Adaptive algorithms for efficient content management in social network services

Claudia Canali, Michele Colajanni, Riccardo Lancellotti
University of Modena and Reggio Emilia
{claudia.canali, michele.colajanni, riccardo.lancellotti}@unimore.it

Abstract

Identifying the set of resources that are expected to receive the majority of requests in the near future, namely hot set, is at the basis of most content management strategies of any Web-based service. Here we consider social network services that open interesting novel challenges for the hot set identification. Indeed, social connections among the users and variable user access patterns with continuous operations of resource upload/download determine a highly variable and dynamic context for the stored resources. We propose adaptive algorithms that combine predictive and social information, and dynamically adjust their parameters according to continuously changing workload characteristics. A large set of experimental results show that adaptive algorithms can achieve performance close to theoretical ideal algorithms and, even more important, they guarantee stable results for a wide range of workload scenarios.

1. Introduction

Social networks represent a new class of Web-based services that support interaction, knowledge and resource sharing among communities of online users. These services are characterized by novel patterns of access where user operations are not limited to navigation and download, but also to upload resources, insert short comments, create links with other users. These novel patterns require a re-design of the traditional strategies for an efficient content management, such as replication, caching, pre-fetching and pre-generation of resources [16, 24, 6]. All these strategies have to determine the subset of resources that are expected to receive more requests in the near future (the so called *hot set*). Operating just or mainly on the hot set rather than on the entire working set allows the content provider to limit the number of expensive content management operations at the level of network, storage and computational power [8, 21].

Most algorithms for the hot set identification were proposed in the so called Web 1.0 period and they basically rely on information about the past resources accesses [23,2]. They can achieve good results in traditional Web-based services, where the resource popularity changes slowly and according to known patterns. On the other hand, these algorithms are unsuitable to forecast the hot set in social network services, where the presence of user generated content [7,17] and social connections among users lead to quite new access patterns to the resources [18,7,17,25], thus causing frequent and rapid changes in resource popularity.

A previous study from the same authors [5] demonstrates that the accuracy of the hot set identification in social network services can be improved with respect to traditional solutions by adopting predictive models and by taking into account the characteristics of user social connections. However, how to merge in an efficacious way heterogeneous information, such as prediction on future accesses and social-related data characterized by different temporal dependencies and probability distributions, is still an open issue. The combination of multiple information must improve the *accuracy* of the hot set identification, and, even more important, must guarantee *robust* performance even in the highly variable scenario of social network services.

The main contribution of this paper is the proposal of a novel class of algorithms for the hot set identification that exploit adaptive techniques to automatically tune the process of merging predictive- and social-based information according to continuously variable workload characteristics. The proposed adaptive algorithms are evaluated to analyze the accuracy of their hot set identification and the robustness of the achieved performance for a large set of workload parameters. Our experiments demonstrate that adaptive techniques are a fundamental element for accurate and stable performance in the highly variable context of social network services. The proposed algorithms can achieve an accuracy for the hot set identification which is close to that of an ideal theoretical algorithm and, even more important, the achieved performance are insensible to a wide range of workload parameters. On the other hand, we show that any static combination of predictive- and social-based metrics may lead to results that are unacceptably unstable.

The remainder of the paper is organized as follows. Section 2 describes the main challenges in identifying the resource hot set for social network services. Section 3 presents the proposed algorithms. Section 4 describes the experimental results. Section 5 discusses the related work. Section 6 concludes the paper with some final remarks.

2. Motivation

The identification of the hot set represents a key task for efficient content management strategies. Its importance derives from the characteristics of the resource popularity in Web-based and in social network services. There is a large literature demonstrating that dimension, popularity, and frequency of the Web resources follow some power law distribution, especially some Zipf-like distribution [4, 7, 26]. These results spanning 15 years of Web characterization confirm that many user requests refer to a limited subset of popular resources.

The hot set identification aims to select the resources that are likely to obtain the majority of accesses in the near future. Because of popularity variations, we can consider that the hot set identification is a periodic task with period Δt.

The identification of the hot set is at the basis of typical content management strategies, such as *content pre-adaptation*, *replication*, *pre-fetching* and *CDN delivery* [8, 21, 10].

Content pre-adaptation in social network services represents a necessary task that will gain even more importance in the next future. The increasing diffusion of mobile Web-enabled devices with small display and limited connectivity [1] requires to tailor social network services and resources to the device capabilities [8, 6]. The computationally expensive operations involved in tailoring contents tend to be carried out offline on a subset of resources [6] and limit on-the-fly adaptation only to the less popular resources, if and when they are requested. In this context, the identification of the best hot set plays a fundamental role because it can limit the expensive pre-adaptation tasks to the most popular resources of the working set.

Content replication strategies are widely used in the context of geographically distributed infrastructures [23, 14, 16]. Replicating resources from some central server(s) to the nodes of the distributed infrastructure improves the scalability, limits the risk of bottlenecks and reduces network-related delays. The hot set identification represents a key element to determine which resources have to be replicated to maximize the number of requests that may be satisfied without accessing the central server, while limiting the overhead for replica consistency.

Content pre-fetching is a well-know technique used to reduce the user perceived response time in Web contexts. Basically, this technique consists in pushing resources into the cache of servers and reverse proxies to reduce the latency in serving the following Web requests [11, 21, 24]. Limiting the pre-fetching operation to the resources that will receive more requests in the next future is a necessary task to avoid the waste of bandwidth and storage space in servers and reverse proxies.

CDN delivery allows the content provider to assign a fraction of its resources to a third-party Content Delivery Network for scalable and high performance delivery. CDN infrastructures deliver popular static and multimedia resources [10, 21], and have been recently applied to social network services, such as YouTube [15]. Forcing these third-party infrastructures to deliver the entire working set is not convenient because of the high costs incurred when traffic is directed to the CDN. Identification of the hot set may limit the CDN delivery only to the most popular resources.

Although the hot set is fundamental for several content management strategies, the novel access patterns and workload trends characterizing the social network services [15, 18] reduce the effectiveness of existing solutions for hot set identification. In [5] we demonstrate the benefit of merging predictive techniques and social information for an effective hot set identification. However, determining the best approach to combine predictive- and social-based metrics represents an open issue for the following reasons.

- The two metrics show very different temporal dependencies: the resource access patterns are highly variable and change very frequently, while the explicit social connections between users change very slowly.
- Predictive metrics are characterized by heavy tailed distributions, with values that can span over several orders of magnitude.
- The workload of social network services is highly dynamic and characterized by rapid changes in the resource popularity. Since the combination of two heterogeneous metrics can amplify the effects of workload variability, an adaptive reconfiguration of the combination parameters may be necessary to avoid instability of results.

We propose different techniques to combine predictive- and social-based metrics, and we evaluate the integration of adaptive controls that automatically tune the combination process to achieve robust performance in the highly variable context of social network services.

3. Algorithms for hot set identification

We now describe the proposed algorithms for the hot set identification. Each algorithm operates at time t on the whole working set $R(t)$ of the social network and tries to compute the expected resource popularity $p_r(t)$ for every resource r belonging to the working set. Resources with the

Symbol	Description
Δt	Period between subsequent runs of the hot set identification algorithms
$R(t)$	Working set at current time t
r	Resource $r \in R(t)$
$a_r(t)$	Age of resource r at time t computed as time elapsed since resource upload
$p_{r,alg}(t)$	Estimated popularity of resource r computed at time t with algorithm alg
$k_{r,alg}(t)$	Popularity rank of resource r computed at time t with algorithm alg
$P_{alg}(t)$	$P_{alg}(t) = \{p_{r,alg}(t) \forall r \in R(t)\}$

Table 1. Symbols used in algorithm formalization

highest popularity are expected to receive more requests in the next period $[t, t + \Delta t]$. Hence, given the expected popularity of each resource in the working set, the hot set can be identified by sorting the workload and selecting the most popular resources. Table 1 provides a summary of the notations used in the algorithms description.

Our proposal consists in three *Predictive-Social* algorithms, namely *Rank-Age*, *Linear-Adaptive*, and *Rank-Adaptive*, that use different techniques to combine predictive- and social-based metrics for the resource popularity estimation. The *Linear-Adaptive* and the *Rank-Adaptive* algorithms introduce an adaptive control to dynamically adjust the algorithm parameters according to the present and past workload characteristics, while the *Rank-Age* algorithm is static. Furthermore, the *Linear-Adaptive* algorithm exploits a linear combination to merge the two metrics, while the *Rank-Age* and the *Rank-Adaptive* algorithms rely on a rank merging technique, that has been proposed in the context of search engines of text and images [13, 20] to address the issue of combining values characterized by heterogeneous statistical properties. It is worth to note that the three proposed Predictive-Social algorithms have a similar computational complexity, that is $\mathcal{O}(n \log n)$ due to the sorting operation required in each algorithm.

3.1. Existing algorithms

To better understand the behavior of the proposed algorithms for the hot set identification, we briefly present the *Predictive-based* and the *Social-based* algorithms, that rely on a single metric (that is prediction and social information, respectively) to estimate the resource popularity. These algorithms have already been proposed in literature [22, 18, 17, 5] and we exploit them as building blocks of the proposed Predictive-Social algorithms and as a comparison from a performance point of view.

The Predictive-based algorithm exploits information available at time t on the past access patterns to predict the future accesses to each resource. Specifically, the algorithm uses a time series describing the past access history to resource r, where each sample in the time series corresponds to a period of length Δt. The predictive model is used to compute $p_{r,predictive}(t)$ as the expected number of accesses in the period $[t, t + \Delta t]$. To this aim, we exploit a predictive model based on the Exponential Weighted Moving Average (EWMA) function [22]. The main advantages of this model are simplicity and robustness, that make it suitable to run-time and highly variable scenarios. The EWMA model adopts weighting factors decreasing exponentially for older data points, thus giving more importance to recent values while not discarding older observations.

The Social-based algorithm computes the hot set by exploiting information on the user social connections, leveraging the strong correlation between the amount of accesses to a resource and the number of social links of the uploading user. Indeed, most social network services allow users to designate other members of the online community as their contacts. Hence, the number of *reverse contacts* of the users can be considered as a measure of their social network size, where a reverse contact for A is a user that has designated A as a contact [17]. The algorithm uses the number of reverse contacts as a measure of the resource popularity $p_{r,social}(t)$, because resources uploaded by users with many social connections are likely to receive more accesses in the future [18, 17, 25]. Furthermore, the resource popularity estimated by the algorithm is inversely proportional to the resource age, because older resources are less likely to become popular with respect to newly uploaded resources [5].

3.2. Rank-Age algorithm

The Rank-Age algorithm combines predictive- and social-based popularity estimations through a rank merging technique [20]. Instead of considering the estimated popularity $p_r(t)$ for each resource $r \in R(t)$, this algorithm uses the resource rank $k_r(t)$ when the resources are sorted according to $p_r(t)$. This algorithm is static because the coefficient of the rank merging operation is based on a per-resource information, that is the age $a_r(t)$.

Figure 1 shows the steps of the algorithm: the estimated probability $p_{r,social}(t)$ and $p_{r,predictive}(t)$ at time t are sorted to obtain the corresponding rank values $k_{r,social}(t)$ and $k_{r,predictive}(t)$.

The rank merging step combines the ranks according to a weight that depends on the resource age. For younger resources we assign more weight to the social-based popularity estimation, because for these resources the predictive approach cannot rely on long time series and achieves poor performance. On the other hand, for older resources we as-

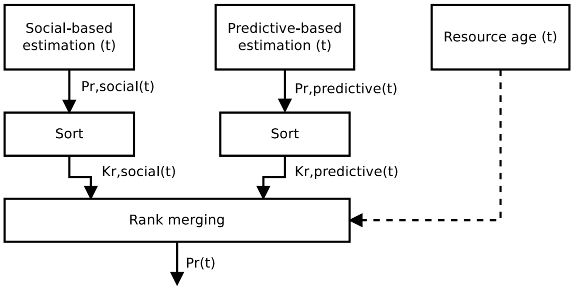

Figure 1. Scheme of the Rank-Age algorithm

sign more weight to the predictive-based popularity estimation. We formalize the rank merging as:

$$p_r(t) = \frac{a_r(t)}{a_{max}(t)} k_{r,predictive}(t) + \left(1 - \frac{a_r(t)}{a_{max}(t)}\right) k_{r,social}(t)$$

where $a_r(t)$ is the age of resource r and $a_{max}(t)$ is the age of the oldest resource in the working set $R(t)$ at time t.

3.3. Linear-Adaptive algorithm

The Linear-Adaptive algorithm uses a linear combination to merge popularity estimation obtained from the Predictive- and Social-based algorithms. The adaptivity of the algorithm lies in the choice to re-compute every time the parameters of the linear combination on the basis of the statistical characteristics of the current working set. Figure 2 shows the main steps of the algorithm: for each resource r, it evaluates the social-based $p_{r,social}(t)$ and the predictive-based metric $p_{r,predictive}(t)$.

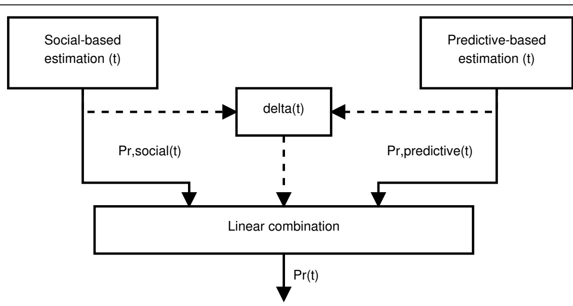

Figure 2. Scheme of the Linear-Adaptive algorithm

The linear combination uses a weight $\delta(t)$ to merge the information from the two algorithms:

$$p_r(t) = \delta(t) p_{r,predictive}(t) + (1 - \delta(t)) p_{r,social}(t) \quad (1)$$

where $\delta(t)$ is computed on the basis of the statistical properties of $p_{r,predictive}(t)$ and $p_{r,social}(t)$ referred to the whole working set. Let $P_p(t) = \{p_{r,predictive}(t), \forall r \in R(t)\}$ and $P_s(t) = \{p_{r,social}(t), \forall r \in R(t)\}$ be the popularity of all the resources of the working set derived from the two algorithms at time t. We aim to normalize the *central tendency* of the two distributions $P_p(t)$ and $P_s(t)$. Since median and average values are not representative in a context where different heavy-tailed distributions must be combined, in our adaptive computation of $\delta(t)$ we exploit a percentile filtering techniques that combines first (Q_{25}), second (Q_{50}) and third (Q_{75}) quartiles to obtain a coefficient that is independent of any assumption on the measure distributions [9].

$$\begin{aligned}\delta(t) \quad = \quad & 0.25 \frac{Q_{25}(P_s(t))}{Q_{25}(P_s(t)) + Q_{25}(P_p(t))} + \\ & +0.50 \frac{Q_{50}(P_s(t))}{Q_{50}(P_s(t)) + Q_{50}(P_p(t))} + \\ & +0.25 \frac{Q_{75}(P_s(t))}{Q_{75}(P_s(t)) + Q_{75}(P_p(t))}\end{aligned}$$

The computation of the weight $\delta(t)$ introduces the adaptive behavior of the algorithm, because the weights of the linear coefficients in Equation 1 are dynamically adapted according to the workload variations.

3.4. Rank-Adaptive algorithm

The Rank-Adaptive algorithm exploits the same rank merging approach of the Rank-Age algorithm, but it introduces an adaptive control based on a run-time feedback. Basically, the weights used for the rank merging operation are assigned on the basis of a feedback control, that takes into account the previous hot set estimation. In this way, the popularity estimation errors are exploited to adaptively adjust the algorithm behavior according to the workload variations.

As shown in Figure 3, at time t, the algorithm ranks the resources according to the Predictive- and Social-based mechanisms, and the rank merging operation determines the expected popularity $p_r(t)$ for each resource $r \in R(t)$. The ranks computed through the predictive- and social-based mechanisms are stored and used to evaluate the effectiveness of each algorithm: at time t, the algorithm uses the information of the popularity estimation $t - \Delta t$. For each resource r, the estimated popularity rank is compared with the actual resource popularity rank $k^*_r(t - \Delta t)$ that is based on the resource accesses in the interval $[t - \Delta t, t]$ over the entire working set. We define the error of the Social- and Predictive-based algorithms over the previous hot set identification as $\epsilon_{r,social}(t - \Delta t) = \text{abs}(k^*_r(t - \Delta t) - k_{r,social}(t - \Delta t))$ and $\epsilon_{r,predictive}(t - \Delta t) = \text{abs}(k^*_r(t - \Delta t) - k_{r,predictive}(t - \Delta t))$, respectively. The rank merging operation computes $p_r(t)$ as follows:

$$p_r(t) = \gamma(t) k_{r,predictive}(t) + (1 - \gamma(t)) k_{r,social}(t),$$

where $\gamma(t) = \frac{\epsilon_{r,social}(t-\Delta t)}{\epsilon_{r,social}(t-\Delta t)+\epsilon_{r,predictive}(t-\Delta t)}$. The weight $\gamma(t)$ is computed by taking into account the workload characteristics and the effectiveness of the algorithms during the previous hot set estimation, thus introducing the adaptive control in the Rank-Adaptive algorithm.

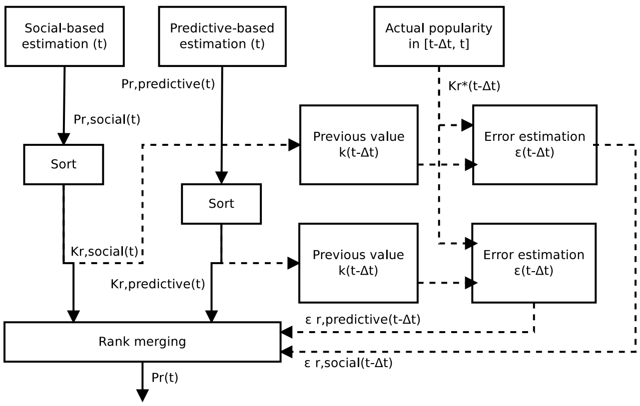

Figure 3. Scheme of the Rank-Adaptive algorithm

4. Experimental results

4.1. Experimental testbed

We evaluate the performance of the algorithms for hot set identification through a discrete event simulator based on the Omnet++ framework. We implement a simulated Web system where a server receives requests from a population of concurrent users.

As it was not possible to access the file logs of popular sites of social network services, we rely on models based on the state-of-the-art workload characterizations. A client interaction with the server may involve the download or the upload of a resource, where the percentage of upload operations may range from 1% up to 20%. This large range includes the case of highly interactive social network applications, such as blogs, where users typically post lots of comments for every blog entry [12]. The amount of upload operations affects the working set size and the resource popularity. Any new upload, indeed, increases the working set size. Furthermore, a high number of uploads tends to cause a turnover within the hot set because recently uploaded resources may replace previously popular contents. In the upload operations, the workload model considers also the dependency of the resource popularity on the user connection degree in the social networks. Accordingly to [18], the correlation factor between resource popularity and user connection degree (namely, *user/resource popularity correlation*) may range from 0.6 to 0.8.

For our experiments we consider a user population that grows from 10000 to 20000 individuals during the experiment duration. Users issue requests to the server for an incoming traffic rate of 100 requests per second on average. The results are collected over runs of 12 hours of simulated time. Each experiment is repeated 10 times and the results are averaged over the runs. We choose a hot set identification interval $\Delta t = 20$ minutes, that is commonly adopted in these contexts, but we should observe that the choice of the best period Δt for the hot set identification may represent an interesting research topic by itself. The naive solution of addressing the high variability of the hot set through very short periods of hot set re-evaluation is not feasible, because it would cause an excessive overhead in terms of computational power, storage space and network bandwidth.

To evaluate the effectiveness of the hot set identification, we consider as a term of comparison the so called ideal hot set, that is obtained by a theoretical algorithm that at time t knows the future access patterns in the interval $[t, t + \Delta t]$ for every resource in the current working set $R(t)$. We consider as the main performance metric the algorithm *accuracy*, that is the number of resources correctly identified as hot (the resources that are both in the hot set estimated by the algorithms and in the ideal hot set), normalized according to the hot set size.

The *robustness* of the results is a critical parameter for the evaluation of the algorithms. Due to the high variability of the workload, solutions that may guarantee stable performance over a wide set of scenarios are preferable with respect to algorithms that achieve the best peak performance for a specific scenario and poor performance elsewhere. To this purpose, we perform a sensitivity analysis with respect to several workload and algorithm parameters: the *hot fraction*, that is the size of the hot set as a fraction of the entire working set, the *upload percentage*, that is the percentage of upload operations over the total number of client requests, and the *user/resource popularity correlation*, that is the correlation between resource popularity and user connection degree. Table 2 summarizes the range and default values in the experimental setup.

Parameter	Range	Default
Hot fraction [%]	5% – 30%	20%
Upload percentage [%]	1% – 20%	5%
User/resource popularity correlation	0.6 – 0.8	0.7

Table 2. Experimental setup parameters

4.2. Algorithms evaluation

Figure 4 shows the performance of the Predictive-Social algorithms compared with Predictive- and Social-Based algorithms for different values of the considered hot fraction. We observe that the Linear-Adaptive and Rank-Adaptive algorithms consistently outperform all the other solutions in the hot set identification for every value of the hot fraction. In particular, the performance of the Rank-Adaptive algorithm is close to that of the best theoretical algorithm, with an accuracy of almost 95%. We can deduce that the presence of an adaptive mechanism to adjust the combination parameters of predictive- and social-based information is a key element for improving the accuracy of the hot set identification.

On the other hand, we observe that a static combination of predictive and social elements may lead to poor results, that are even worse with respect to existing algorithms for high hot fraction values. As shown in Figure 4, the Rank-Age algorithm achieves performance comparable with the Linear-Adaptive algorithm only when the hot fraction is below 10%, while its accuracy degrades as soon as the hot fraction increases. This apparently counterintuitive result can be explained by considering the average resource age of the hot set estimated by the Rank-Age algorithm, that is up to 40% lower with respect to the ideal hot set. Hence, the Rank-Age algorithm appears to overestimate the impact of the resource age on the expected popularity. As the hot fraction increases, the algorithm tends to prefer young resources that are not very popular instead of older resources that may still have a high number of accesses.

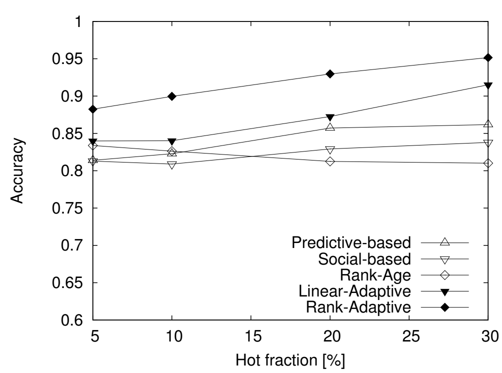

Figure 4. Performance comparison of the algorithms

We now present an extensive sensitivity analysis to evaluate which Predictive-Social solutions may guarantee the most robust performance. This analysis is extremely important for the Predictive-Social algorithms because the combination of different strategies for popularity estimation involves several parameters, and a higher complexity of the algorithms may lead to unstable performance.

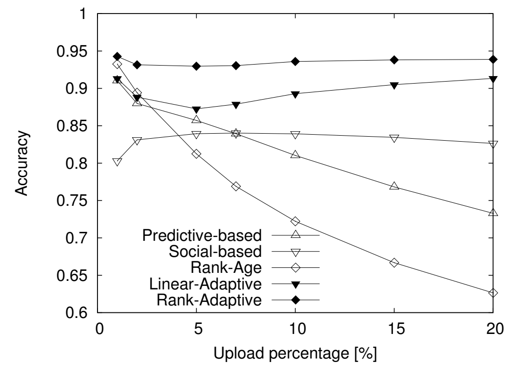

Figure 5. Sensitivity to upload percentage

Figure 5 shows the sensitivity of the Predictive-Social algorithms to the upload percentage, that represents a measure of the workload variability. We observe that the Linear-Adaptive and the Rank-Adaptive algorithms not only achieve the best results: the presence of a run-time adaptive mechanism provides stable performance even for highly dynamic workloads. This is confirmed by the almost negligible variation of the algorithms accuracy as the upload percentage ranges from 1% to 20%. On the other hand, the Rank-Age and the Predictive-based algorithms are highly sensitive to the workload dynamic behavior, with a performance degradation of about 30% for the Rank-age algorithm when the upload percentage passes from 1% to 20%. The sensitivity of Predictive-based algorithms is expected because for the newly uploaded resources the time series of past accesses is too short for an effective prediction. On the other hand, the sensitivity of Rank-Age algorithm to the upload percentage has the same cause of its poor performance, shown in Figure 4: the algorithm overestimates the importance of the young age for the resource popularity, thus identifying the youngest resources as belonging to the hot set. When the workload is highly dynamic, there is a large amount of very young resources and the algorithm decisions become less accurate.

Finally, we investigate the sensitivity of the algorithm accuracy with respect to the correlation between the resource popularity and the connection degree of the uploading user. This parameter is likely to affect the accuracy of the algorithms that rely on social-based metrics. From Figure 6 we observe that the performance of the Social-based algorithm is highly sensitive to the variations of the user/resource popularity correlation, while, as expected, the Predictive-based algorithm is insensitive to this parameter. The most impor-

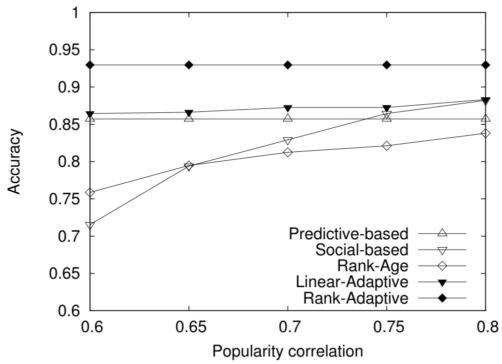

Figure 6. Sensitivity to user/resource popularity correlation

tant result is that the performance variation of the Linear-Adaptive and the Rank-Adaptive algorithms remains far below 5% for when the user/resource popularity correlation parameter ranges from 0.6 to 0.8. On the other hand, the Rank-Age algorithm shows a higher sensitivity to the user/resource popularity correlation. This result may be motivated by the algorithm behavior, that tends to select young resources for the hot set and, consequently, to assign more weight to the social-based metric.

We can summarize the main findings of our experiments as follows. We confirm that the combination of heterogeneous data coming from predictive models and social connections remains a complex problem, where a not robust algorithm may even amplify the effects of the workload characteristics, leading to unacceptably unstable performance. On the other hand, we demonstrate that the use of an adaptive control, that adjusts the combination parameters, is a key element to achieve significant gains in terms of accuracy and stability in case of highly variable workloads.

5. Related work

The combined effect of highly interactive user access patterns and social connections opens novel issues related to the efficient content management in social network services. The focus of our research is on the specific issue of identifying the hot set of the most popular resources, because it represents a key element for several content management strategies, such as replication, caching, pre-fetching and pre-generation of adapted resource versions [21, 8, 24, 16].

For traditional Web-based applications, where the resource popularity changes slowly through predictable patterns, the hot set may be accurately identified by analyzing the past resource accesses [21, 23]. However, in social network services the resource popularity follows novel patterns, changing rapidly due to frequent resource uploads and to the social connections among users, thus hindering the effectiveness of existing algorithms [7, 15, 12]. These results motivate the need of integrating predictive- and social-based techniques in the algorithms for the hot set identification. Predictive models have been widely used in Web-based applications. In particular, time series have been often adopted to model past measurements and predict future behavior in different but related Web contexts, such as Internet traffic [22], server load [2], and hot spots [3]. The impact of social connections on the resource popularity in social network services has been demonstrated by several studies in the last few years [18, 17, 25]. However, both predictive- and social-based metrics have never been exploited for hot set identification purposes in social network services.

A first attempt to combine user access patterns and information on social connections for the hot set identification has been investigated by the authors in [5]. This study demonstrates the potential benefits of considering both these information to accurately identifying the hot set in the context of social network services, showing the poor performance of algorithms considering just one metric. However, the combination of heterogeneous metrics characterized by different statistical properties represents a not trivial task.

The most straightforward approach to this problem is represented by a linear combination of the heterogeneous metrics; however, the identification of the best linear coefficients represents a critical choice. An alternative approach is to use rank merging algorithms to combine heterogeneous information, as in the context of search engines for text and images [13, 20]. However, to the best of our knowledge, rank merging has never been applied to problem of hot set identification, and as in the case of linear combination, the coefficient for the rank merging operation must be carefully selected.

Static coefficients are not suitable for a highly variable context such as social network services, because they cannot ensure stable performance. Hence, we consider the effect of adaptive mechanisms. Adaptivity has been proposed in different contexts related to Web-based systems [19, 27]. Lu *et al.* [19] make use of a feedback control in a time-critical context, that is the development of an adaptive Web server architecture offering services with QoS constraints. In [27] the adaptive technique is used to improve the performance of a ranking algorithm for commercial search engines: an offline calculation of the ranking errors allows the authors to adjust the search engine behavior. On the basis of these results, we introduce adaptive techniques in the algorithms for the hot set identification to improve the robustness of the results in presence of highly variable workloads.

6. Conclusions

Social network services are characterized by access patterns, such as highly interactive accesses, user generated contents and social connections among users, that have completely changed the workload with respect to traditional Web-based applications. As a consequence, even strategies for an efficient content management should change as well. This paper considers the problem of the hot set identification for social network services that is at the basis of most content management strategies. We propose and compare algorithms that exploit an adaptive combination of predictive models and social information to estimate the hot set. Our study demonstrates that any static approach achieves poor performance or too variable results for different workload and scenarios. On the other hand, adaptation is an essential feature for algorithms that must combine heterogeneous information in highly variable scenarios. Indeed, the proposed adaptive algorithms achieve performance close to the ideal algorithm and guarantee robust performance with respect to any considered workload parameter.

Acknowledgements

The authors acknowledge the support of MIUR-PRIN project DOTS-LCCI "Dependable Off-The-Shelf based middleware systems for Large-scale Complex Critical Infrastructures".

References

[1] ABI Research Report. Social Networking Goes Mobile. Mobile Content Research Service, September 2008.

[2] M. Andreolini, S. Casolari, and M. Colajanni. Models and framework for supporting runtime decisions in Web-based systems. *ACM Transactions on the Web*, 2(3):1–43, 2008.

[3] Y. Baryshnikov, E. Coffman, G. Pierre, D. Rubenstein, M. Squillante, and T. Yimwadsana. Predictability of Web-server traffic congestion. In *Proc. of WCW Workshop*, Sep. 2005.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proc. of IEEE Infocom 1999*, Mar. 1999.

[5] C. Canali, M. Colajanni, and R. Lancellotti. Hot set identification for social network applications. In *Proc. of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC'09)*, Jul. 2009.

[6] C. Canali, M. Colajanni, and R. Lancellotti. Performance impact of future Mobile-Web based services on the server infrastructure. *IEEE Internet Computing*, Feb. 2009.

[7] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proc. of ACM SIGCOMM Conference*, San Diego, CA, Oct 2007.

[8] M. Colajanni, R. Lancellotti, and P. Yu. Distributed architectures for Web content adaptation and delivery. *Web Content Delivery*, 2005. (Tang, Xu, Chanson eds.), Springer.

[9] S. A. Crosby, D. S. Wallach, and R. H. Riedi. Opportunities and limits of remote timing attacks. *ACM Transactions on Information and System Security*, 12(3):1–29, 2009.

[10] A. Davis, J. Parikh, and W. E. Weihl. EdgeComputing: extending enterprise applications to the edge of the Internet. In *Proc. of WWW Conference*, pages 180–187, 2004.

[11] J. Domènech, J. A. Gil, J. Sahuquillo, and A. Pont. Web prefetching performance metrics: a survey. *Performance Evaluation Journal*, 63(9):988–1004, 2006.

[12] F. Duarte, B. Mattos, A. Bestavros, V. Almeida, and J. Almeida. Traffic Characteristics and Communication Patterns in Blogosphere. In *Proc. of ICWSM Conference*, Seattle, WA, Apr. 2007.

[13] J. C. French, J. V. S. Watson, X. Jin, and W. N. Martin. W.n.: Integrating multiple multichannel cbir systems. In *Proc. of MIS Workshop*, pages 85–95, 2003.

[14] L. Gao, M. Dahlin, A. Nayate, J. Zheng, and A. Iyengar. Application specific data replication for edge services. In *Proceedings of WWW Conference*, Budapest, HU, 2003.

[15] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube traffic characterization: A view from edge. In *Proc. of IMC Conference*, Oct. 2007.

[16] M. Karlsson. Replica placement and request routing. *Web content delivery*, 2005. (Tang, Xu, Chanson eds.), Springer.

[17] K. Lerman. Social Information Processing in News Aggregation. *IEEE Internet Computing*, 11(6):16–28, 2007.

[18] K. Lerman and L. Jones. Social Browsing on Flickr. In *Proc. of ICWSM Conference*, March 2007.

[19] C. Lu, Y. Lu, T. F. Abdelzaher, J. A. Stankovic, and S. H. Son. Feedback control architecture and design methodology for service delay guarantees in web servers. *IEEE Transactions on Parallel and Distributed Systems*, 17(9):1014–1027, 2006.

[20] Y. Lu, W. Meng, L. Shu, C. Yu, and K. lup Liu. Evaluation of result merging strategies for metasearch engines. In *Proc. of WISE Conference*, New York, NY, Nov. 2005.

[21] M. Rabinovich and O. Spatscheck. *Web Caching and Replication*. Addison Wesley, 2002.

[22] A. Sang and S.-Q. Li. A predictability analysis of network traffic. In *Proc. of IEEE INFOCOM Conference*, Mar. 2000.

[23] S. Sivasubramanian, G. Pierre, M. van Steen, and G. Alonso. Analysis of Caching and Replication Strategies for Web Applications. *IEEE Internet Computing*, 11(1):60–66, 2007.

[24] W.-G. Teng, C.-Y. Chang, and M.-S. Chen. Integrating Web Caching and Web Prefetching in Client-Side Proxies. *IEEE Transactions on Parallel and Distributed Systems*, 16(5):444–455, 2005.

[25] M. Valafar, R. Rejaie, and W. Willinger. Beyond friendship graphs: a study of user interactions in Flickr. In *Proc. of the 2nd ACM Workshop on Online Social Networks (WOSN'09)*, Aug. 2009.

[26] T. Yamakami. A Zipf-Like Distribution of Popularity and Hits in the Mobile Web Pages with Short Life Time. In *Proc. of PDCAT Conference*, Taipei, Taiwan, 2006.

[27] R. Zhang, Y. Chang, Z. Zheng, D. Metzler, and J.-y. Nie. Search result re-ranking by feedback control adjustment for time-sensitive query. In *Proc. of Human Language Technologies Conference (HLT'09)*, Jun. 2009.