Journal of Machine Learning Research 14 (2013) 2899-2903 Submitted 11/12; Revised 4/13; Published 9/13

The `CAM` Software for Nonnegative Blind Source Separation in `R-Java`

Niya Wang WANGNY@VT.EDU
Fan Meng MENGFAN@VT.EDU
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Arlington, VA 22203, USA

Li Chen CHENL14@MAIL.NIH.GOV
Pediatric Oncology Branch, National Cancer Institute
National Institutes of Health
Gaithersburg, MD 20850, USA

Subha Madhavan SM696@GEORGETOWN.EDU
Innovation Center for Biomedical Informatics
Georgetown University
Washington DC 20007, USA

Robert Clarke CLARKER@GEORGETOWN.EDU
Lombardi Comprehensive Cancer Center
Georgetown University
Washington, DC 20057, USA

Eric P. Hoffman EHOFFMAN@CNMCRESEARCH.ORG
Research Center for Genetic Medicine
Children's National Medical Center
Washington, DC 20010, USA

Jianhua Xuan XUAN@VT.EDU
Yue Wang YUEWANG@VT.EDU
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Arlington, VA 22203, USA

Editor: Antti Honkela

Abstract

We describe a `R-Java CAM` (convex analysis of mixtures) package that provides comprehensive analytic functions and a graphic user interface (`GUI`) for blindly separating mixed nonnegative sources. This open-source multiplatform software implements recent and classic algorithms in the literature including Chan et al. (2008), Wang et al. (2010), Chen et al. (2011a) and Chen et al. (2011b). The `CAM` package offers several attractive features: (1) instead of using proprietary `MATLAB`, its analytic functions are written in `R`, which makes the codes more portable and easier to modify; (2) besides producing and plotting results in `R`, it also provides a `Java GUI` for automatic progress update and convenient visual monitoring; (3) multi-thread interactions between the `R` and `Java` modules are driven and integrated by a `Java GUI`, assuring that the whole `CAM` software runs responsively; (4) the package offers a simple mechanism to allow others to plug-in additional `R`-functions.

Keywords: convex analysis of mixtures, blind source separation, affinity propagation clustering, compartment modeling, information-based model selection

1. Overview

Blind source separation (BSS) has proven to be a powerful and widely-applicable tool for the analysis and interpretation of composite patterns in engineering and science (Hillman and Moore, 2007; Lee and Seung, 1999). BSS is often described by a linear latent variable model **X = AS**, where **X** is the observation data matrix, **A** is the unknown mixing matrix, and **S** is the unknown source data matrix. The fundamental objective of BSS is to estimate both the unknown but informative mixing proportions and the source signals based only on the observed mixtures (Child, 2006; Cruces-Alvarez et al., 2004; Hyvarinen et al., 2001; Keshava and Mustard, 2002).

While many existing BSS algorithms can usefully extract interesting patterns from mixture observations, they often prove inaccurate or even incorrect in the face of real-world BSS problems in which the pre-imposed assumptions may be invalid. There is a family of approaches exploiting the source non-negativity, including the non-negative matrix factorization (NMF) (Gillis, 2012; Lee and Seung, 1999). This motivates the development of alternative BSS techniques involving exploitation of source nonnegative nature (Chan et al., 2008; Chen et al., 2011a,b; Wang et al., 2010). The method works by performing convex analysis of mixtures (CAM) that automatically identifies pure-source signals that reside at the vertices of the multifaceted simplex most tightly enclosing the data scatter, enabling geometrically-principled delineation of distinct source patterns from mixtures, with the number of underlying sources being suggested by the minimum description length criterion.

Consider a latent variable model $\mathbf{x}(i) = \mathbf{A}\mathbf{s}(i)$, where the observation vector $\mathbf{x}(i) = [x_1(i),...,x_M(i)]^T$ can be expressed as a non-negative linear combination of the source vectors $\mathbf{s}(i) = [s_1(i),...,s_J(i)]^T$, and $\mathbf{A} = [\mathbf{a}_1,...,\mathbf{a}_J]$ is the mixing matrix with $\mathbf{a}_j$ being the jth column vector. This falls neatly within the definition of a convex set (Fig. 1) (Chen et al., 2011a):

$$\mathbf{X} = \left\{ \sum_{j=1}^{J} s_j(i)\mathbf{a}_j | \mathbf{a}_j \in \mathbf{A}, s_j(i) \geq 0, \sum_{j=1}^{J} s_j(i) = 1, i = 1,...,N \right\}.$$

Assume that the sources have at least one sample point whose signal is exclusively enriched in a particular source (Wang et al., 2010), we have shown that the vertex points of the observation simplex (Fig. 1) correspond to the column vectors of the mixing matrix (Chen et al., 2011b). Via a minimum-error-margin volume maximization, `CAM` identifies the optimum set of the vertices (Chen et al., 2011b; Wang et al., 2010). Using the samples attached to the vertices, compartment modeling (CM) (Chen et al., 2011a) obtains a parametric solution of **A**, nonnegative independent component analysis (nICA) (Oja and Plumbley, 2004) estimates **A** (and **s**) that maximizes the independency in **s**, and nonnegative well-grounded component analysis (nWCA) (Wang et al., 2010) finds the column vectors of **A** directly from the vertex cluster centers.

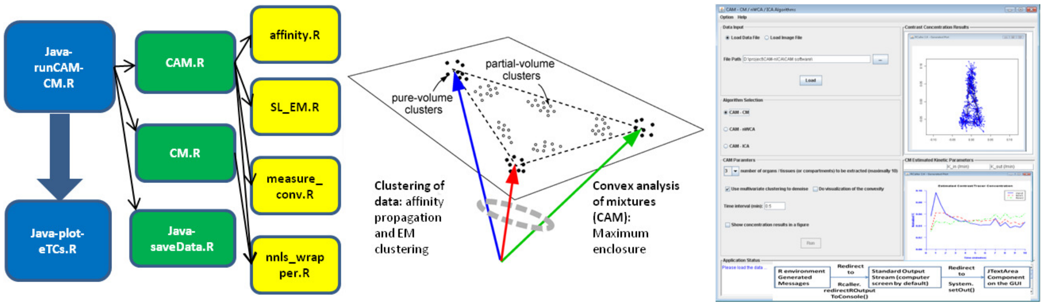

Figure 1: Schematic and illustrative flowchart of `R-Java CAM` package.

In this paper we describe a newly developed `R-Java CAM` package whose analytic functions are written in `R`, while a graphic user interface (`GUI`) is implemented in `Java`, taking full advantages of both programming languages. The core software suite implements `CAM` functions and includes normalization, clustering, and data visualization. Multi-thread interactions between the `R` and `Java` modules are driven and integrated by a `Java GUI`, which not only provides convenient data or parameter passing and visual progress monitoring but also assures the responsive execution of the entire `CAM` software.

2. Software Design and Implementation

The `CAM` package mainly consists of `R` and `Java` modules. The `R` module is a collection of *main* and *helper* functions, each represented by an `R` function object and achieving an independent and specific task (Fig. 1). The `R` module mainly performs various analytic tasks required by `CAM`: figure plotting, update, or error message generation. The `Java` module is developed to provide a `GUI` (Fig. 2). We adopt the model-view-controller (MVC) design strategy, and use different `Java` classes to separately perform information visualization and human-computer interaction. The `Java` module also serves as the *software driver and integrator* that use a multi-thread strategy to facilitate the interactions between the `R` and `Java` modules, such as importing raw data, passing algorithmic parameters, calling `R` scripts, and transporting results and messages.

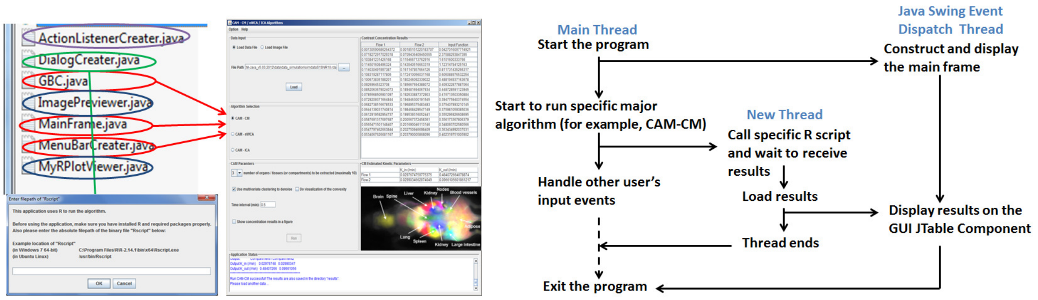

Figure 2: Interactive `Java GUI` supported by a multi-thread design strategy.

2.1 Analytic and Presentation Tasks Implemented in `R`

The `R` module performs the `CAM` algorithm and facilitates a suite of subsequent analyses including `CM, nICA, and nWCA`. These tasks are performed by the three *main* functions: `CAM-CM.R`, `CAM-nICA.R`, and `CAM-nWCA.R`, which can be activated by the three `R` scripts: `Java-runCAM-CM.R`, `Java-runCAM-ICA.R`, and `Java-runCAM-nWCA.R`. The `R` module also performs auxiliary tasks including automatic `R` library installation, figure drawing, and result recording; and offers other standard methods such as nonnegative matrix factorization (Lee and Seung, 1999), Fast ICA (Hyvarinen et al., 2001), factor analysis (Child, 2006), principal component analysis, affinity propagation, k-means clustering, and expectation-maximization algorithm for learning standard finite normal mixture model.

2.2 Graphic User Interface Written in `Java Swing`

The `Java GUI` module allows users to import data, select algorithms and parameters, and display results. The module encloses two packages: `guiView` contains classes for handling frames and

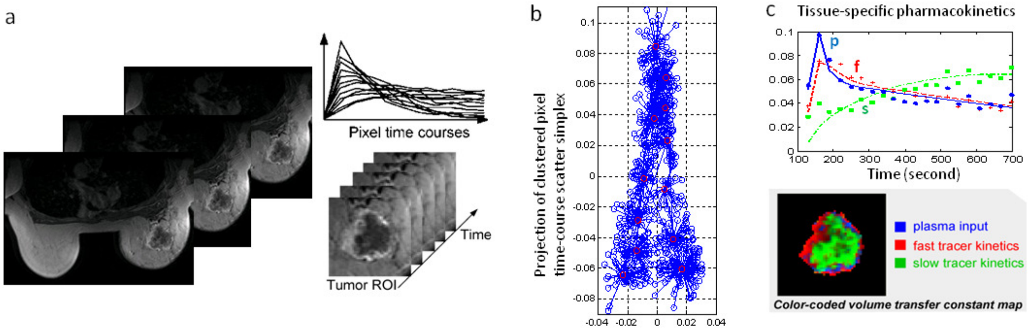

Figure 3: Application of `R-Java CAM` to deconvolving dynamic medical image sequence.

dialogs for managing user inputs; `guiModel` contains classes for representing result data sets and for interacting with the `R` script caller. Packaged as one jar file, the `GUI` module runs automatically.

2.3 Functional Interaction Between `R` and `Java`

We adopt the open-source program `RCaller (http://code.google.com/p/rcaller)` to implement the interaction between `R` and `Java` modules (Fig. 2), supported by explicitly designed `R` scripts such as `Java-runCAM-CM.R`. Specifically, five featured `Java` classes are introduced to interact with `R` for importing data or parameters, running algorithms, passing on or recording results, displaying figures, and handing over error messages. The examples of these classes include guiModel.MyRCaller.java, guiModel.MyRCaller.readResults(), and guiView.MyRPlotViewer.

3. Case Studies and Experimental Results

The `CAM` package has been successfully applied to various data types. Using dynamic contrast-enhanced magnetic resonance imaging data set of an advanced breast cancer case (Chen, et al., 2011b),“double click” (or command lines under Ubuntu) activated execution of `CAM-Java.jar` reveals two biologically interpretable vascular compartments with distinct kinetic patterns: fast clearance in the peripheral “rim” and slow clearance in the inner “core”. These outcomes are consistent with previously reported intratumor heterogeneity (Fig. 3). Angiogenesis is essential to tumor development beyond 1-2mm^3. It has been widely observed that active angiogenesis is often observed in advanced breast tumors occurring in the peripheral “rim” with co-occurrence of inner-core hypoxia. This pattern is largely due to the defective endothelial barrier function and outgrowth blood supply. In another application to natural image mixtures, `CAM` algorithm successfully recovered the source images in a large number of trials (see Users Manual).

4. Summary and Acknowledgements

We have developed a `R-Java CAM` package for blindly separating mixed nonnegative sources. The open-source cross-platform software is easy-to-use and effective, validated in several real-world applications leading to plausible scientific discoveries. The software is freely downloadable from `http://mloss.org/software/view/437/`. We intend to maintain and support this package in the future. This work was supported in part by the US National Institutes of Health under Grants CA109872, CA 100970, and NS29525. We thank T.H. Chan, F.Y. Wang, Y. Zhu, and D.J. Miller for technical discussions.

References

T.H. Chan, W.K. Ma, C.Y. Chi, and Y. Wang. A convex analysis framework for blind separation of non-negative sources. *IEEE Transactions on Signal Processing*, 56:5120–5143, 2008.

L. Chen, T.H. Chan, P.L. Choyke, and E.M. Hillman et al. Cam-cm: a signal deconvolution tool for in vivo dynamic contrast-enhanced imaging of complex tissues. *Bioinformatics*, 27:2607–2609, 2011a.

L. Chen, P.L. Choyke, T.H. Chan, and C.Y. Chi et al. Tissue-specific compartmental analysis for dynamic contrast-enhanced mr imaging of complex tumors. *IEEE Transactions on Medical Imaging*, 30:2044–2058, 2011b.

D. Child. The essentials of factor analysis. *Continuum International*, 2006.

S.A. Cruces-Alvarez, Andrzej Cichocki, and Shun ichi Amari. From blind signal extraction to blind instantaneous signal separation: criteria, algorithms, and stability. *IEEE Transactions on Neural Networks*, 15:859–873, 2004.

N. Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13:3349–3386, 2012.

E.M.C. Hillman and A. Moore. All-optical anatomical co-registration for molecular imaging of small animals using dynamic contrast. *Nature Photonics*, 1:526–530, 2007.

A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley, New York, 2001.

N. Keshava and J.F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19:44–57, 2002.

D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

E. Oja and M. Plumbley. Blind separation of positive sources by globally convergent gradient search. *Neural Computation*, 16:1811–1825, 2004.

F.Y. Wang, C.Y. Chi, T.H. Chan, and Y. Wang. Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:857–888, 2010.