Fast recognition and application of Web users' behavioral patterns*

T. Chovaňák
Istituto Officina dei Materiali
P.O. Box 1212
Italy
gubbiotti@corporation.com

O. Kaššák
Dipartimento di Fisica e Geologia
P.O. Box 6221
Italy
malago@affiliation.org

M.Bieliková
Dipartimento di Fisica e Scienze
P.O. Box 5000
Italy
fin@affiliation.org

ABSTRACT

Behavioral patterns can be understood as typical and repeating features of user's behavior during their visit of website. In this work we represent behavioral patterns as frequent itemsets of actions frequently taken by user's in their sessions. Frequent source of knowledge about behavior of users are web logs and actions taken during their visits to website aggregated to sessions. Whole process of processing web logs, finding behavioral patterns and their analysis is also known as Web Usage Mining. Found behavioral patterns may be used to create recommendations, predict user's intentions (which can be used to cache predicted pages), as support for website design change or complex understanding of website users' behavior. Existing methods of Web Usage Mining usually search for behavioral patterns common for whole set of web site users in static web logs. This work responds to actual trend of Web personalization and focusing on needs of individual users and also challenge to mine knowledge from fast streaming data. We propose solution that is able to process data about user sessions as streaming data and search for behavioral patterns telling us not only about behavior of global community of users, but also about actual behavior and changes in behavior of smaller user communities. We evaluate contribution of combining global and group behavioral patterns in their application in recommendation task. We also observe way this method is able to detect unique behavior of specific groups of users in domain of e-learning system and newspapers web portal.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics • **Networks** → Network reliability

KEYWORDS

ACM proceedings, text tagging

1 INTRODUCTION

Understanding website users' behavior precisely is crucial for better personalization and adaptation of website content and structure. Every user is less or more different from others but often there are groups of users with similar behavior in some specific situations and time. Detecting groups of users with similar behavior and their behavioral patterns may lead to better understanding of users' intentions.

Web logs and users' actions stored there are often source of implicit information about users' behavior. Whole process of processing web logs, finding behavioral patterns and their analysis is known as Web Usage Mining (WUM). WUM is process consisting of 5 phases:

1. Collecting data from data sources
2. Preprocessing data
3. Patterns discovery
4. Analysis, validation and use of discovered patterns

One of main advantages of using WUM process to search for behavioral patterns is that data inputs consists of objective user feedback - actions taken by users. We can represent user session as set of actions taken by user (items) and behavioral patterns as frequent itemsets of actions gathered from preprocessed logs.

One of important challenges nowadays is fast in-memory processing of data generated as potentially infinite stream. Data streaming algorithms are built upon models that are incrementally updated with incoming data instances. Traditional data mining algorithms usually require more than one scan of all instances in database. In streaming data every instance should be processed only once. With data stream evolution conceptual drift can appear. Conceptual drift is important problem when performing data mining with streaming data. It is derived from changes in real environment where streaming data instances are created. Factors activating such changes are usually hidden and unknown. Conceptual drift is problem mainly with models created from old data before the drift happened which are consequently improper.

Goal of this work is to respond to current trends of Web personalization and focusing on needs of individual users by discovering behavioral patterns not only from global community od website users but also from smaller communities of users. We propose process with regard to requirements of data streaming algorithms, that is integrating data stream clustering algorithm used for segmenting active users to groups according their actual behavior with algorithm mining frequent closed itemsets over data stream that is able to respond quickly to conceptual drifts and is used to discover actual behavioral patterns represented as frequent closed itemsets from global community of users and from detected groups of users. Proposed method is able to detect recent behavior of global community and smaller communities and also changes of their behavior in time.

There are many applications of such discovered knowledge. For Web personalization as such, for predicting users' behavior, for caching web pages to memory, as important input when changing design of website or making some business decisions such e.g. about market segmentation.

We apply discovered knowledge about users' behavior in recommendation task and evaluate gain from combination of behavioral patterns for global community with patterns discovered for smaller communities against approach where we use only patterns discovered for global community in this specific task.

The paper is organized as follows. In §2, we review three related research directions. In §3, we describe method for mining global and group behavioral patterns over data stream and use them in recommendation task. In §4, we describe methodology of experiments and data we used to evaluate proposed method and simultaneously results we observed. We conclude the paper in §5.

2 Related Work

In this section we review few related research works. Our proposed process is combination of existing stream data mining algorithms from which every has its own research area. It's new application of WUM process that is able to run as whole in online environment. We apply extracted knowledge to recommend items to users, but there are other possible applications to it.

2.1 Existing applications of WUM to predict users' behavior and recommend items

Found knowledge about users' behavior with some of different representations of behavioral patterns can be applied to predict next steps of user or recommend items or products.

In [1] WebPUM method was proposed which represents behavioral patterns as partitions resulting from graph partitioning algorithm applied on user navigation graph, where nodes are web pages, connections between them have weight computed from time and frequency information of users' sessions. Weight of connection rises as more frequently two pages appear in sessions together and smaller time gap between them and more time users' spent on pages. They use gained patterns to recommend pages to users according their actual behavior.

Next interesting method predicting users' next steps was proposed in [2]. It is sophisticated system using fuzzy c-means clustering to find behavioral patterns represented as association rules.

In [3] 3 new contributions to enhance WUM process were proposed. First is heuristic to identify users' sessions. Second is usage of DBScan algorithm to cluster users' sessions. DBScan is able to reveal otherwise ignored patterns because of their low support but high confidence when represented as association rules. Their last proposed contribution is using inverted index to effectively predict users' behavior online.

2.2 Algorithms for mining frequent closed itemsets over data stream

Our method represents behavioral patterns as closed frequent itemsets extracted from fast streaming data. Several algorithms were proposed for this task. We focused on algorithms mining only frequent closed itemsets which are complete and not redundant representation of all frequent itemsets. That significantly reduces size of search space.

Existing algorithms can be classified according to window model they use. It could be landmark window containing all items from start of the stream or sliding window containing only most recent elements. Algorithms could be mining exact set of frequent itemsets or approximate set of frequent itemsets. Approximate mining can be much more effective because it doesn't have to track all itemsets (frequent and not frequent) in history and is able to respond to conceptual drift well.

First algorithm for incremental mining of closed frequent itemsets over a data stream was MOMENT, proposed in [4]. It mines exact frequent itemsets using sliding window approach. It has become a reference for solutions proposed later. They propose in-memory prefix-tree-based data structure called closed enumeration tree which effectively stores information about infrequent itemsets, nodes that are likely to become frequent, and closed itemsets.

Algorithm NEWMOMENT proposed in [5] makes MOMENT more efficient. It represents itemsets and window as bitsets. This representation allows usage of efficient bitwise operations for example to count support of itemsets or perform sliding of window with bitwise shift.

CLOSTREAM proposed in [6] uses different data structures and approach to mine exact closed frequent itemsets over sliding window than MOMENT.

Abandoning requirement to mine exact frequent itemsets helps to design fast algorithm for mining approximation of frequent closed itemsets like that proposed in [7] called IncMine. They use relaxed minimal support threshold to keep infrequent itemsets that are promising to become frequent later. They use update per batch policy that is different to all other algorithms we described here. It results in better time-per-transaction at risk of temporarily loosing accuracy of the maintained set while each batch is being collected [8]. In [8] They use inverted index to efficiently address stored itemsets with IncMine algorithm.

Next algorithm named CLAIM for approximate frequent closed itemsets mining was proposed in [9]. This algorithm is trying to solve problem when conceptual drifts appear frequently and slows down algorithm by redefining frequent itemset definition and proposing usage of support value intervals considered as same value.

2.3 Algorithms for clustering over data stream

Our method clusters user models into groups according their similar behavior. We adapt approach proposed in [10] where authors introduced Clustream framework for clustering over data stream. It is based on online microclustering component performing fast transformation of incoming data instances into compact approximate statistical representation and offline component using this representation to perform macroclustering and get results of clustering on demand.

3 Method for mining personalized behavioral patterns over data stream

In this section, we describe method for mining behavioral patterns common to global community of users and behavioral patterns common to dynamically identified groups of users over data stream and their application in recommendation task.

3.1 Input data format

We consider one data instance in data stream as one user's session represented as set of actions user made identified by unique identifiers. Actions could be diverse like etc. visits of webpages, buying of products, adding or removing items to or from shopping basket in e-shop.

3.2 User model representation

Important part of designed process is clustering of users to groups according their similar recent behavior. As data stream could be potentially infinite we need to prevent possible memory leak caused by adding new user models to memory and not deleting old. We represent users' behavior simply as different frequency spectrums of their recent actions. We use queue with limited capacity to store this actions. Capacity of this queue is one of input parameters. As we said we use Clustream framework for clustering user models consisting of fast update of microclusters part and macroclustering part where k-means or other traditional clustering algorithm can be applied to found final clusters. Macroclustering is performed on regular basis. Every macroclustering has its id number assigned.

User model *u* consists of this fields:

- *uid* : user identifier,
- *aq*: actions queue. Queue with limited capacity to store actions user took in recent history.
- *gid*: group identifier. Id of group where user was classified to in last macroclustering.
- *nsc*: new sessions count. Number of new sessions of this user since last macroclustering.
- *lmid*: last macroclustering id. Identifier of last macroclustering user model when this user model was active.

Let current macroclustering id be *lmid* (last macroclustering id). User *u* is assigned new group identifier on his first session after new macroclustering was performed. So only if his *u.lmid* is other than *lmid*. Always after new macroclustering is performed every user model *u* where *lmid* - *u.lmid* > *tcdiff*, where *tcdiff* (threshold of clustering identifiers difference) is input parameter, are deleted to prevent memory leak.

3.3 Behavioral pattern representation

We represent behavioral patterns as frequent closed itemsets with computed support value. We discriminate between global and group behavioral patterns. Global behavioral patterns are mined from behavior of whole community of users. Group behavioral patterns are patterns mined from behavior of concrete segment of community of users identified with clustering part of designed process.

3.4 Application of behavioral patterns

As we said found behavioral patterns could be applied as input to diverse other tasks. We apply found behavioral patterns to recommend pages to users in session. Let actual session be represented as vector of user actions $S = < a_1, a_2, \dots, a_n >$. Found behavioural patterns are $P = \{P_1, P_2, \dots P_m\}$. Each pattern P_i is represented as set of actions $P_i = \{a_1, a_2, \dots, a_k\}$. Let *ews* be size of evaluation window part of session. Condition $ews > 0 \wedge ews < |S|$ must be true to use actual session for evaluation of recommendation. Let first *k* actions of *S* be dedicated as evaluation window $W_e = < a_1, a_2, \dots, a_k >$ and all other actions from *S* as testing part $T_e = < a_{k+1}, \dots, a_n >$. Let *r* be number of recommended items. If condition $n >= (ews + r)$ is not met, then *S* is ignored for evaluation. We use following strategy to choose behavioral patterns according to actual evaluation window:

1. For each P_i in P approximate support value is computed. Let mark it as $support(P_i)$. Intersection of P_i with W_e is found with LCS algorithm (Least common subset). Let's mark it as $lcs(P_i, W_e)$.
2. All patterns (global and group) are sorted. First by size of intersection with We descending. Second by support value descending.
3. Let *M*, be map of items and their *"votes"*. By iterating over all patterns votes values of items are updated. Votes value of item *i* that is contained in pattern P_i and not in W_e is incremented by $support(P_i) * lcs(P_i, W_e)$ both normalized to *<0,1>* interval.
4. Finally, *M* is sorted descending by votes values and best *r* items are picked to be recommended to user.

3.4 User session processing

In this section we describe how every user session is processed. On Figure 1 is activity diagram for this process. This process is designed with regards to evaluation and experiments we perform on it. It is framework where individual components can be changed (etc. different clustering or frequent closed itemsets mining algorithm). User session represented as set of actions is loaded into process from data stream. First it is used in recommendation task as we proposed in previous section and other evaluation purposes as we describe them later. Next user model *u* is updated. Actions from current session are added to queue in user model and *u.nsc* (new sessions count) counter is incremented. If *u.nsc* is greater than input parameter *tcu* (threshold number of changes in user model) then *u.nsc* is nulled and microclusters are updated with instance generated from *u.aq* (user model's actions queue) and *u.muc* (microclusters updates counter) is incremented by 1. If number of updates in microclusters (*muc*) is greater than given threshold *tcm* (threshold number of changes in microclusters) then *muc* is nulled and macroclustering is performed. With macroclustering, *lmid* (last macroclustering id) is incremented and every old user model *u*, meeting condition that $lmid - u.lmid > tcdiff$, (where *tcdiff* parameter is threshold of clustering identifiers difference) is deleted from memory. Next if user model *u* has *u.lmid* attribute value other than current global *lmid* then *u.gid* is assigned identifier of group he belongs to according to last macroclustering performed. Lastly user session with appended id of group user belongs to is input to algorithm for mining frequent closed itemsets for specific group and same user session alone is input to same algorithm but performed for all users.

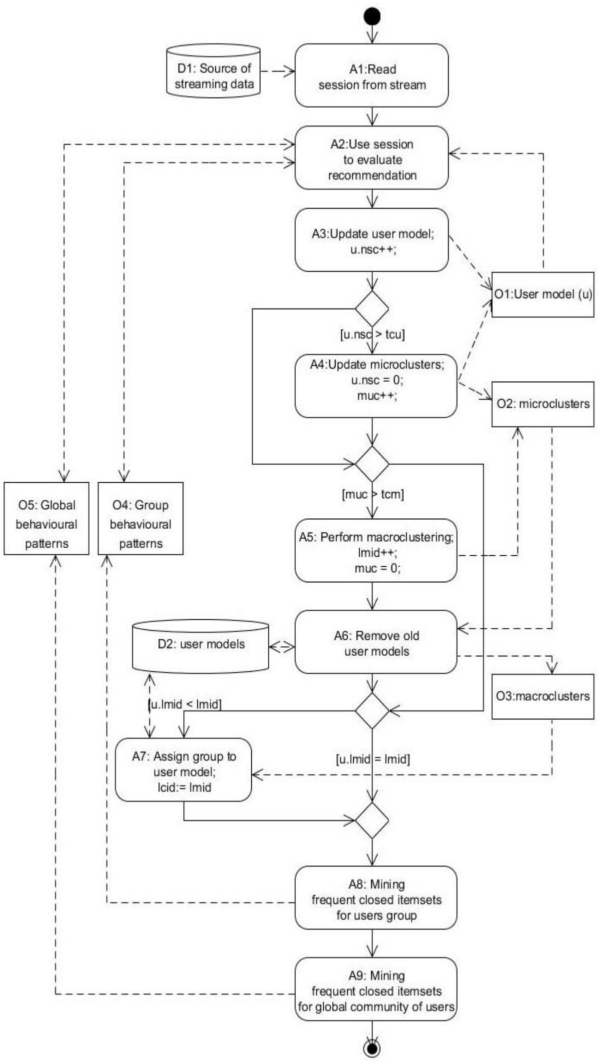

Figure 1: Activity diagram displaying processing of user sessions. Prefix A means action, prefix O means data object, prefix D means data source.

Some parts of designed process could be parallelized. Data stream could be copied to 3 separate branches. One branch performing clustering part of process, second branch performing mining of group behavioral patterns and third branch performing mining of global behavioral patterns. For our evaluation purposes sequential process was used. But in real application parallelization could be easily adapted to perform processing even faster.

3.4 Speed regulation

As mentioned in [11] task of mining frequent itemsets requires balancing requirements for accuracy and effectivity. Higher speed means worse accuracy and better accuracy means lower speed. User should have option to set this balance according to his actual needs. In our method we give to user option to set minimal required speed as input parameter *mts* (minimal transactions per second). As we mentioned, *IncMine* algorithm processes transactions in batches. Every update of batch (segment) is critical part. Let *actspeed* be actual average speed of processing in transactions per second. Let *ctrans* be actual number of processed transactions from start of measuring. Then:

$$actspeed = \frac{ctrans}{mts} \tag{1}$$

Let *tstart* be time when measuring started. Let *tsupdate* be time when update of batch started. We compute maximal allowed time of update *tmax* before each update as:

$$tmax = actspeed - tsupdate + tstart \tag{1}$$

When frequent itemsets update in current batch takes more than *tmax* it is simply stopped. It means some frequent closed itemsets won't be discovered. In next section we evaluate what effect does it have on accuracy when changing *mts* parameter value. Interesting solution to this problem would be using other algorithm for mining frequent closed itemsets from batch of transactions with approach of mining *k* most frequent itemsets because there is no need to set minimal support threshold like one proposed in [12] and best patterns are discovered first. Implementing and evaluating this kind of solution is actually outside of scope. As we see in evaluation part high dimensionality is problem that primarily causes speed deceleration. But with moderate number of different possible user actions logged solution is very fast.

3.4 Summary of parameters used

One of critical parts of designed process is number of input parameters. There are input parameters for clustering algorithm and for mining frequent closed itemsets algorithm and some parameters required by process itself. We divided these parameters into four categories according their purpose. Summary of parameters and categories is in Table 1. We use *IncMine* algorithm as implemented in [13] for mining frequent closed itemsets over data stream. It has its own parameters. Minimal support is value corresponding to σ parameter in *IncMine* algorithm that is used to compute progressive function of minimal support (MST) for different segments of actual window. Relaxation rate is value corresponding to *r* parameter in *IncMine* algorithm that is used to compute relaxed MST which prevents from deleting potentially frequent itemsets. Segment length is number of transactions in one batch update. Window size is number of segments sliding window consists of. We use

CluStream framework as implemented in MOA framework. Important input parameter is desired number of clusters. For clustering also other threshold parameters are important we already mentioned in process description (*tcu*, *tcm*).

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
$	4 in 5	Used in business

.4.2 Micromagnetic. For each micromagnetic cell the reduced magnetization takes the form where the magnetization (saturation magnetization) in the *k*-th cell; note that the saturation magnetization now depends on the ferromagnetic material through the index *k*. Hence, in a polar reference frame

$$(x+a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k} \tag{1}$$

where *K* is the azimuthal (polar) angle of the magnetization (the time dependence is omitted). The second derivatives of the energy density depend on the micromagnetic cell indexes, and through them on the material index corresponding either to Py or Co. The expressions of E_{ext}, E_{exch}, E_{dmg} and E_{ani} are the same as the ones of the single-component system apart from the explicit dependence of the magnetic parameters on the given ferromagnetic material. Moreover, the uniaxial anisotropy energy density of Co is neglected.

It is possible to write the following periodicity rule valid for the dynamic magnetization $\delta m(r)$ of each collective mode, a version of the Bloch theorem, viz. Note that, exchange contribution is set equal to zero, because in each unit cell the two elliptical dots are separated. Moreover, the uniaxial anisotropy energy density of Co is neglected Table 1.

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
$	4 in 5	Used in business

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2}$$

Therefore one can observe either an in-phase (acoustic) or an out-of-phase (optical) character of the modes, with respect to the precession of the in-plane magnetization components in adjacent Py and Co dots.

We would like to mention that the DMM presents several advantages with respect to OOMMF for calculating the spectrum of magnetic eigenmodes for the following reasons: *a*) There is no need to excite the system by any magnetic field pulse, *b)* A single calculation llows to determine the frequencies and eigenvectors of all spin-wave modes of any symmetry, *c*) The spectrum is computed directly in the frequency domain, *d*) The mode degeneracy is successfully solved, *e*) The spatial profiles of the spin-wave modes are directly determined as eigenvectors and, finally, *f*) The differential scattering cross-section can be calculated accurately from the eigenvectors associated to each spin-wave mode. This is a clear indication that both the Py and Co sub-elements are in a single domain state where Py and Co magnetizations are all oriented with their magnetic moment along the chain and field direction. At point β ($H = -372$ Oe) of the hysteresis loop, where the plateau is observed in the *M-H* loop, the dark and bright spots of the Py dots are reversed with respect to those of Co, accounting for an antiparallel relative alignment of magnetization.

3 RESULTS AND DISCUSSION

3.1 Magnetization Curves and MFM Characterization

The major hysteresis loop measured by MOKE, plotted in Fig. 1, displays a two-step switching process due to the distinct magnetization reversal of the Py and Co sub-elements, characterized by a different coercivity. As the field is reduced from positive saturation (upper branch of the M-H loop), a 100% remanence is attained. Within each bi-component unit (about 36%) in good agreement with experimental result (about 40%).

To directly visualize the evolution of the magnetization in the Py and Co subunits of our bi-component dots during the reversal process, we performed a field-dependent MFM analysis whose main results are reported in Fig. 2. At large positive field ($H = +800$ Oe, not shown here) and at remanence (α point of the hysteresis loop of Fig. 1), the structures are characterized by a strong dipolar contrast due to the stray fields emanated from both the Py and Co dots.

Figure2 : MFM images of the bi-component Py/Co dots for different values of the applied magnetic field which are indicated by greek letters along both the major and minor hysteresis loop.

This is a clear indication that both the Py and Co sub-elements are in a single domain state where Py and Co magnetizations are all oriented with their magnetic moment along the chain and field direction. At point β ($H = -372$ Oe) of the hysteresis loop, where the plateau is observed in the *M-H* loop, the dark and bright spots of the Py dots are reversed with respect to those of Co, accounting for an antiparallel relative alignment of magnetization.

At relatively large negative fields (point γ, $H = -770$ Oe) the magnetization reversal is completed and the magnetization of the two adjacent sub-elements are saturated in the negative direction. The ground state remains unchanged when the field is now reduced to zero, i.e. remanent state coming from negative saturation, as confirmed by the MFM image taken at point δ of Fig. 1.

We have also used MFM to measure the magnetic configurations along the minor hysteresis loop, described above. Once the AP ground state has been generated at $H = -500$ Oe, the applied field is increased in the positive direction. The MFM image taken at point α' of Fig. 2, remanent state of the minor loop ($H = 0$), shows that the AP state is stable and remains unchanged until the magnetic field is increased up to +300 Oe where the Py magnetization reverses its orientation and returns to be aligned with that of Co dots. On the basis of the above MFM investigation, one can say that the structures are always in a single domain state, while the relative magnetization orientation between the adjacent Py and Co elements depends on both the field value and the sample history.

3.2 Field Dependent BLS Measurements and DMM Calculations

Fig. 3 displays the frequencies of BLS peaks plotted as a function of the applied field magnitude starting from positive values. The field is then decreased and reversed following the upper branch of the hysteresis loop, shown in the same figure. Up to five peaks are measured in the spectra, as shown in spectrum measured at H = 0 Oe in the Fig. 3 inset, and their field evolution analyzed over the whole field range investigated. The detected modes are identified and labeled on the basis of their calculated spatial profiles, shown in Fig. 4 for $H = 500$ and -500 Oe.

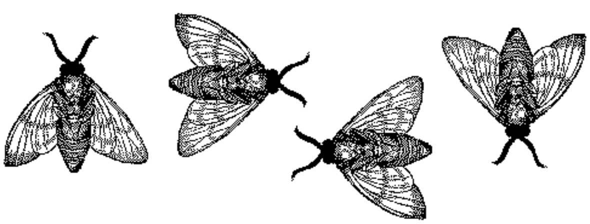

Figure3 : Dependence of the magnetic eigeinmode wave frequency on the applied field strength.

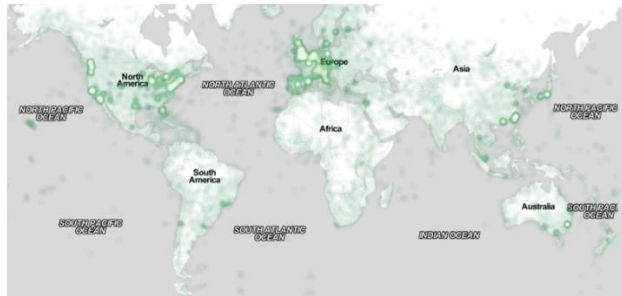

Figure4 : Calculated spatial distribution of the in-plane dynamic magnetization.

They exhibit marked localization into either the Co or the Py dots, as stated at the end of the previous Section, were it was introduced the labelling notation containing the dominant localization region (either Py or Co) and the spatial symmetry (EM, F, DE, etc).

When the dots are in the P state, up to five modes were detected in BLS spectra. On the basis of the calculated profiles (right panel of Fig. 4), we identified in the P state the two modes at lowest frequencies as the EM(Py) and the F(Py), with a very small spin precession amplitude into the Co dot. This is because for this material we are below the frequency threshold for the existence of spin waves. A similar effect has been observed in periodic array of alternating Permalloy and Co nanostripes

Note that the nodal lines present in the spatial profile of the F (Co) mode perpendicular to the long axis of the ellipse do not correspond to a real change of sign of the dynamic magnetization and are due to the partial hybridization of the F mode with higher-order modes having frequencies close to the one of the F mode. Interestingly, the frequency slope of modes localized into the Co dots is larger than that of Py modes, due to larger values of the Co magnetization and gyromagnetic ratio. An overall good agreement between the calculated (dotted curves) and measured frequency (full points) has been achieved (see Fig. 3) even if some discrepancies are observed for the frequency of the EM and 1DE (Py) modes.

The corresponding spatial profiles of the modes are shown in the left panels of Fig. 4. Here one can see that the only mode which is purely localized in one dot is the EM of Co, because now it is sub-threshold for Py. A further reduction of H, which is sufficient to cause the Co magnetization reversal, produces a P state at negative fields and the frequency starts to increase.

Figure5 : Full point are the frequencies measured along the minor hysteresis.

Again as a function of the applied field. In this field range the frequencies of modes in the Py dots monotonously increase in a way similar to that measured in the P state for positive field values while an abrupt change in the frequency of Co modes occurs.

Notice that if one stops increasing the negative field to about –300 Oe and comes back towards positive applied fields, BLS measurements can be performed following the minor hysteresis loop. This method permits to study, for example, the magnetization dynamics at remanence (without any external applied magnetic field) when the system is in the AP state (see MFM image α' in Fig. 2), a configuration which cannot be achieved at remanence along the major M-H loop. In Fig. 5 we show the modes frequency measured along the minor loop (full points) and compare them with values measured along the major M-H loop (open points).

By inspection of the frequency slope of the modes, one can immediately understand the localization of modes into dots of different materials looking at their slope.

In particular, for three (two) modes we measure a negative (positive) frequency slope with an almost linear dependence on H. It is evident that modes with negative frequency slope are modes localized into the Py dot (EM, F and 1DE) while the two with positive slope are the F(Co) and the EM(Co) modes.

3.3 Analysis of the Dynamic Coupling as a Function of the Gap Size

One interesting point which emerges from analysis of Figs. 3 and 4 is that the frequency values of the eigenmodes are not the same at +500 Oe and at –500 Oe. This is expected for modes localized into the Co elements, since the external field is either parallel or antiparallel to their magnetization. However, for those mode localized into the Py sub-element. One could have predicted to find the same frequency values at ±500 Oe, unless the dipolar coupling arising from the adjacent Co dot plays a significant role. In fact, as seen in Figs. 3 and 4, reversing the field from +500 to –500 Oe, the frequencies of EM(Py) and 1DE(Py) modes increase by about 0.2 GHz and 0.6 GHz, respectively, while that of F(Py) decreases by 0.25 GHz. The reason of this complex behavior will be addressed in the following, analyzing the interplay of both static and dynamic dipolar coupling between the adjacent Py and Co dots Table 2.

This is a clear indication that both the Py and Co sub-elements are in a single domain state where Py and Co magnetizations are all oriented with their magnetic moment along the chain and field direction.

Table 2: Comparison of Coefficients from Atomistic

Atm	MS-CG	MS-CG/DPD
1.78	14.32	1.74 (–2%)
0.43	31.00	0.40 (–7%)
0.062	15.61	0.048 (–23%)
0.032	9.76	0.024 (–24%)
0.020	4.66	0.015 (–25%)
0.012	2.32	-"-
0.0076	0.016	-"-

Figure6 : Calculated frequency evolution of modes detected in the BLS spectra.

In Fig. 6 the calculated frequencies of the most representative eigenmodes at +500 Oe (FM state) and – 500 Oe (AP state) are plotted as a function of the gap size d between the Py and Co sub units (please remind that in the real sample studied here, d = 35 nm). As a general comment, it can be seen that the frequencies for the system in the AP state are more sensitive to d than those of the P state. In particular, the lowest three frequency modes of the AP state (EM(Co), EM(Py) and F(Py)) are downshifted with respect to the case of isolated elements (dotted lines) and show a marked decrease with reducing d, while the two modes at higher frequencies (F(Co) and 1DE(Py)) have an opposite behavior even though they exhibit a reduced amplitude. In the P state (right panel), the modes concentrated into the Py dots exhibit a moderate decrease with reducing d, while an opposite but less pronounced behavior is exhibited by the F(Co) mode.

1. Never, ever use vertical rules.
2. Never use double rules.

4 CONCLUSIONS

In summary, we have performed both an experimental and theoretical study of the spin eigenmodes in dipolarly coupled bi-component cobalt and permalloy elliptical nanodots. Several eigenmodes have been identified and their frequency evolution as a function of the intensity of the applied magnetic field has been measured by Brillouin light scattering technique, encompassing the ground states where the cobalt and permalloy dots magnetizations are parallel or anti-parallel, respectively. In correspondence to the transition between the two different ground states, the mode frequency undergoes an abrupt variation and more than that, in the anti-parallelstate, the frequency is insensitive to the applied field strength. The experimental results have been successfully interpreted by the dynamic matrix method which permits to calculate both the mode frequencies and the spatial profiles.

A HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are di.erent in the appendices. In the appendix environment, the command section is used to indicate the start of each Appendix, with alphabetic order designation (i.e., the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure within an Appendix, start with subsection as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 Experimental and Computational Details

A.2.1 Sample Fabrication

A.2.2 Quasi-Static Measurements: MOKE and MFM

Component Structures

Magnetization.

A.2.3 Dynamic Measurements: BLS

A.2.4 Ground-State Magnetization Determination and DMM Micromagnetic Simulations

Determined.

Micromagnetic

A.3 Results and Discussion

A.3.1 Magnetization Curves and MFM Characterization

A.3.2 Field Dependent BLS Measurements and DMM Calculations

A.3.3 Analysis of the Dynamic Coupling as a Function of the Gap Size

A.4 Conclusions

A.5 References

ACKNOWLEDGMENTS

This work was partially supported by the MIUR-PRIN 2010–11 Project 2010ECA8P3 “DyNanoMag” and by the National Research Foundation, Prime Minister's office, Singapore under its Competitive Research Programme (CRP Award No. NRF-CRP 10-2012-03).

REFERENCES

[1] Jalali, M., Mustapha, N., Nasir Sulaiman, M. and Mamat, A.2010. WebPUM: A Web-based recommendation system to predict user future movements Expert Systems With Applications, 37, 6201–6212. DOI: http://dx.doi.org/10.1016/j.eswa.2010.02.105

[2] Liraki Z. and Harounabadi A. 2015. Predicting the Users’ Navigation Patterns in Web, using Weighted Association Rules and Users’ Navigation Information. International Journal of Computer Applications. 110.12, 16-21.

[3] Anandhi, D., and MS Irfan Ahmed. "An Improved Web Log Mining and Online Navigational Pattern Prediction." Research Journal of Applied Sciences, Engineering and Technology 8.12 (2014): 1472-1479.

[4] Chi, Yun, et al. "Moment: Maintaining closed frequent itemsets over a stream sliding window." Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on. IEEE, 2004.

[5] Li, Hua-Fu, et al. "A new algorithm for maintaining closed frequent itemsets in data streams by incremental updates." Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06). IEEE, 2006.

[6] Yen, Show-Jane, et al. "A fast algorithm for mining frequent closed itemsets over stream sliding window." Fuzzy Systems (FUZZ), 2011 IEEE International Conference on. IEEE, 2011.

[7] Cheng, James, Yiping Ke, and Wilfred Ng. "Maintaining frequent closed itemsets over a sliding window." Journal of Intelligent Information Systems 31.3 (2008): 191-215.

[8] Quadrana, Massimo, Albert Bifet, and Ricard Gavalda. "An efficient closed frequent itemset miner for the MOA stream mining system." AI Communications 28.1 (2015): 143-158.

[9] Song, Guojie, et al. "CLAIM: An efficient method for relaxed frequent closed itemsets mining over stream data." International Conference on Database Systems for Advanced Applications. Springer Berlin Heidelberg, 2007.

[10] Aggarwal, Charu C., et al. "A framework for clustering evolving data streams." Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003.

[11] Lee, Victor E., Ruoming Jin, and Gagan Agrawal. "Frequent pattern mining in data streams." Frequent Pattern Mining. Springer International Publishing, 2014. 199-224.

[12] Tzvetkov, Petre, Xifeng Yan, and Jiawei Han. "TSP: Mining top-k closed sequential patterns." Knowledge and Information Systems 7.4 (2005): 438-457.

[13] Quadrana, Massimo, Albert Bifet, and Ricard Gavalda. "An efficient closed frequent itemset miner for the MOA stream mining system." AI Communications 28.1 (2015): 143-158.