu T Y, et al. Adapting ranking SVM to document retrieval. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006: 186-193.