doi: 10.12068/j.issn.1005-3026.2019.01.005

基于 Adaboost 学习的 ICN 自适应缓存算法

蔡 凌[1], 汪晋宽[2], 王兴伟[3], 胡 曦[4]
(1. 东北大学秦皇岛分校 控制工程学院, 河北 秦皇岛 066004; 2. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 3. 东北大学 软件学院 辽宁 沈阳 110169; 4. 东北大学秦皇岛分校 计算中心, 河北 秦皇岛 066004)

摘 要: 针对信息中心网络(ICN)中缓存内容优化放置的问题,提出一种基于 Adaboost 学习的自适应缓存算法 ACAL. 该算法首先将提取的节点和内容数据流作为网络资源,然后利用集成学习算法 Adaboost 对数据流进行分析挖掘,利用挖掘出的状态属性与缓存匹配之间的函数映射关系对未来时间段内的节点与内容间的匹配关系进行预测,该预测结果用于指导缓存的部署. 实验结果表明,ACAL 在延时、缓存命中率和链路利用率等指标方面,与 CEE 策略、LCD 策略、prob0.5 策略和 OPP 策略相比有显著的优势.

关 键 词: 信息中心网络; 缓存网络; 缓存策略; 学习算法; Adaboost 算法
中图分类号: TP 393 **文献标志码**: A **文章编号**: 1005-3026(2019)01-0021-05

Adaptive Caching Algorithm Based on Adaboost Learning for Information Centric Networking(ICN)

CAI Ling[1], *WANG Jin-kuan*[2], *WANG Xing-wei*[3], *HU Xi*[4]
(1. School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China; 2. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 3. School of Software, Northeastern University, Shenyang 110169, China; 4. Computing Center, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China. Corresponding author: CAI Ling, E-mail: cailing9852@126.com)

Abstract: In order to optimize the cache placement in ICN(information centric networking), an ACAL(adaptive caching algorithm based on Adaboost learning) algorithm was proposed. According to the algorithm, first, the extracted data flow including node data and content data was employed as the network resources, then the ensemble learning algorithm Adaboost was used to analyze and mine the data flow, and the mapping relationship between the state attribution data and the matching relationship value was utilized to predict the matching relationship between the node and the content in next period. Finally, the matching relationship algorithm was used to guide the cache placement. The simulation experiments demonstrate that the proposed ACAL, compared with CEE, LCD, prob0.5 and OPP yields a significant performance improvement, such as delay, hit rate and average link utilization.

Key words: information centric networking(ICN); caching network; caching strategy; learning algorithm; Adaboost algorithm

根据 Cisco 可视化网络指数(visual networking index, VNI)报告的预测,2021 年互联网中视频类应用消耗的网络流量将占全网总流量的 82%[1]. 为了满足大量视频类应用对网络的需求,研究者提出了信息中心网络(information centric networking, ICN)架构,使网络中的节点可为内容提供路由及缓存双重服务.

网络化缓存是 ICN 的重要特征,缓存首先需研究的是内容放置问题. ICN 的最初方案执行的是处处缓存(cache everything everywhere, CEE)

收稿日期: 2017-05-24
基金项目: 国家自然科学基金资助项目(61501102); 河北省高等学校科学技术研究项目(QN2014327).
作者简介: 蔡 凌(1980-),女,湖南武冈人,东北大学秦皇岛分校讲师,博士; 汪晋宽(1957-),男,辽宁营口人,东北大学教授,博士生导师; 王兴伟(1968-),男,辽宁盖州人,东北大学教授,博士生导师.

策略[2]. 为了提高利用率,部分文献提出了基于节点数据的算法,探讨缓存节点的选取. 文献[3]提出的LCD方案,将内容缓存在命中节点的下游节点处. 文献[4]根据节点的度中心性、紧密中心性和介数中心性等指标来选取缓存节点. 文献[5]提出的Prob策略,定义缓存概率为 p. 文献[6]中缓存概率因子是基于缓存节点与源节点距离及缓存容量这两因素. 文献[7]认为缓存概率因子与内容流行度和替换代价有关. 针对这些文献并未考虑内容在空间分布的合理性和均衡性问题,部分学者研究了基于内容数据的算法,研究缓存内容的选择. 文献[8]将由内容热度和缓存收益组合成的缓存概率作为是否被缓存的判断依据. 文献[9-10]以内容的使用效率为判断基准. 还有部分学者研究了结合节点数据和内容数据的算法. 内容数据主要度量的是流行度,而节点数据的选取原则不尽相同,例如文献[11]主要考虑的是缓存容量这个因素;文献[12]考虑的是缓存节点与源节点之间的距离;文献[13-14]中考虑的是节点在拓扑结构中的位置;文献[15]考虑的是节点的缓存容量和路径跳数.

上述成果为基于集成学习的缓存研究提供了基础. 本文所提出的缓存机制与算法,在大量节点数据和内容数据相互感知的基础上,通过对缓存规律的学习,自适应地实现缓存匹配.

1 基于Adaboost学习的缓存

1.1 多维状态属性数据的定义

缓存内容的放置问题是一个节点与内容的偶合、匹配问题,因此,算法所需的数据需要能兼顾对节点特性和内容属性的描述. 本文从节点和内容两个维度对数据进行提取,在节点的维度上,主要描述节点的负载率;在内容的维度上,主要描述内容热度与节点的相关性.

1.1.1 节点维度

不同节点在不同时间段的访问流量不尽相同,节点维度就是通过缓存率和缓存替换率来分别描述节点在不同时期的访问流量.

定义节点缓存率为CR, $CR = \dfrac{\sum_{i=0}^{n} CCS_i}{CSS(v)}$, 其中 CCS_i 表示单位时间内在节点 v 上缓存的内容 i 的大小,$CSS(v)$ 表示节点 v 的缓存容量. 缓存率用于定义节点缓存负荷未满阶段的负载率.

定义节点缓存替换率为RR, $RR = \dfrac{\sum_{i=0}^{n'} RCS_i}{CSS(v)}$, 其中 RCS_i 表示单位时间内在节点 v 上被替换的内容 i 的大小. 替换率用于定义节点缓存容量满负荷阶段的工作状态.

1.1.2 内容维度

内容维度的定义用于从时间和空间两个层面分析其分别与内容流行度的相关性,流行度描述的是时间对内容流行程度的影响,被请求的内容权重,则是定义节点的空间位置对内容流行程度的影响.

定义流行度为 LP_{vi}, $LP_{vi} = \dfrac{IRN_{vi}}{\sum_{i=1}^{n''} IRN_{vi}}$, 其中 IRN_{vi} 计算的是在单位时间内,内容 i 在节点 v 上的访问次数,$\sum_{i=1}^{n''} IRN_{vi}$ 计算的是单位时间内对内容的总访问量.

定义被请求的内容权重为 RW_{vi}, $RW_{vi} = \lg \dfrac{m}{m(i)}$, 其中 $m(i)$ 为对内容 i 产生请求的节点数,m 为节点总数. $\lg \dfrac{m}{m(i)}$ 定义的是一种全局因子,与具体节点无关,而与节点集合相关,当越少的节点产生对内容 i 的请求时,值越大,意味着请求内容 i 和节点 v 的相关性越强,通过对所有内容权重的计算,可以区分出不同内容的重要性.

1.2 匹配值的定义

内容与节点是否匹配问题是一个二分类问题,匹配值对应分类结果.

定义内容的缓存黏度为 CV_{vi}, $CV_{vi} = CRN_{vi} \times \lg \dfrac{m}{m'(i)}$, 其中 $m'(i)$ 为缓存有内容 i 的节点数量, $CRN_{vi} = \dfrac{crn_{vi}}{\sum_{i=1}^{n} crn_{vi}}$, crn_{vi} 表示的是单位时间内缓存内容 i 在节点 v 上的访问量, $\sum_{i=1}^{n} crn_{vi}$ 统计的是节点 v 上对所有被缓存内容的总访问量. 黏度与全网中缓存有内容 i 的节点数量成反比,与内容在缓存节点上的访问率成正比. 缓存黏度反映出被缓存内容与节点的匹配程度,值越大,说明二者匹配度越大,越适宜缓存.

对于缓存匹配这样一个二分类问题,定义分类匹配的门限值为 ε,当缓存黏度 $CV_{vi} \geqslant \varepsilon$ 时,意味着二者的匹配度较高,适宜缓存,则令匹配值为1,否则为-1. 门限值采用的是缓存黏度的中位值,能有效避免极端数据对门限值的影响.

1.3　数据集的定义

定义属性向量 $\boldsymbol{a}_{vi}=(\mathrm{CR}_{vi},\mathrm{RR}_{vi},\mathrm{LP}_{vi},\mathrm{RW}_{vi})=(a_{vi}^1,a_{vi}^2,a_{vi}^3,a_{vi}^4)$，为节点 v 与内容 i 的状态属性信息.

定义数据集 $A_{vi}=\{a_{11},a_{12},\cdots,a_{vi}\}$ 为 t 时刻的标记数据集，简记为 $A_{vi}=X_l=\{x_1,x_2,\cdots,x_l\}$，其中，$x_1=a_{11}$.

定义数据集 $Y_l=\{y_1,y_2,\cdots,y_l\}$ 为类别标记集合，其中，$y_j\in\{-1,1\}$，匹配值与类别标记一一对应，如果 $y_j=1$，二者匹配度高，则将内容缓存在节点上，反之匹配度低，不缓存.

定义数据集 $X_u=\{x_{l+1},x_{l+2},\cdots,x_{l+u}\}$ 是时间序列 $t+\Delta t$ 上的未标记数据集，其类别标记集合为 $Y_u=\{y_{l+1},y_{l+2},\cdots,y_{l+u}\}$，具体值未知，这也是测试数据集，若 $y_u=1$，则将内容缓存在节点上，若 $y_u=-1$，不缓存.

1.4　Adaboost 集成学习算法

本文的目标就是在给定训练集 X_l 与 Y_l 的情况下，预测测试集数据 X_u 对应的类别标记集合 Y_u，确定 y_{l+j} 的取值是 1 还是 −1. 由于单一的分类器在不同问题上的泛化表现可能不同，而按照一定的原则对多种分类器进行集成，则有可能实现算法性能的均衡和提升，减少因分类器选择不当而导致的泛化性能表现过差风险[16]，因此本文采用集成学习算法来进行缓存匹配问题的研究. 由于经典的 Adaboost(adaptive boosting) 集成学习算法不需要积累弱学习分类器的先验知识，只根据当前样本的分布进行学习，无需 ICN 对先验知识的存储与分析，因而采用 Adaboost 算法生成缓存匹配决策的集成分类器.

Adaboost 算法的思想是对训练集进行迭代学习，每次迭代生成的弱分类器的分类结果都与标记进行比较，然后根据样本分类是否正确来更新样本的分布，并将更新后的样本作为下一轮学习器的输入进行新弱分类器的学习，再根据弱分类器的误差率确定每个弱分类器的权重，最后加权组合生成集成分类器. Adaboost 算法流程如下：

1) 输入：训练样本 $\{(x_1,y_1),\cdots,(x_l,y_l)\}$，其中 $x_j\in X$，标签标记为 $y_j\in\{-1,1\}$；迭代次数 T.

2) 输出：集成分类器 h^*.

3) 初始化：赋予每个样本相等的权重，$w_j=1/l$.

4) for $k=1$ to T do

①在训练样本集上，利用样本权重 w^k 和弱分类算法学习得到弱分类器 $h_k=X\rightarrow Y$；

②计算弱分类器 h_k 的错误率：$\varepsilon_k=\sum_{j=1}^{l}w_j^kI(h_k(x_j)\neq y_j)$，

$$I(h_k(x_j)\neq y_j)=\begin{cases}1, & h_k(x_j)\neq y_j;\\ 0, & \text{其他.}\end{cases}$$

If $\varepsilon_k>0.5$ then

重新初始化每个样本的权重 $1/l$，并转向步骤 1)；

③计算弱分类器 h_k 在最终集成分类器中的加权系数：

$$\alpha_k=\frac{1}{2}\ln\left(\frac{1-\varepsilon_k}{\varepsilon_k}\right);$$

④下轮迭代时样本的权重更新为

$$w_j^{k+1}=w_j^k\exp(-\alpha_kI(h_k(x_j)=y_j))/\sum_{j=1}^{l}w_j^k\exp(-\alpha_kI(h_k(x_j)=y_j)).$$

5) 集成分类器为 $h^*(x)=\mathrm{sign}\left(\sum_{k=1}^{t}\alpha_kh_k(x)\right)$.

1.5　缓存算法

本文的基本思想是通过获得的关于节点和内容的多维状态属性值与缓存匹配值之间的对应关系进行分析挖掘，并利用挖掘出的属性值与匹配值之间的映射关系对未来时间段内的节点与内容间的匹配关系进行预测，预测方法主要分为 2 步，其整体流程如下：

首先，采用线性函数归一化方法对多维的状态属性数据进行预处理，通过将不同量纲的数据统一映射到同一取值空间，消除由于多维数据取值范围的不同对特征挖掘造成的影响.

由于多维的状态属性值与缓存黏度值反映了信息与节点的匹配关系特征，例如：将不同的状态属性值组合成矩阵，行信息表示的是不同节点间状态的差异，而这些差异代表的就是不同节点与不同内容的缓存黏度之间的区别，如图 1 所示. 因此，在归一化处理的基础上，采用 Adaboost 集成学习算法，利用多维状态属性值与匹配值之间的函数映射关系，预测未来时间段的对应关系. 具体的步骤如下：

步骤 1　建立训练样本及预测目标. 根据节点的多维历史状态属性值(缓存率、替换率、流行度、权重)和对应的缓存黏度值构造训练样本集合，并确立预测目标. 假设在时间 t_0 节点 v 对信息 i 的状态属性值为 $a_{vi}^1,a_{vi}^2,a_{vi}^3,a_{vi}^4$，缓存黏度为 CV_{vi}，则训练集可由 $\{(x_1,y_1),\cdots,(x_l,y_l)\}$ 来表示. 在时间 $t_0+\Delta t$ 时 (x',y') 就是预测目标.

步骤 2　构造映射函数. 根据训练样本集合

构造状态属性值和匹配值之间的映射函数 $h^*(x)$. 其基本思想是,通过对训练样本集合的多次迭代训练出多个弱分类器 $h_t(x)$,然后将多个弱分类器加权组成一个集成分类器 $h^*(x)$.

步骤 3　输入 $t_0+\Delta t$ 时刻的节点状态属性值到训练好的映射函数中,预测与内容之间的匹配值. 假设 Y_u 是未标记数据集的类别标记集,$X_u=\{x_{1+l},x_{2+l},\cdots,x_{l+u}\}$ 是未标记数据集,即测试集,则本方法预测的目标可表示为

$$Y_u=h^*(X_u)\ . \tag{1}$$

具体的求解过程在 1.4 节中有详细的描述.

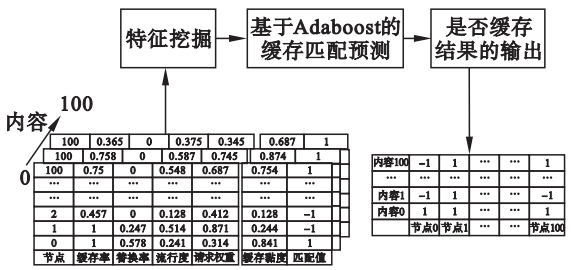

图 1　缓存算法流程示意图

Fig. 1　Flowchart of cache algorithm

2　仿真实验与分析

2.1　实验参数设置

本文使用开源的仿真软件 ndnSIM 对所提策略进行仿真实现,并与 CEE 策略[2]、LCD 策略[3]、prob0.5 策略[5]和 OPP 策略[12]的性能进行比较分析. 实验采用真实的域间拓扑结构 AS-1755,假设用户的平均请求速率为每秒 100 个兴趣包,请求模式遵循 Zipf 分布,网络中对内容的请求过程呈现泊松分布特性. 假定每个内容缓存时占用一个缓存单位,初始状态时每个节点无缓存内容,网络中需缓存的内容总量为 71 000 个.

2.2　实验结果

为了观察网络性能分别受缓存容量及 Zipf 参数的影响,在实验过程中,一次实验只修改一个参数,其余参数保持不变.

2.2.1　缓存容量的影响

通过对网络缓存容量(0.25 ~ 1.5 GB)的改变,观察缓存性能如延时、命中率和链路利用率 3 个评价指标的变化趋势(图 2).

从图 2 可以看出,随着缓存容量的增加,5 种缓存策略的命中率逐渐增大,而延时和链路利用率逐渐减少. 这是由于随着缓存容量的增加,节点可缓存的内容也相应增加,用户从中间节点获得所需内容的概率加大,因此命中率增大,延时和链路利用率减少. 对比分析可以看出,ACAL 在延时、命中率及链路利用率等 3 项评价指标上均优于另外 4 种缓存策略,这是由于 CEE 策略、LCD 策略、prob0.5 策略对内容重复冗余缓存,使缓存内容的多样性不足,缓存性能较差;OPP 策略考虑了内容的流行度与节点位置的匹配关系,缓存性能较好;而 ACAL 策略通过对节点数据和内容数据的学习,进一步提高了缓存空间的利用率和服务效率.

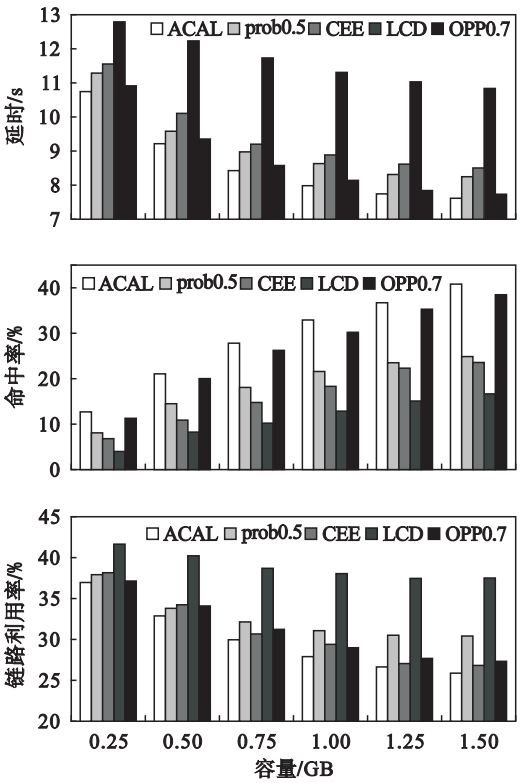

图 2　不同缓存性能随缓存容量的变化趋势

Fig. 2　Change trend of different cache performance with cache capacity

2.2.2　Zipf 参数的影响

通过对 Zipf 参数(0.7 ~ 1.5)的修改,观察在缓存容量为 1.5 GB 时,缓存性能如延时、命中率和链路利用率 3 个评价指标的变化趋势(图 3).

从图 3 中可以看出,随着 Zipf 参数的增大,5 种缓存策略的命中率逐渐增大,而延时和链路利用率逐渐减少. 这是因为随着 Zipf 参数的增大,网络中对热点内容的请求度越来越高. 由于 CEE 策略、LCD 策略、prob0.5 策略对内容流行度变化不够敏感,因而性能改善有限;随着流行度因子的增大,流行内容越来越集中,OPP 策略与 ACAL 策略能将流行内容缓存在合理的节点上,但是 ACAL 的性能更优于 OPP.

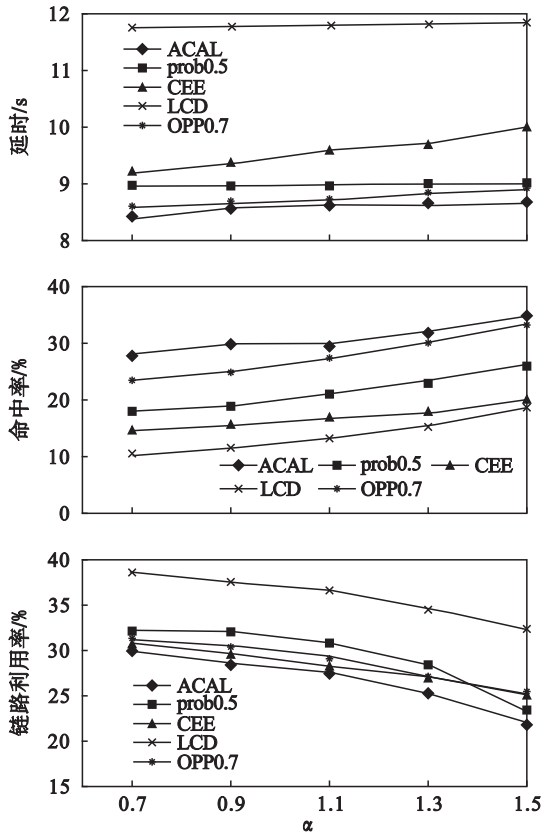

图 3　不同缓存性能随 Zipf α 的变化趋势

Fig. 3　Change trend of different cache performance with Zipf α

3　结　　论

本文提出一种基于集成学习的自适应缓存算法 ACAL，通过对兼顾节点和内容信息的多维状态数据和缓存匹配值之间关系的集成学习，将学习规律用于预测下一阶段的节点与内容的匹配关系. 实验结果表明，ACAL 与 CEE 策略、LCD 策略、prob0. 5 策略和 OPP 策略相比，在延时、命中率和链路利用率等方面，性能都有所提升.

参考文献：

[1] Cisco. Cisco visual networking index: forecast and methodology[EB/OL]. (2017 - 9 - 15) [2018 - 7 - 20]. https://www. cisco. com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11 - 481360. html#_Toc 484813971.

[2] Ghodsi A, Shenker S, Koponen T. Information-centric networking: seeing the forest for the trees[C]//Proceedings of the 10th ACM Workshop on Hot Topics in Networks. New York: ACM, 2011: 483 - 510.

[3] Laoutaris N, Che H, Stavrakakis I. The LCD interconnection of LRU caches and its analysis[J]. *Performance Evaluation*, 2006, 63(7): 609 - 634.

[4] 蔡岳平，刘军，樊欣唯. 基于节点中心性度量的内容中心网络缓存机制[J]. 通信学报，2017，38(6)：10 - 18.
(Cai Yue-ping, Liu Jun, Fan Xin-wei. Node centrality metric based caching mechanism in content-centric network[J]. *Journal on Communication*, 2017, 38(6): 10 - 18.)

[5] Laoutaris N, Syntila S, Stavrakakis I. Meta algorithms for hierarchical Web caches[C]//IEEE International Conference on Performance, Computing, and Communications. Phoenix: IEEE, 2005: 445 - 452.

[6] Psaras I, Wei K C, Pavlou G. In-network cache management and resource allocation for information-centric networks[J]. *IEEE Transactions on Parallel & Distributed Systems*, 2014, 25(11): 2920 - 2931.

[7] Wu H, Li J, Zhi J. MBP: a max-benefit probability-based caching strategy in information-centric networking[C]// IEEE International Conference on Communications. London: IEEE, 2015: 5646 - 5651.

[8] 吴海博，李俊，智江. 基于概率的启发式 ICN 缓存内容放置方法[J]. 通信学报，2016，37(5)：62 - 72.
(Wu Hai-bo, Li Jun, Zhi Jiang. Probability-based heuristic content placement method for ICN caching[J]. *Journal on Communications*, 2016, 37(5): 62 - 72.)

[9] Dehghan M, Massoulie L, Towsley D, et al. A utility optimization approach to network cache design[C]//IEEE INFOCOM 2016—IEEE International Conference on Computer Communications. San Francisco: IEEE, 2016: 1 - 9.

[10] Li W, Oteafy S M A, Hassanein H S. Stream cache: popularity-based caching for adaptive streaming over information-centric networks[C]//IEEE International Conference on Communications. Kuala Lumpur: IEEE, 2016: 1 - 6.

[11] Kim D, Lee S W, Ko Y B, et al. Cache capacity-aware content centric networking under flash crowds[J]. *Journal of Network & Computer Applications*, 2015, 50(C): 101 - 113.

[12] Hu X Y, Gong J. Opportunistic on-path caching for named data networking[J]. *IEICE Transactions on Communications*, 2014, E97-B(11): 2360 - 2367.

[13] Wang W, Yi S, Yang G, et al. CRCache: exploiting the correlation between content popularity and network topology for ICN caching[C]//2014 IEEE International Conference on Communications. Sydney: IEEE, 2014: 3191 - 3196.

[14] 张果，胡宇翔，黄万伟. 基于流行内容感知和跟踪的协同缓存策略[J]. 通信学报，2017，38(2)：132 - 142.
(Zhang Guo, Hu Yu-xiang, Huang Wan-wei. Coordinated caching scheme based on popular content awareness and tracking[J]. *Journal on Communications*, 2017, 38(2): 132 - 142.)

[15] 李俊，冯宗明，吴海博，等. 基于层次划分的 CCN 网络缓存存储策略[J]. 通信学报，2016，37(1)：35 - 41.
(Li Jun, Feng Zong-ming, Wu Hai-bo, et al. Hierarchical division-based cache storage strategy in content-centric networking[J]. *Journal on Communications*, 2016, 37(1): 35 - 41.)

[16] Polikar R. Ensemble based systems in decision making[J]. *IEEE Circuits & Systems Magazine*, 2006, 6(3): 21 - 45.