OPEN MULTI-PROCESSING ACCELERATION FOR UNSUPERVISED LAND COVER CATEGORIZATION USING PROBABILISTIC LATENT SEMANTIC ANALYSIS

S. Bernabé[1], *C. García*[1], *R. Fernández-Beltrán*[2], *M. E. Paoletti, Student Member, IEEE*[3], *J. M. Haut, Student Member, IEEE*[3], *J. Plaza, Senior Member, IEEE*[3], and *A. Plaza, Fellow, IEEE*[3]

[1]Department of Computer Architecture and Automation, Complutense University, 28040 Madrid, Spain.
[2]Institute of New Imaging Technologies, University Jaume I, 12071 Castellón, Spain.
[3]Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications,
Escuela Politécnica, University of Extremadura, E-10003 Cáceres, Spain.

ABSTRACT

The probabilistic Latent Semantic Analysis (pLSA) model has recently shown a great potential to uncover highly descriptive semantic features from limited amounts of remote sensing data. Nonetheless, the high computational cost of this algorithm often constraints its operational application for land cover categorization tasks. In this scenario, this paper presents an Open Multi-Processing (OpenMP) implementation of the pLSA algorithm for unsupervised Synthetic Aperture Radar (SAR) and Multi-Spectral Imaging (MSI) image categorization. The experimental results suggest that multi-core systems are an important architecture for the efficient processing of both SAR and MSI datasets. Specifically, the proposed approach is able to cover a real scenario exhibiting good results in both accuracy and performance terms.

Index Terms— Open Multi-Processing (OpenMP), multi-core processors, probabilistic Latent Semantic Analysis (pLSA), land cover categorization.

1. INTRODUCTION

Over the past years, unsupervised land cover categorization [1] has shown to play an important role within the remote sensing community to cope with different Earth monitoring challenges and needs [2]. Whereas traditional clustering-based categorization approaches are often unable to deal with the complex nature of airborne and space optical imagery [3], the generative framework provided by the probabilistic Latent Semantic Analysis (pLSA) model [4] has recently shown a great potential to uncover high-level feature patterns to effectively categorize remote sensing Synthetic Aperture Radar (SAR) and Multi-Spectral Imaging (MSI) data [5, 6, 7].

Nonetheless, the high computational cost of the pLSA algorithm [8] often constraints its practical use in operational remote sensing scenarios, especially under challenging data volume processing requirements [9]. As a result, more research work is required to improve pLSA efficiency within the remote sensing domain.

In the literature, it is possible to find few works that exploit some parallelism mechanisms for text analysis using the pLSA model [10, 11]. However, processing operational remotely sensed data using parallel architectures faces some technical challenges that motivate this research [12]. Accordingly, this paper proposes a multi-core pLSA implementation specially designed for unsupervised land cover categorization tasks using the OpenMP API. More specifically, our implementation is based on guided-vectorization and OpenMP directives to accelerate the land cover categorization process. This experimental study reveals that Xeon multi-core processors can provide significant speedup factors maintaining similar accuracy with respect to the baseline version, using real SAR and MSI datasets.

2. METHODOLOGY

2.1. Probabilistic Latent Semantic Analysis

The pLSA model [13] defines a probabilistic generative data process which is performed as follows: (1) selecting a document d with probability $p(d)$; (2) picking a hidden class z according to the conditional probability $p(z|d)$; (3) generating a word w with probability $p(w|z)$. Accordingly, given the observed data distribution $p(w|d)$, which describes a corpus of documents $D = \{d_1, d_2, ..., d_M\}$ in a particular word-space $W = \{w_1, w_2, ..., w_N\}$, the pLSA model estimates two probability distributions, the description of topics in words $p(w|z)$ and the description of documents in topics $p(z|d)$.

In this work, the $p(w|z)$ and $p(z|d)$ model parameters are estimated by maximizing the complete log-likelihood presented in Eq. (2), where $n(w, d)$ represents the observable document-word counts and K is the total number of topics. Specifically, we use the Expectation-Maximization (EM) algorithm [14] which work in two stages: (i) E-step, where the likelihood expected values are estimated and (ii) M-step,

This work was supported in part by Spanish Ministry under Grant FPU14/02012-FPU15/02090 and Grant ESP2016-79503-C2-2-P, in part by the Junta de Extremadura under Grant GR15005, in part by EU (FEDER), in part by the Spanish MINECO under Grant TIN 2015-65277-R, and in part by Generalitat Valenciana under Grant APOSTD/2017/007.

978-1-5386-9154-0/19/$31.00 ©2019 IEEE

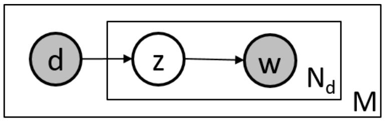

Fig. 1. The pLSA model graphical representation.

where the new optimal values for the model parameters are calculated. The E-step can be computed using the Bayes' rule and the chain rule as Eq. (2) shows. In the case of the M-step, we compute the pLSA likelihood partial derivatives, set them as equal to zero and solve the equations to obtain Eqs. (3)-(4).

$$\ell_c = \sum_{d}^{D} \sum_{w}^{N} n(w,d) \log \left(p(d) \sum_{z}^{K} p(w|z) p(z|d) \right) \qquad (1)$$

$$p(z|w,d) = \frac{p(z,w,d)}{p(w,d)} = \frac{p(z,w,d)}{\sum_{z} p(w,d)} = \frac{p(w|z)p(z|d)}{\sum_{z} p(w|z)p(z|d)} \qquad (2)$$

$$p(w|z) = \frac{\sum_{d} n(w,d)p(d)p(z|w,d)}{\sum_{w}\sum_{d} n(w,d)p(d)p(z|w,d)} \qquad (3)$$

$$p(z|d) = \frac{\sum_{w} n(w,d)p(z|w,d)}{\sum_{z}\sum_{w} n(w,d)p(z|w,d)} \qquad (4)$$

The pLSA process is performed as Algorithm 1 shows. First, $p(w|z)$ and $p(z|d)$ are randomly initialized. Then, the E-step [Eq. (2)] and the M-step [Eqs. (3)-(4)] are alternated until $p(w|z)$ and $p(z|d)$ parameters converge. As default convergence conditions, we use a 10^{-6} stability threshold in the log-likelihood and a maximum number of 2000 EM iterations.

Algorithm 1: EM algorithm for pLSA.

```
input: $n(w,d)$, $K$
$I = 0$; $T = \infty$; $L = 0$;
$p(w|z)$, $p(z|d)$ random initialization;
while ($I < 2000$) and ($T > 10^{-6}$) do
    E-step: $p(z|w,d) \Leftarrow$ Eq. (2);
    M-step: $p(w|z)$, $p(z|d) \Leftarrow$ Eqs. (3)-(4);
    $\ell_c \Leftarrow$ Eq. (1); $T = \ell_c - L$; $L = \ell_c$; $I = I + 1$;
end
```

2.2. Unsupervised Land Cover Categorization Framework

The considered pLSA-based land cover categorization framework consists of the following three steps (Fig. 2):

(i) Image characterization: first, we use the visual-bag-of-words (vBoW) approach [15] to apply pLSA over remotely sensed optical data. The input images are initially tiled into 32×32 image patches to define topic model documents (d). Then, the k-means clustering algorithm [16] is used to build the visual vocabulary considering vectorized 3×3 image patches as local primitive features and 50 clusters. Finally, the local primitive features of each topic model document are encoded in a single histogram of visual words by accumulating the number of local features into their closest clusters. From this characterization step, we obtain a collection of M documents $D = \{d_1, d_2, ..., d_M\}$ described in a 50-word visual vocabulary, i.e. $d_i = \{n(w_j, d_i)\} \forall j \in \{1, 2, ..., 50\}$.

(ii) pLSA modeling: second, we use the pLSA algorithm (Algorithm 1) to estimate both $p(w|z)$ and $p(z|d)$ model parameters by considering K topics, which is set to the number of ground-truth image categories.

(iii) Unsupervised land cover categorization: third, each document is categorized according to its dominant topic, that is, the highest probability value in $p(z|d)$ ($\arg\max_k p(z_k|d)$) provides the Earth surface categorization we use as land cover prediction.

3. PARALLEL IMPLEMENTATION

In this section, a parallel implementation of the aforementioned methodology is described. Our implementation is based on two levels of parallelism: multi-threading using the OpenMP paradigm and SIMD (single-instruction, multiple-data) by means of guided-vectorization. OpenMP parallelization has been carried out on the external loops while, for SIMD exploitation, the internal loops were chosen.

Guided vectorization allows the Intel compiler to identify and optimizes code fragments exploiting SIMD parallelism. In order to enable compiler guided vectorization, memory pointer disambiguation is enabled with the use of the *restrict* tag. Due to the fact that reduction operations usually inhibit auto-vectorization by potential data dependencies, reduction fragments have been rewritten using temporal local variables, and the pragma *omp parallel for simd* is also added to force vectorization when the compiler is not able to detect potential SIMD parallelism.

In this work, an efficient serial C (baseline version) was developed from a Python implementation. The layouts of the arrays have been selected according to their pattern access in

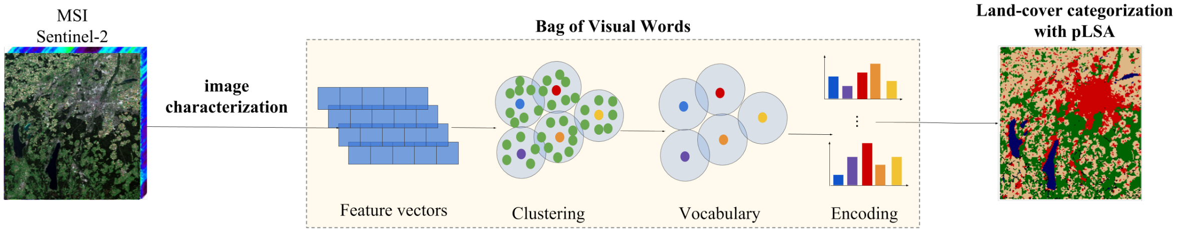

Fig. 2. The land cover categorization framework is feeded by an image, and after apply an image characterization implemented by a vBoW, the land cover categorization is performed with pLSA.

order to exploit the memory hierarchy efficiently. In addition, data structures were aligned in memory.

We briefly summarize the main optimization techniques used for the pLSA algorithm in the following items:

1. For the E-step, *#pragma omp parallel for* directives are used to achieve the multi-threading version. It is important to note that the most costly part corresponds to the computation of $\sum_z p(w|z)p(z|d)$ [Eq.(2)]. As we need to check the denominator in order to avoid zero division, the last loop iteration was isolated. This loop-split strategy not only favors multi-threading exploitation by avoiding non-zero checking, but also improves the SIMD extraction. At this point, it is important to note that it is also possible to configure the work-load distribution among the threads by using the schedule clause. As each *for loop* iteration is nearly constant, static scheduling has been adopted in our implementation.

2. For the M-step, all the optimization techniques described above have been performed: multi-threading expression by means of *#pragma omp parallel for* directives, use of local variables in reductions, explicit use of the *omp parallel for simd* pragmas, loop-split to avoid zero division in the denominator of Eqs. (3)-(4) and static scheduling policy selection.

4. EXPERIMENTAL RESULTS

4.1. Dataset

The Munich [5] dataset has been considered in this work. Specifically, this remote sensing collection includes a Sentinel-1B (SAR) and a Sentinel-2A (MSI) operational product of Munich (Germany), acquired on September 29 and 30, 2016. The data used in this work is available from the German Earth Observation Center website (`http://goo.gl/ma9dUt`) where ground-truth land-cover information ('Agriculture', 'Building', 'Forest' and 'Water') is also accessible for assessment purposes. Accordingly, the number of topics (K) has been fixed to 4 for the unsupervised land cover categorization experiments.

4.2. Accuracy Evaluation

Table 1 provides a quantitative evaluation of the unsupervised land cover categorization results for Munich dataset in terms of accuracy, precision, recall and f-score metrics. In particular, ground truth image categories are shown in rows and Sentinel-1 (SAR) and Sentinel-2 (MSI) results are presented in columns. It should be noted that Table 1 reports the average percentage and the corresponding standard deviation obtained after five runs of the indicated algorithms, pLSA or OpenMP-pLSA.

	CATEGORY	MUNICH			
		SENTINEL-1 (SAR)		SENTINEL-2 (MSI)	
		pLSA	OpenMP-pLSA	pLSA	OpenMP-pLSA
ACCURACY	Agriculture	81.51±0.03	81.51±0.03	78.13±0.03	78.56±0.54
	Forest	74.58±0.14	74.4±0.12	92.3±0.07	92.38±0.15
	Building	85.79±0.14	85.62±0.13	79.92±0.06	81.98±2.54
	Water	99.33±0.0	99.33±0.0	99.59±0.01	97.24±2.88
	AVG	85.3±0.06	85.22±0.05	87.48±0.02	87.54±1.2
PRECISION	Agriculture	80.81±0.03	80.79±0.05	84.54±0.16	84.57±0.14
	Forest	58.14±0.17	57.93±0.14	87.37±0.22	87.45±0.24
	Building	94.48±0.4	94.85±0.26	48.66±0.09	52.93±5.26
	Water	94.55±0.03	94.55±0.03	96.19±0.31	72.86±28.81
	AVG	81.99±0.15	82.03±0.09	79.19±0.08	74.45±11.84
RECALL	Agriculture	75.02±0.07	75.08±0.05	60.45±0.17	61.63±1.41
	Forest	87.43±0.07	87.46±0.05	90.15±0.15	90.31±0.24
	Building	27.83±0.9	26.77±0.76	77.39±0.25	75.01±2.72
	Water	87.78±0.11	87.78±0.11	93.07±0.49	93.13±0.66
	AVG	69.52±0.36	69.27±0.3	80.26±0.13	80.02±0.94
F-SCORE	Agriculture	77.81±0.04	77.83±0.04	70.49±0.07	71.29±0.98
	Forest	69.83±0.1	69.7±0.1	88.74±0.08	88.86±0.22
	Building	42.99±1.03	41.75±0.9	59.75±0.13	61.79±2.58
	Water	91.04±0.06	91.04±0.06	94.6±0.11	78.18±20.05
	AVG	70.42±0.42	70.08±0.36	78.4±0.02	75.03±8.18

Table 1. Quantitative assessment of the unsupervised land cover categorization results.

4.3. Performance Evaluation

In order to evaluate the performance, it is important to emphasize that our parallel and baseline versions provide very similar results in terms of accuracy, using the Intel C++ Compiler

18.0.1 and OpenMP 4.5 with $-O3$, $-restrict$, $-xAVX$ and $-qopenmp$ flags. The considered versions have been tested on a multi-core system equipped with an Intel Xeon E5-1620 v3 (4 physical cores) at 3.50 GHz, with 32 GBytes of RAM memory and Debian GNU/Linux 9 as the operating system installed.

For illustrative purposes, Table 2 shows the timing results and speedups for the Munich dataset, considering both Sentinel-1 and Sentinel-2 sensors. The vectorization version is performed through the -xAVX (Intel Advanced Vector Extensions) and $-O3$ flags, where specific SIMD instructions are generated, data locality is exploited, and redundant computations are avoided. Accordingly, a slight improvement is achieved regarding to the baseline version but it is not enough to significantly accelerate the process. For this purpose, OpenMP directives are added to the previous version in order to obtain a better parallelization, achieving a relevant speedup of around 4x using 4 threads.

Table 2. Processing times (in seconds) and speedups achieved for the proposed multi-core implementation considering the best thread configuration (in the parentheses).

Approaches	**Munich**	
	Sentinel-1 (SAR)	**Sentinel-2 (MSI)**
Baseline	113.178	135.419
SIMD	84.606	87.331
SIMD + OpenMP	31.969 (4)	32.726 (4)
Speedup	3.54x	4.14x

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have discussed the possibility of exploiting parallel architectures for unsupervised land cover categorization of remotely sensed data. As a case study, we have presented an OpenMP+SIMD implementation of the pLSA algorithm. Probabilistic semantic analysis has the advantage that it can be performed in unsupervised fashion. Our experimental results show the effectiveness of the proposed parallel implementation, not only in terms of clustering accuracy but also in terms of computational performance. As future work, we will use this implementation for spectral unmixing of hyperspectral images.

6. REFERENCES

[1] Cristina Gomez, Joanne C. White, and Michael A. Wulder, "Optical remotely sensed time series data for land cover classification: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 116, pp. 55–72, 2016.

[2] Thomas Lillesand, Ralph W. Kiefer, and Jonathan Chipman, *Remote sensing and image interpretation*, John Wiley & Sons, 2014.

[3] Giles M. Foody, "Status of land cover classification accuracy assessment," *Remote sensing of environment*, vol. 80, no. 1, pp. 185–201, 2002.

[4] Ruben Fernandez-Beltran and Filiberto Pla, "Prior-based probabilistic latent semantic analysis for multimedia retrieval," *Multimedia Tools and Applications*, vol. 77, no. 13, pp. 16771–16793, 2017.

[5] Reza Bahmanyar, Daniela Espinoza-Molina, and Mihai Datcu, "Multisensor earth observation image classification based on a multimodal latent dirichlet allocation model," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 459–463, 2018.

[6] Ruben Fernandez-Beltran, Juan M. Haut, Mercedes E. Paoletti, Javier Plaza, Antonio Plaza, and Filiberto Pla, "Multimodal probabilistic latent semantic analysis for sentinel-1 and sentinel-2 image fusion," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 9, pp. 1347–1351, 2018.

[7] Ruben Fernandez-Beltran, Juan M. Haut, Mercedes E. Paoletti, Javier Plaza, Antonio Plaza, and Filiberto Pla, "Remote sensing image fusion using hierarchical multimodal probabilistic latent semantic analysis," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2018.

[8] Ruben Fernandez-Beltran and Filiberto Pla, "Incremental probabilistic latent semantic analysis for video retrieval," *Image and Vision Computing*, vol. 38, pp. 1–12, 2015.

[9] Mingmin Chi, Antonio Plaza, Jón Atli Benediktsson, Zhongyi Sun, Jinsheng Shen, and Yangyong Zhu, "Big data for remote sensing: Challenges and opportunities," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207–2219, 2016.

[10] Raymond Wan, Vo Ngoc Anh, and Hiroshi Mamitsuka, "Efficient probabilistic latent semantic analysis through parallelization," in *AIRS '09 Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology*, 2009, pp. 432–443.

[11] Zhao Liang, Wenye Li, and Yuxi Li, "A parallel probabilistic latent semantic analysis method on mapreduce platform," in *IEEE International Conference on Information and Automation (ICIA)*, 2013, pp. 1–10.

[12] Muhammad Mazhar Ullah Rathore, Anand Paul, Awais Ahmad, Bo-Wei Chen, Bormin Huang, and Wen Ji, "Real-time big data analytical architecture for remote sensing application," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 8, no. 10, pp. 4610–4621, 2015.

[13] Thomas Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[14] Todd K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[15] Yin Zhang, Rong Jin, and Zhi-Hua Zhou, "Understanding bag-of-words model: a statistical framework," *Int. Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[16] J. M. Haut, M. E. Paoletti, J. Plaza, and A. Plaza, "Cloud implementation of the K-means algorithm for hyperspectral image analysis," *Journal of Supercomputing*, vol. 73, no. 1, 2017.