FastSRM: A fast, memory efficient and identifiable implementation of the shared response model

September 16, 2022

Abstract

The shared response model (SRM) provides a simple but effective framework to analyze fMRI data of subjects exposed to naturalistic stimuli. However when the number of subjects or runs is large, fitting the model requires a large amount of memory and computational power, which limits its use in practice. Furthermore, SRM is not identifiable, which makes the shared response difficult to interpret.

In this work, we implement an identifiable version of SRM and show on real data that it improves the stability of the recovered shared response. We then introduce FastSRM, that relies on a dimension reduction step and yields the same solution as the original algorithm. We show experimentally using synthetic and real fMRI data that FastSRM is considerably faster and more memory efficient than current implementations.

The experiments performed in this article are *fully* reproducible: our code available at `https://github.com/hugorichard/FastSRM` allows you to download the data, run the experiments and plot the figures.

1 Introduction

Complex stimulations such as movie-watching, story or music listening are more and more popular in the neuro-scientific community. Indeed such naturalistic paradigms are unconstrained from behavioral manipulations and thus, more ecological with respect to every-day cognitive conditions. If one wants to use data acquired under naturalistic conditions to map functional responses to the brain, i.e. in an encoding setting [1], one has to deal with the fact that no design matrix is specified for naturalistic stimuli. Such a design matrix encodes the temporal events that affect brain signal during an experiment; these events typically reflect the occurrence of some features of interest in the stimuli. Although some works have used manual annotations [2] or deep neural networks [3] [4], see also `https://neuroscout.org`, to create an estimate of the design matrix of naturalistic stimuli, it is a hard task. Without a design matrix, models such as the general linear model [5] cannot be used.

Another approach is to learn jointly a set of reference time courses and the spatial maps in an unsupervised way. These reference time courses formally replace the design matrix in the spatio-temporal decomposition of the data, but cannot be considered as a design matrix, as it is data-driven. In the case where the dataset only contains one subject, independent component analysis [6] (ICA) is the method of choice. ICA assumes that components are independent, a defendible assumption which ensures identifiability up to permutation and scaling of the model and can be fitted efficiently. When the dataset contains multiple subjects undergoing the same protocol, a natural assumption is to assume that the temporal reference is shared across subjects. Many different methods can produce shared components from different subjects. Some assume independent components [7] [8] [9] [10], while others are only based on second order information such as Multiset Canonical Correlation Analysis [11] or the Shared Response Model (SRM) [12].

This work focuses on the SRM. Because of its simplicity and its built-in dimension reduction, this model is widely used [13] [14] [15] [16] [17], in particular, as a preprocessing step for source identification [7] [8], classification [18] [19][20] or as a basis for transfer learning [21].

What makes the analysis of fMRI data particularly difficult is that the number of features (voxels) is usually much larger than the number of available samples (time frames). Yet, SRM has initially been designed to work within regions of interest using only few subjects. When using full brain data, computational costs become high. In addition, memory requirements are difficult to meet since the full dataset needs to hold in memory. Lastly, another issue of SRM is its non-identifiability (see the proof in [8] Appendix D also reported in Appendix B for completeness) which makes it challenging to interpret the extracted shared components as they are defined up to a rotation.

In this work, we introduce FastSRM, an identifiable model that uses an optimal spatial decomposition to speed up the computations with provably no loss of performance. FastSRM gains identifiability by imposing that the covariance of the shared components is diagonal (see [8] Appendix D or Appendix B of this paper for a proof). The gains in speed and memory usage are based on the observation that a low-dimensional, yet distance-preserving representation of the images yields the same result as the full data. Such a representation can be interpreted as an initial spatial decomposition of the data. We show that FastSRM brings several practical benefits: on real data, it produces more stable estimates of the shared components than its non-identifiable counterpart, and it is much faster and more memory efficient than other implementations that do not make use of an optimal spatial decomposition.

A Python implementation is available at `https://github.com/hugorichard/FastSRM`. All our experiments are fully reproducible. Scripts to download the data, run the experiments and plot the figures are included. A continuous integration pipeline runs the tests automatically when any change to the core algorithm is made.

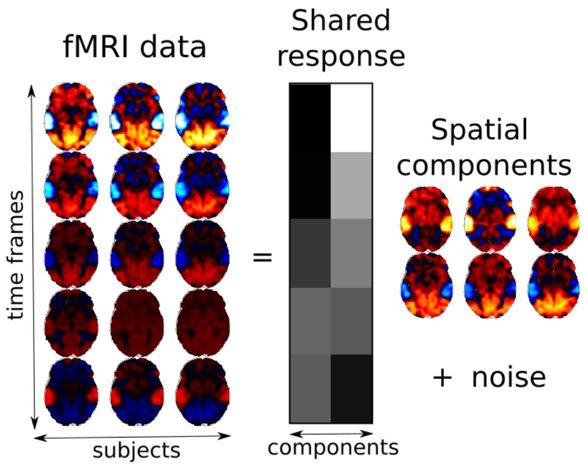

Figure 1: **Shared response model**: The raw fMRI data are modeled as a weighted combination of subject-specific spatial components with additive noise. The weights are shared between subjects and constitute the shared response to the stimuli.

2 Background: the shared response model (SRM)

The shared response model [12] is a multi-view latent factor model. The data $\mathbf{x}_1 \dots \mathbf{x}_m$ are modeled as random vectors following the model:

$$\mathbf{x}_i = A_i \mathbf{s} + \mathbf{n}_i \tag{1}$$

$$A_i^\top A_i = I_p \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^v$ is the data of subject i, called view i, $A_i \in \mathbb{R}^{p \times v}$ is the mixing matrix of view i, $\mathbf{n}_i$ is the noise of view i and $\mathbf{s} \in \mathbb{R}^p$ are the shared components referred to as the *shared response* in fMRI applications. p is the number of time frames, v is the number of voxels and m is the number of subjects.

The mixing matrices A_i are assumed to be orthogonal so that $A_i^\top A_i = I_p$. However, in general the matrix $A_i A_i^\top$ is different from identity. The noise $\mathbf{n}_i$ is assumed to be Gaussian with covariance Σ_i and independent across views. We assume the number of features v to be much larger than the number of components p: $v \gg p$.

The conceptual figure 1 illustrates an application of the shared response model to fMRI data. The mixing matrices are spatial topographies specific to each subject while the shared components represent the common timecourses. In [12, 22], two versions of the shared response model are introduced, namely the deterministic and probabilistic models.

2.1 Deterministic shared response model

Let us consider n observations (scans) of $\mathbf{x}_i$ and $\mathbf{s}$ that we stack into matrices $X_i \in \mathbb{R}^{v,n}$ and $S \in \mathbb{R}^{p,n}$. The deterministic shared response model considers both the mixing matrices A_i and the n observations of the shared response S as parameters to be estimated. The noise variance is fixed to a multiple of identity: $\forall i, \Sigma_i = \sigma^2 I_v$ where σ is an hyper-parameter to choose.

The model is optimized by maximizing the log-likelihood, assuming that the noise is Gaussian distributed. The likelihood is then given by: $p(\mathbf{x}) = \prod_i \mathcal{N}(\mathbf{x}_i; A_i \mathbf{s}, \sigma^2 I)$ and therefore the empirical negative log-likelihood is given up to a constant independent of A_i and S by:

$$\mathcal{L} = \frac{1}{n} \sum_i \|A_i S - X_i\|^2 = \frac{1}{n} \left(\|S\|^2 - 2\langle A_i S, X_i \rangle + \|X_i\|^2 \right) \tag{3}$$

$\mathcal{L}$ is optimized by performing alternate minimization on $(A_1 \dots A_m)$ and S. Note that the hyper-parameter σ does not have an influence on the loss and can therefore be ignored.

The gradient with respect to S is given by $\sum_i A_i^\top (A_i S - X_i) = \sum_i (S - A_i^\top X_i)$ yielding the closed form updates:

$$S \leftarrow \frac{1}{m} \sum_i (A_i^\top X_i) \tag{4}$$

From (3), minimizing $\mathcal{L}$ with respect to A_i is equivalent to maximizing $\langle A_i, X_i S^\top \rangle$ and therefore we have:

$$A_i \leftarrow \mathcal{P}\left(\frac{1}{n} X_i S^\top \right) \tag{5}$$

where $\mathcal{P}$ is the projection on the Stiefel manifold: $\mathcal{P}(M) = M(M^\top M)^{-\frac{1}{2}}$.

The complexity of Deterministic SRM is in $\tilde{O}(\mathrm{n}_{\mathrm{iter}} mpvn)$, where $\mathrm{n}_{\mathrm{iter}}$ is the number of iterations. We monitor the convergence by computing the ℓ_∞ norm of the gradient. Note that we can monitor the gradient without any increase in complexity. Indeed, after the updates with respect to each mixing matrix have been carried out, the gradient with respect to $(A_i)_{i=1}^m$ is zero and therefore to compute the ℓ_∞ norm of the gradient we only need the gradient with respect to S: $mS - \sum_i A_i^\top \mathbf{x}_i$ where the right hand side is used in the updates of S. The algorithm is stopped when the gradient falls below a chosen tolerance.

2.2 Probabilistic SRM

In Probabilistic SRM , $\Sigma_i = \sigma_i^2 I_v$ and the shared components are assumed to be Gaussian $\mathbf{s} \sim \mathcal{N}(0, \Sigma_s)$.

In [12] and [22], Σ_s is only assumed to be definite positive. As already highlighted in introduction, this causes the model to be unidentifiable (see [8] Appendix D or Appendix B of this paper for a proof). Enforcing a diagonal

Σ_s ensures identifiability, provided that the diagonal values are different. So we assume here that Σ_s is diagonal (and refer the interested reader to [12] and [22] for the original formulation of Probabilistic SRM without the diagonal constraint). The following paragraph gives update rules for λ_i and A_i and Σ_s. We highlight that the derivations used to obtain these updates are not new. They are obtained following the strategy introduced in [22] and [12].

The model is optimized via the expectation maximization algorithm. We give all details on the formulation and derivation of update rules in section A. Denoting $\mathbb{V}[\mathbf{s}|\mathbf{x}] = (\sum_i \frac{1}{\sigma_i^2} I + \Sigma_s^{-1})^{-1}$ and $\mathbb{E}[\mathbf{s}|\mathbf{x}] = \mathbb{V}[\mathbf{s}|\mathbf{x}] \sum_i \frac{1}{\sigma_i^2} A_i^\top \mathbf{x}_i$, the updates are given by:

$$\sigma_i^2 \leftarrow \frac{1}{v}(\mathbb{E}[\|\mathbf{x}_i - A_i \mathbb{E}[\mathbf{s}|\mathbf{x}]\|^2] + \|\mathrm{diag}(\mathbb{V}[\mathbf{s}|\mathbf{x}])\|) \tag{6}$$

$$A_i \leftarrow \mathcal{P}(\mathbb{E}[\mathbf{x}_i \mathbb{E}[\mathbf{s}|\mathbf{x}]^\top]) \tag{7}$$

$$\Sigma_s \leftarrow \mathbb{V}[\mathbf{s}|\mathbf{x}] + \mathbb{E}[\mathbb{E}[\mathbf{s}|\mathbf{x}]\mathbb{E}[\mathbf{s}|\mathbf{x}]^\top] \tag{8}$$

The complexity of Probabilistic SRM is $\tilde{O}(\mathrm{n_{iter}} mpvn)$, the same as in Deterministic SRM. We can monitor the convergence by computing the log-likelihood decrease at each iteration and stop the algorithm when the magnitude of the decrease is below some tolerance. The storage requirements of Deterministic or Probabilistic SRM are in $\tilde{O}(mvn)$ which simply means that the dataset needs to hold in memory.

3 The FastSRM algorithm

3.1 Reducing the computational burden by the use of spatial decompositions

SRM algorithms use different sets of parameters θ to represent the data. In deterministic SRM $\theta = ((A_i)_{i=1}^m, S)$ where $(A_i)_{i=1}^m$ are the mixing matrices and S is the shared response, while in probabilistic SRM $\theta = ((A_i)_{i=1}^m, \Sigma_s, (\sigma_i)_{i=1}^m)$ where $(A_i)_{i=1}^m$ are the mixing matrices, $(\sigma_i)_{i=1}^m$ the noise levels and Σ_s the components variance.

In fMRI, spatial decompositions are often used to reduce the dimensionality of the data. A classical approach is to apply a deterministic atlas such as [23] which is a parcellation of the brain into r regions. There also exist probabilistic atlases such as [24] that describes each region as a set of weights across the full brain.

Deterministic and probabilistic atlases are spatial decompositions that do not depend on the view at hand. In FastSRM, we consider a set of view-specific spatial decompositions $U_i \in \mathbb{R}^{v \times r}$ such that $U_i^\top U_i = I_r$ where r is the number of components in the spatial decompositions. Data are reduced using $\mathbf{z}_i = U_i^\top \mathbf{x}_i$ and an SRM algorithm is applied on data $\mathbf{z}_i$ yielding parameters θ'. Figure 2 illustrates this process.

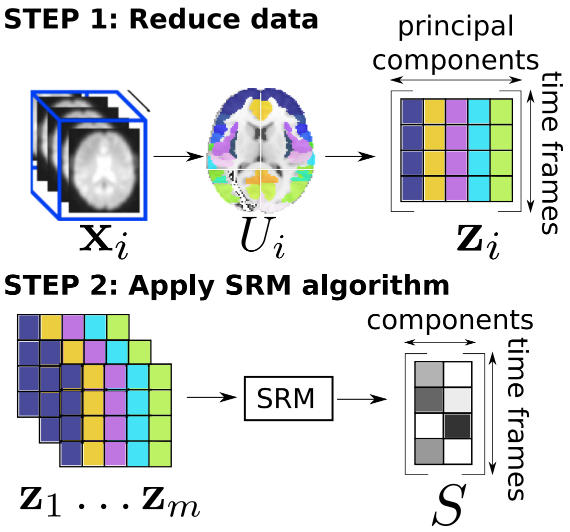

Figure 2: **FastSRM algorithm** In step 1, data $\mathbf{x}_i$ are projected onto a spatial decomposition U_i that may depend on the subject i (top). In step 2 a SRM algorithm is applied on reduced data to compute the shared response.

Note that the parameters obtained with FastSRM θ' are different from the parameters obtained with the corresponding SRM algorithm θ (the unmixing matrices in θ' do not even have the same shape as the unmixing matrices in θ). However, as we will see in the next section, there exist spatial decompositions such that the models parametrized by θ and θ' are equivalent.

From a computational stand point, dimension reduction provides a large reduction in memory usage. Indeed as the original data are seen only once, it is no longer necessary to keep the full dataset in memory (we can load data X_i one after the other and similarly for the spatial decomposition U_i). Therefore the memory consumption is only in $\tilde{O}(vn)$ (where v is the number of voxels and n is the number of samples) which is lower than that of SRM by a factor of m, the number of subjects. The number of subjects is typically between 10 and 1000. This yields a practical benefit: on fMRI datasets with many subjects, one no longer needs a large cluster to run the shared response model but only a modern laptop. Additionally, low memory consumption reduces the risk of thrashing [25], a phenomenon that causes large increase in computation time when the memory used is close to the total available memory in the hardware.

After preprocessing, the reduced representation $\mathbf{z}_i$ is used instead of the original data $\mathbf{x}_i$ yielding a time complexity of $\tilde{O}(\mathrm{T}_{\text{preprocessing}} + \mathrm{n}_{\text{iter}} mpnr)$. Let us highlight that an experiment is often run multiple times, typically when the analysis requires cross-validation procedures. In these cases, the pre-processing is performed only once and the apparent complexity becomes $\tilde{O}(\mathrm{n}_{\text{iter}} mpnr)$ which is faster than SRM by a factor of $\frac{v}{r}$. The number of components in large spatial decompositions is about $r = 1000$ and in full brain data, the number of voxels is typically in the 50 000 − 400 000 range, so that $\frac{v}{r}$ is typically about 50 to 400. It remains to be shown how to draw a correspondence between FastSRM and SRM, which we do in the following section.

3.2 A class of optimal spatial decompositions

In principle, FastSRM can be used with any spatial decomposition. However, in general, working with reduced data induces a loss of information that can be minimized if the spatial decomposition is carefully chosen. In any case, there is little hope to recover the parameters that would have been obtained from SRM from the parameters of FastSRM. Yet, we show that there exists an optimal spatial decomposition in the sense that SRM and FastSRM yield the same results.

Let us consider $\mathbf{x}_i = U_{\mathbf{x}_i}\mathbf{z}_i$ a PCA of $\mathbf{x}_i$ with the maximum number of components. As the number of samples n is lower than the number of features, $U_{\mathbf{x}_i} \in \mathbb{R}^{v \times n}$ and $\mathbf{z}_i \in \mathbb{R}^n$. We also have $U_{\mathbf{x}_i}^\top U_{\mathbf{x}_i} = I$. Therefore $U_{\mathbf{x}_i}$ constitutes a possible choice of subject specific spatial decomposition. As the next property shows, $U_{\mathbf{x}_i}$ is an optimal spatial decomposition for deterministic FastSRM. Note that this optimal spatial decomposition is not unique: any orthogonal matrix with the same span as the columns of X_i would also work.

Proposition 1 (Optimal spatial decomposition for deterministic FastSRM)**.** *Let $(A_i)_i, S$ be the solution obtained by deterministic SRM on data $(X_i)_i$ and $(A'_i)_i, S'$ the solution obtained by deterministic FastSRM on data $(X_i)_i$ using spatial decompositions $(U_{X_i})_i$ where $X_i = U_{\mathbf{x}_i}Z_i$ is a PCA of X_i. Then $A_i = U_{X_i}A'_i$ and $S = S'$.*

The proof is available in appendix C. In the case of probabilistic SRM we can obtain very similar results. However the algorithm applied on reduced data needs to be slightly modified. We have the following result:

Proposition 2 (Optimal spatial decomposition for probabilistic FastSRM)**.** *Let $(A_i)_i, \sigma_i, \Sigma_s$ be the solution obtained by probabilistic SRM on data $\mathbf{x}_i \in \mathbb{R}^v$ and $(A'_i)_i, \sigma'_i, \Sigma'_s$ the solution obtained by ProbSRM on data $\mathbf{z}_i = U_{\mathbf{x}_i}^\top \mathbf{x}_i$, $\mathbf{z}_i \in \mathbb{R}^n$ but where updates:*

$$\sigma_i^2 \leftarrow \frac{1}{n}(\mathbb{E}[\|\mathbf{z}_i - A_i\mathbb{E}[\mathbf{s}|\mathbf{z}]\|^2] + \|\mathrm{diag}(\mathbb{V}[\mathbf{s}|\mathbf{z}])\|^2) \tag{9}$$

are replaced by updates

$$\sigma_i^2 \leftarrow \frac{1}{v}(\mathbb{E}[\|\mathbf{z}_i - A_i\mathbb{E}[\mathbf{s}|\mathbf{z}]\|^2] + \|\mathrm{diag}(\mathbb{V}[\mathbf{s}|\mathbf{z}])\|^2) \tag{10}$$

where n is the number of samples and v is the number of voxels.

Then $A_i = U_{\mathbf{x}_i}A'_i$, $\sigma_i = \sigma'_i$ and $\Sigma_s = \Sigma'_s$.

The proof is available in appendix D. Proposition 1 and Proposition 2 show that no information is lost by replacing $\mathbf{x}_i \in \mathbb{R}^v$ by its reduced representation $\mathbf{z}_i \in \mathbb{R}^n$. A key observation is that Proposition 1 and Proposition 2 hold whether or not the model for deterministic (respectively probabilistic) SRM is indeed the generative model of the data.

A complexity analysis shows that finding an optimal spatial decomposition becomes the limiting step of the pipeline. Even with fast implementations, subject specific PCA is costly. However FastSRM only works on $\mathbf{z}_i$ so we do not need to know the value of $U_{\mathbf{x}_i}$. In practice, we observe data $X_i \in \mathbb{R}^{v \times n}$ and we want to get $Z_i \in \mathbb{R}^{n \times n}$ such that $X_i = U_{\mathbf{x}_i} Z_i$. This can be done by performing an SVD of $X_i^\top X_i$ yielding $X_i^\top X_i = V_i D_i V_i^\top$ and setting $Z_i = D_i^{\frac{1}{2}} V_i^\top$. Although computing the product $X_i^\top X_i$ has time complexity $\tilde{O}(vt^2)$ (the same as a PCA) the constant in the $\tilde{O}$ is exactly one so this operation costs a lot less than a PCA on full data. When estimates of the mixing matrices are needed, they can be obtained by applying equation (26) in the deterministic SRM case and equation (31) in the probabilistic SRM case which only costs $\tilde{O}(mvp^2)$. In practice the cost of the matrix products $X_i^\top X_i$ is often the limiting step of the pipeline (this depends on the number of iterations) but as we show next, it is much more efficient than performing SRM on the full data. Note that if large memory resources are available, these matrix products can be computed in parallel.

4 Related work

The implementation of SRM proposed in [12] is quadratic in the number of features, which prevents its application to full brain data. In [22], this issue is addressed by using the inversion lemma to remove the quadratic dependence. Their implementation is now the most widely used one. In our work, we propose to reduce further the computation time and memory usage compared to the implementation in [22] by the use of optimal spatial decompositions.

Other popular methods used to speed up SRM are searchlight [20] or piecewise [26] approaches. While these methods are efficient, they do not optimize the same objective as the original SRM algorithm and are arguably less principled since the searchlights or pieces are chosen a priori.

We consider SRM as a well principled formulation of the hyperalignment method [27]. Many methods exist to perform hyperalignment: deep hyperalignment [28] , robust SRM [18] , multi-view canonical correlation analysis [29], ShICA [7], MultiViewICA [8] optimal transport [30] and many more. SRM has been introduced after hyper-alignment in [12] and has been shown to outperform hyperalignment in many different settings. Specifically, SRM is natively a group model, which hyperalignment is not, it is more computationally efficient, with the avoidance of searchlight loops, and sometimes obtains higher accuracy than Procrustes hyperalignment [26]. In this article, we do not discuss further the merit of FastSRM with respect to other methods. We simply observe that SRM is widely used and aim to provide a faster, more memory efficient and identifiable implementation. In our view, FastSRM should be used as a plugin replacement for SRM.

5 Experiments

We make several experiments on both synthetic and real data. We used Nilearn [31] (version 0.8.1) for fMRI data preprocessing, Brainiak [32] (version 0.9) for the non-identifiable version of SRM that implements the work of [12] and [22], Numpy [33] (version 1.19.0) for array processing, Scipy [34] (version 1.5.3) its implementation of the Hungarian algorithm, Matplotlib [35] (version 3.1.2) for plotting and Sklearn [36] (version 0.23.2) for machine learning pipelines.

5.1 Comparing Fitting time and performance of FastSRM and SRM on synthetic data

We generate synthetic data $\mathbf{x}_i$ according to the model of probabilistic SRM. The parameters σ_i^2, A_i and Σ_s are generated randomly. We sample the value of the subject specific noise level from a normal distribution: $\sigma_i \sim \mathcal{N}(0, 0.1)$ The mixing matrices A_i are obtained by sampling their coefficient from a standardized normal distribution. Lastly, the covariance of the shared response Σ_s is diagonal and the diagonal values are obtained by sampling from a Dirichlet distribution with parameter $(1 \dots 1)$. We set the number of voxels to $v = 125\ 000$, the number of subjects to $m = 10$ and the number of components to $p = 50$. We generate $n = 1000$ samples.

We benchmark deterministic SRM, probabilistic SRM and their FastSRM counterparts in terms of fitting time and performance. Note that in this section, the identifiable implementation of deterministic SRM and probabilistic SRM described in section 2.2 and 2.1 is used. Algorithms are designated by the spatial decomposition they use and therefore SRM algorithms are referred to as *None* because no spatial decomposition is used and FastSRM algorithms will have the label *Optimal*. Note that it would be possible to use FastSRM with sub-optimal spatial decompositions (there exists a wide variety of brain spatial decompositions [37, 23, 38]) but using them does not bring any guarantee that the performance is the same as SRM.

We use a number of iterations between 1 and 100 and report the performance, fitting time and a measure of convergence. In FastSRM, we do not compute the unmixing matrices but only the shared response. We measure the performance of an algorithm by computing the error between the true component $S \in \mathbb{R}^{p \times n}$ and the predicted component $\hat{S} \in \mathbb{R}^{p \times n}$ using the quantity:

$$\mathrm{mse}(\hat{S}, S) = min_{B \in \mathbb{R}^{p \times p}} \|B\hat{S} - S\|_F^2 = \|S\hat{S}^\dagger \hat{S} - S\|_F^2 \tag{11}$$

This way of measuring errors is insensitive to the indeterminacies in DetSRM. We measure the fitting time in seconds. Lastly, we measure convergence by computing the gradient ℓ_∞ norm in case of DetSRM given by $\max(\mathrm{abs}(G))$ where G is the gradient and use the distance between consecutive values of the loss for ProbSRM. Results are plotted in Figure 3.

We empirically see that the optimal approach is equivalent to using no spatial decomposition in terms of performance. This is predicted by the theory in section 3.2, where we demonstrate that these two algorithms yield exactly the same output from a given input and initialization. We also see that ProbSRM gives much better results than DetSRM. In terms of fitting time, FastSRM is about a thousand time faster than SRM after 100 iterations. When no spatial decomposition is used, the number of iterations has a very strong impact on performance while it has a small impact when an optimal spatial decomposition is used. Lastly, looking at the convergence curves, we see that even after 100 iterations, algorithms did not fully converge. This means that in practice a larger number of iterations is needed, which would yield an even higher difference in fitting time between methods using no spatial decomposition and methods using an optimal spatial decomposition.

5.2 Experiment on fMRI data: identifiability increases stability

We evaluate the performance of the different SRM implementations on the *Sherlock* datasets where fMRI of 17 participants watching "Sherlock" BBC TV show (episode 1) is performed. These data are downloaded from `http://arks.princeton.edu/ark:/88435/dsp01nz8062179`. Data were acquired using a 3T scanner with an isotropic spatial resolution of 3 mm. More information including the preprocessing pipeline is available in [39]. Subject 5 is removed because of missing data, leaving us with 16 participants. Although SHERLOCK data originally contain 1 run only, we split them into 4 blocks of 395 time frames and one block of 396 time frames for the needs of our experiments.

We first show that identifiability is a desirable property as it increases stability. Then we show that our FastSRM algorithm (that works with optimal spatial decompositions) matches the performance of SRM (that works on full data) but use much less computational resources.

5.3 Identifiability increases stability

Assuming that the data follow the model, identifiability ensures that the problem is well-posed since the solutions are well-characterized. More precisely, if the data are generated according to an identifiable model, in the limit of infinite samples, a perfect optimization algorithm is guaranteed to recover the parameters used to generate the data. In contrast, in an unidentifiable model, there are more than one set of parameters that can generate the same data making the search for the good parameters ill-defined. In practice, unidentifiable models have a more complex optimization landscape with many more local minima than identifiable models (any local minima in the identifiable model produces as many local minimas as they are indeterminacies in the corresponding unidentifiable model). This is why the unidentifiable models are therefore expected to be more sensitive to initialization and less robust to small changes in the data than their identifiable counterparts. Note however that SRM remains non-convex so there is no guarantee that the optimal solution is found.

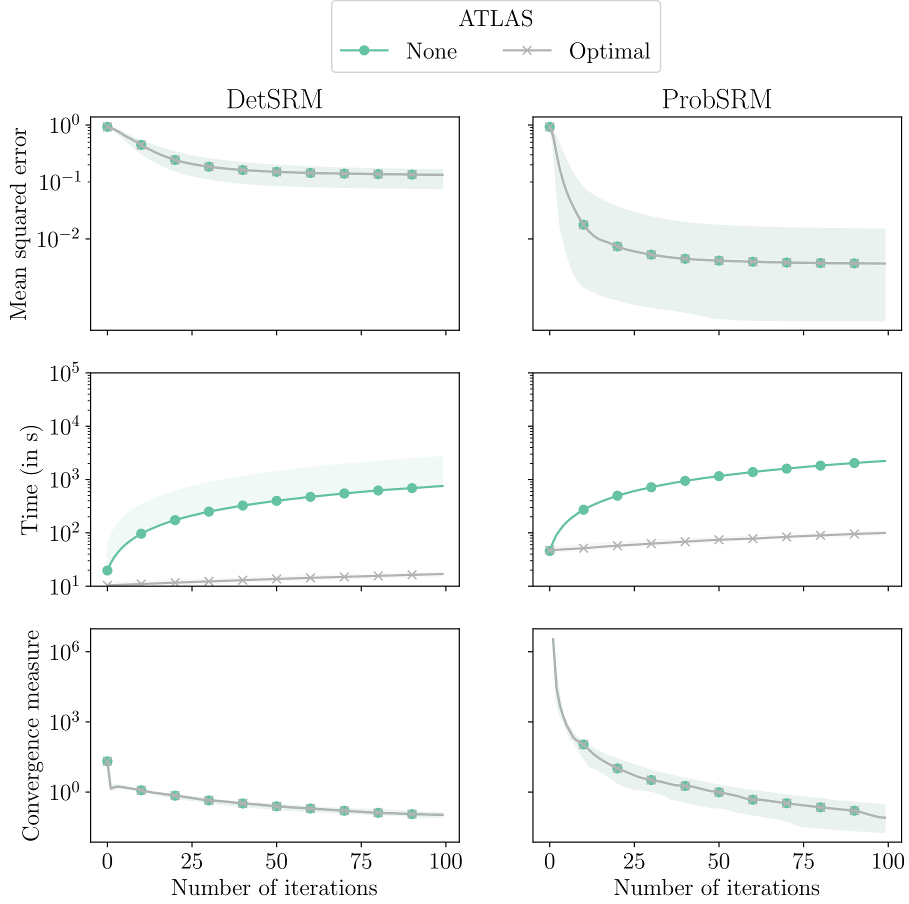

Figure 3: **Benchmark of SRM algorithms on synthetic data:** Performance, fitting time and convergence of SRM algorithms in the deterministic (left) or probabilistic (right) case. As expected, when optimal spatial decompositions are used, the performance is the same as if no spatial decomposition is used but the fitting time is much lower. This is even more pronounced when the number of iterations is high (and looking at convergence curves, we see that more iterations could be performed to be closer to a stationary point).

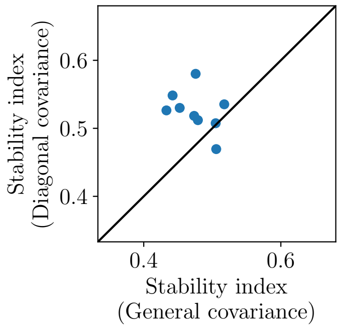

Figure 4: **Identifiability increases stability:** We first divide the subjects into two groups and extract the common components in each group. The components of the two groups are then matched using the Hungarian algorithm and the stability index is determined by the average correlation of the matched components. The procedure is repeated 9 times with the Brainiak implementation (not identifiable since the shared components covariance is unconstrained) and our FastSRM implementation (identifiable since the shared components covariance is diagonal).

In real fMRI data, the model cannot be expected to hold perfectly, but we can hope for greater stability in the parameters recovered than if an non-identifiable model is used.

To measure the stability of the shared components obtained from the Sherlock dataset, we divide the subjects into two roughly equal groups and extract the common components in each group. The components are then matched using the Hungarian algorithm [40] and the stability measure is obtained from the average correlation of the matched components. The procedure is repeated 9 times using the Brainiak implementation (that is not identifiable since the shared components covariance is unconstrained) and our FastSRM implementation. We plot the results as a scatter plot in Figure 4, where, for each repetition, the x-axis indicates the stability measure obtained with Brainiak's implementation and the y-axis the stability measure obtained with FastSRM (identifiable since the shared components covariance is diagonal). We see that introducing a diagonal source covariance improves stability.

5.4 Comparing fitting time, memory usage and performance on a timesegment matching experiment

Time-segment matching has first been introduced in [12]. In a nutshell, the time-segment matching accuracy measures the similarity between two multivariate time-series by trying to localize a time-segment in one time-series by correlation with the other. In the context of movie watching, this measure is quite meaningful: if we split the movies in scenes and compute a representation per scene and per subject, it can be assumed that different subjects watching the movie would still have closer representation of the same scenes than of different scenes. This explains why timesegment matching is a standard evaluation of SRM-like methods also used in [41], [42] or [20].

We now describe more precisely the experimental design. We split the runs into a train and test set. After fitting the model on the training set, we apply the unmixing matrices $W_i = A_i^{-1}$ of each subject on the test set yielding

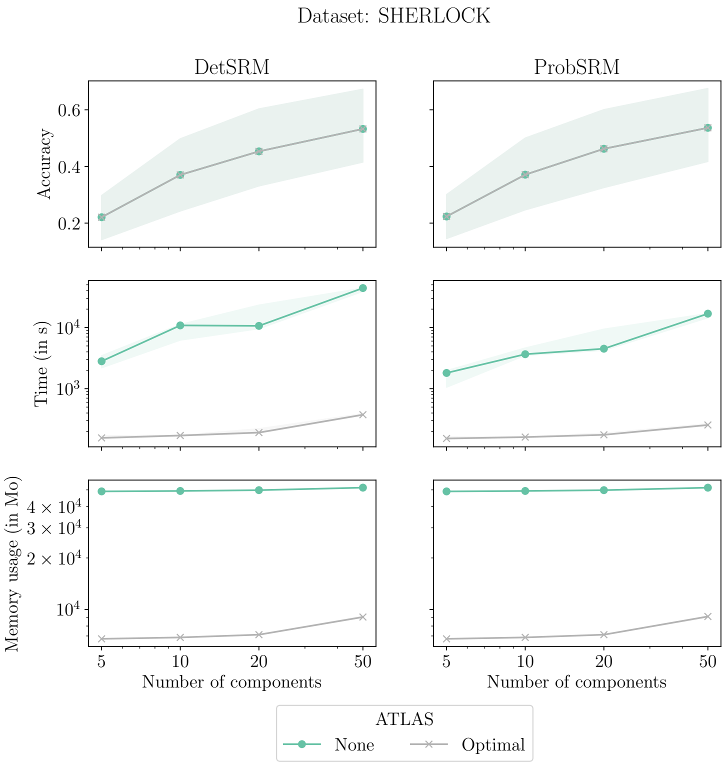

Figure 5: **Benchmark of SRM algorithms on fMRI data** (top) Timesegment matching accuracy (middle) Fitting time (bottom) Memory usage. When the optimal spatial decomposition is used, the accuracy is the same as when no spatial decomposition is used but the fitting time is reduced by a factor 10 to 100 and so is the memory usage.

individual components matrices. We estimate the shared responses by averaging the individual components of each subject but one. We select a target time-segment (9 consecutive time frames) in the shared responses and try to localize the corresponding time segment in the components of the left-out subject using a maximum-correlation classifier.

The time-segment is said to be correctly classified if the correlation between the sample and target time-segment is higher than with any other time-segment (partially overlapping time windows are excluded).

We use 5-Fold cross-validation across runs: the training set contains 80% of the runs and the test set 20%, and repeat the experiment using all possible choices for left-out subjects. The mean accuracy is reported in Figure 5 (top). When an optimal spatial decomposition is used, the accuracy is the same as when no spatial decomposition is used but the fitting time is reduced by a factor 10 to 100 and so is the memory usage (see Figure 5, bottom).

6 Implementation and reproducibility

Our work is fully reproducible and our code is tested and well documented. All material is available in the `https://github.com/hugorichard/FastSRM` Github repository.

The API is reminiscent of the scikit-learn one where the model object is instantiated with its parameters and then fitted on data via a method *fit()*. After the model is fit, the learned parameters can be accessed. Then, a *transform()* method computes the shared response from the learned parameters and the data passed as the argument of the *transform()* method.

While Brainiak and Nilearn are useful to reproduce some of our experiments, they are not strong dependencies and FastSRM only needs Numpy ($\geq$ 1.12), Scipy ($\geq$ 0.18.0), Matplotlib ($\geq$ 2.0.0), Scikit-learn ($\geq$ 0.23), Pytest ($\geq$ 6.2.5) , and Joblib ($\geq$ 1.1.0).

The package comes with a documentation and a tutorial that reproduces, at a smaller scale, the experiment available in Figure 3. These are found at `https://hugorichard.github.io/FastSRM/index.html`. The instructions to fully reproduce the experiments presented in the papers are available in the README: `https://github.com/hugorichard/FastSRM/blob/master/README.md`.

A set of tests reduces the risk of introducing a bug. These tests are run any time a pull request or a merged is performed on the main codebase.

7 Conclusion

As studies using naturalistic stimuli will tend to become more common within and across subjects, we need scalable models in terms of computation time and memory usage. This is particularly related to the development of deep phenotyping models such as the Courtois project on Neural Modeling (see `https://docs.cneuromod.ca`). This is precisely what FastSRM provides.

FastSRM is an efficient implementation of SRM that uses optimal spatial decompositions to speed up computations and reduce memory requirements with provably no loss of performance.

We show on synthetic and real data that FastSRM is much faster and more memory efficient that implementations not using an optimal spatial decomposition. Furthermore, FastSRM is identifiable. On real data, we show that compared to Brainiak's implementation, that is not identifiable, FastSRM provides more stable estimates of the shared components.

FastSRM inherits from all the weaknesses of SRM including the fact that mixing matrices are assumed to be orthogonal, which is debattable in practice. Another desirable property of such decompositions is the spatial smoothness of the components. While it is often observed that SRM components are smooth, nothing imposes such a constraint in the model. We highlight that the optimal spatial decomposition is only optimal with respect to the SRM model. More precisely, it gives a reduced representation such that applying the SRM model on the reduced representation or on the full data gives the same result. Whether this representation is interpretable or not is irrelevant here, since we only focus on accelerating the SRM algorithm. However, as this representation is not unique, it may be interesting to find the "most interpretable" optimal spatial decomposition. Note that this would not change the end result of the SRM algorithm but could be of interest for interpretation purposes. It is an open question whether dimension reduction and learning of the shared components could be done jointly and efficiently without assuming orthogonal mixing matrices. Our approach based on optimal spatial decomposition shows that SRM-like methods do not need the full data to provide accurate estimates of shared components. We believe that such insights may guide the design of future methods.

Last, FastSRM is useful when the number of features is much larger than the number of samples. This setting is common in fMRI but less common in MEG/EEG where the number of samples is usually much larger than the number of features. Even in this case, it is possible to considerably speed up SRM. Indeed, SRM only depends on second order statistics. These statistics can be pre-computed making the rest of the pipeline insensitive to the number of samples in the dataset. An implementation of this simple idea would make SRM easily applicable to other modalities. We leave it to future work.

References

1 Naselaris T, Kay KN, Nishimoto S, Gallant JL. Encoding and decoding in fMRI. Neuroimage. 2011 May;56(2):400–410.

2 Huth AG, Nishimoto S, Vu AT, Gallant JL. A continuous semantic space describes the representation of thousands of object and action categories across the human brain. Neuron. 2012;76(6):1210–1224.

3 Güçlü U, van Gerven MA. Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. NeuroImage. 2017;145:329–336.

4 Richard H, Pinho A, Thirion B, Charpiat G. Optimizing deep video representation to match brain activity. Computational Cognitive Neuroscience. 2018;.

5 Poline JB, Brett M. The general linear model and fMRI: does love last forever? Neuroimage. 2012;62(2):871–880.

6 Jutten C, Herault J. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal processing. 1991;24(1):1–10.

7 Richard H, Ablin P, Thirion B, Gramfort A, Hyvarinen A. Shared Independent Component Analysis for Multi-Subject Neuroimaging. In: Advances in Neural Information Processing Systems 34; 2021. .

8 Richard H, Gresele L, Hyvarinen A, Thirion B, Gramfort A, Ablin P. Modeling Shared responses in Neuroimaging Studies through MultiView ICA. In: Advances in Neural Information Processing Systems 33; 2020. .

9 Varoquaux G, Sadaghiani S, Poline JB, Thirion B. CanICA: Model-based extraction of reproducible group-level ICA patterns from fMRI time series. arXiv preprint arXiv:09114650. 2009;.

10 Calhoun VD, Adali T, Pearlson GD, Pekar JJ. A method for making group inferences from functional MRI data using independent component analysis. Human brain mapping. 2001;14(3):140–151.

11 Vía J, Anderson M, Li XL, Adalı T. Joint blind source separation from second-order statistics: Necessary and sufficient identifiability conditions. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2011. p. 2520–2523.

12 Chen PH, Chen J, Yeshurun Y, Hasson U, Haxby J, Ramadge PJ. A reduced-dimension fMRI shared response model. In: Advances in Neural Information Processing Systems; 2015. p. 460–468.

13 Baldassano C, Chen J, Zadbood A, Pillow JW, Hasson U, Norman KA. Discovering event structure in continuous narrative perception and memory. Neuron. 2017;95(3):709–721.

14 Cohen JD, Daw N, Engelhardt B, Hasson U, Li K, Niv Y, et al. Computational approaches to fMRI analysis. Nature neuroscience. 2017;20(3):304–313.

15 Baldassano C, Hasson U, Norman KA. Representation of real-world event schemas during narrative perception. Journal of Neuroscience. 2018;38(45):9689–9699.

16 Jolly E, Sadhukha S, Chang LJ. Custom-molded headcases have limited efficacy in reducing head motion during naturalistic fMRI experiments. NeuroImage. 2020;222:117207.

17 Lee CS, Aly M, Baldassano C. Anticipation of temporally structured events in the brain. Elife. 2021;10:e64972.

18 Turek JS, Ellis CT, Skalaban LJ, Turk-Browne NB, Willke TL. Capturing shared and individual information in fmri data. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2018. p. 826–830.

19 Chen J, Leong YC, Honey CJ, Yong CH, Norman KA, Hasson U. Shared memories reveal shared structure in neural activity across individuals. Nature neuroscience. 2017;20(1):115–125.

20 Zhang H, Chen PH, Chen J, Zhu X, Turek JS, Willke TL, et al. A searchlight factor model approach for locating shared information in multi-subject fMRI analysis. arXiv preprint arXiv:160909432. 2016;.

21 Zhang H, Chen PH, Ramadge P. Transfer learning on fMRI datasets. In: International Conference on Artificial Intelligence and Statistics; 2018. p. 595–603.

22 Anderson MJ, Capota M, Turek JS, Zhu X, Willke TL, Wang Y, et al. Enabling factor analysis on thousand-subject neuroimaging datasets. In: 2016 IEEE International Conference on Big Data (Big Data). IEEE; 2016. p. 1151–1160.

23 Bellec P, Rosa-Neto P, Lyttelton OC, Benali H, Evans AC. Multi-level bootstrap analysis of stable clusters in resting-state fMRI. Neuroimage. 2010;51(3):1126–1139.

24 Dadi K, Varoquaux G, Machlouzarides-Shalit A, Gorgolewski KJ, Wassermann D, Thirion B, et al. Fine-grain atlases of functional modes for fMRI analysis. NeuroImage. 2020;221:117126.

25 Denning PJ. Thrashing: Its causes and prevention. In: Proceedings of the December 9-11, 1968, fall joint computer conference, part I; 1968. p. 915–922.

26 Bazeille T, Dupre E, Richard H, Poline JB, Thirion B. An empirical evaluation of functional alignment using inter-subject decoding. NeuroImage. 2021;p. 118683.

27 Haxby JV, Guntupalli JS, Connolly AC, Halchenko YO, Conroy BR, Gobbini MI, et al. A common, high-dimensional model of the representational space in human ventral temporal cortex. Neuron. 2011;72(2):404–416.

28 Yousefnezhad M, Zhang D. Deep hyperalignment. arXiv preprint arXiv:171003923. 2017;.

29 Li YO, Adali T, Wang W, Calhoun VD. Joint blind source separation by multiset canonical correlation analysis. IEEE Transactions on Signal Processing. 2009;57(10):3918–3929.

30 Bazeille T, Richard H, Janati H, Thirion B. Local optimal transport for functional brain template estimation. In: International Conference on Information Processing in Medical Imaging. Springer; 2019. p. 237–248.

31 Abraham A, Pedregosa F, Eickenberg M, Gervais P, Mueller A, Kossaifi J, et al. Machine learning for neuroimaging with scikit-learn. Frontiers in neuroinformatics. 2014;8:14.

32 Kumar M, Anderson MJ, Antony JW, Baldassano C, Brooks PP, Cai MB, et al. BrainIAK: The brain imaging analysis kit. Aperture. 2020;Available from: https://osf.io/db2ev/.

33 Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020 Sep;585(7825):357–362. Available from: https://doi.org/10.1038/s41586-020-2649-2.

34 Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. 2020;17:261–272.

35 Hunter JD. Matplotlib: A 2D graphics environment. Computing in science & engineering. 2007;9(3):90–95.

36 Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. the Journal of machine Learning research. 2011;12:2825–2830.

37 Schaefer A, Kong R, Gordon EM, Laumann TO, Zuo XN, Holmes AJ, et al. Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity MRI. Cerebral Cortex. 2017;28(9):3095–3114.

38 Mensch A, Mairal J, Thirion B, Varoquaux G. Extracting Universal Representations of Cognition across Brain-Imaging Studies. arXiv preprint arXiv:180906035. 2018;.

39 Chen J, Leong YC, Norman KA, Hasson U. Shared experience, shared memory: a common structure for brain activity during naturalistic recall. bioRxiv. 2016;Available from: https://www.biorxiv.org/content/early/2016/01/05/035931.

40 Kuhn HW. The Hungarian method for the assignment problem. Naval research logistics quarterly. 1955;2(1-2):83–97.

41 Guntupalli JS, Feilong M, Haxby JV. A computational model of shared fine-scale structure in the human connectome. PLoS computational biology. 2018;14(4):e1006120.

42 Nastase SA, Liu YF, Hillman H, Norman KA, Hasson U. Leveraging shared connectivity to aggregate heterogeneous datasets into a common response space. bioRxiv. 2019;Available from: https://www.biorxiv.org/content/early/2019/08/21/741975.

A Detailed derivation of the Probabilistic SRM algorithm

Here we describe in detail the log-likeihood underlying the Probabilistic SRM and derive the update rules. We highlight that the derivations used to obtain these updates are not new. They are obtained following the strategy introduced in [22] and [12].

Denoting $\mathbb{V}[\mathbf{s}|\mathbf{x}] = (\sum_i \frac{1}{\sigma_i^2} I + \Sigma_s^{-1})^{-1}$ and $\mathbb{E}[\mathbf{s}|\mathbf{x}] = \mathbb{V}[\mathbf{s}|\mathbf{x}] \sum_i \frac{1}{\sigma_i^2} A_i^\top \mathbf{x}_i$, we have

$$
\begin{aligned}
p(\mathbf{x}, \mathbf{s}) &= \prod_i \frac{\exp\left(-\frac{\|\mathbf{x}_i - A_i \mathbf{s}\|^2}{2\sigma_i^2}\right)}{(2\pi\sigma_i^{2v})^{\frac{1}{2}}} \frac{\exp(-\frac{1}{2}\langle \mathbf{s}, \Sigma_s^{-1}\mathbf{s}\rangle)}{(2\pi|\Sigma_s|)^{\frac{1}{2}}} && (12)\\
&= c_1 \exp(-\frac{1}{2}\left(\sum_i \frac{1}{\sigma_i^2}\|\mathbf{x}_i\|^2 - 2\langle \sum_i \frac{1}{\sigma_i^2} A_i^\top \mathbf{x}_i, \mathbf{s}\rangle\right. && (13)\\
&\left. + \sum_i \frac{1}{\sigma_i^2}\|\mathbf{s}\|^2 + \langle \mathbf{s}, \Sigma_s^{-1}\mathbf{s}\rangle\right)) && (14)\\
&= c_2(\mathbf{x}) \exp(-\frac{1}{2}\left(\langle \mathbf{s} - \mathbb{E}[\mathbf{s}, \mathbf{x}], \mathbb{V}[\mathbf{s}|\mathbf{x}]^{-1}(\mathbf{s} - \mathbb{E}[\mathbf{s}|\mathbf{x}])\rangle\right)) && (15)
\end{aligned}
$$

where

$$
c_1 = \frac{1}{(2\pi\sigma_i^{2v})^{\frac{1}{2}}} \frac{1}{(2\pi|\Sigma_s|)^{\frac{1}{2}}} \tag{16}
$$

and

$$
c_2(\mathbf{x}) = c_1 \exp\left(-\frac{1}{2}\left(\sum_i \frac{1}{\sigma_i^2}\|\mathbf{x}_i\|^2 - \langle \mathbb{E}[\mathbf{s}, \mathbf{x}], \mathbb{V}[\mathbf{s}|\mathbf{x}]^{-1}\mathbb{E}[\mathbf{s}|\mathbf{x}]\rangle\right)\right) \tag{17}
$$

are independent from $\mathbf{s}$. Therefore,

$$
\mathbf{s}|\mathbf{x} \sim \mathcal{N}(\mathbb{E}[\mathbf{s}|\mathbf{x}], \mathbb{V}[\mathbf{s}, \mathbf{x}]) \tag{18}
$$

The negative expected completed log-likelihood is given by

$$
\mathcal{L} = \sum_i \frac{1}{2} v \log(\sigma_i^2) + \frac{1}{2\sigma_i^2}\mathbb{E}[\|\mathbf{x}_i - A_i\mathbf{s}\|^2] \tag{19}
$$

updates are therefore given by:

$$
\begin{aligned}
\sigma_i^2 &\leftarrow \frac{1}{v}(\mathbb{E}[\|\mathbf{x}_i - A_i\mathbb{E}[\mathbf{s}|\mathbf{x}]\|^2] + \|\mathrm{diag}(\mathbb{V}[\mathbf{s}|\mathbf{x}])\|) && (20)\\
A_i &\leftarrow \mathcal{P}(\mathbb{E}[\mathbf{x}_i \mathbb{E}[\mathbf{s}|\mathbf{x}]^\top]) && (21)\\
\Sigma_s &\leftarrow \mathbb{V}[\mathbf{s}|\mathbf{x}] + \mathbb{E}[\mathbb{E}[\mathbf{s}|\mathbf{x}]\mathbb{E}[\mathbf{s}|\mathbf{x}]^\top] && (22)
\end{aligned}
$$

It is useful to access the log-likelihood to check the implementation of the algorithm and monitor the convergence. From equation (15), the likelihood is given by:

$$p(\mathbf{x}) = c_2(\mathbf{x}) \int_{\mathbf{s}} \exp(-\frac{1}{2} \left(\langle \mathbf{s} - \mathbb{E}[\mathbf{s}, \mathbf{x}], \mathbb{V}[\mathbf{s}|\mathbf{x}]^{-1} (\mathbf{s} - \mathbb{E}[\mathbf{s}|\mathbf{x}])\rangle\right)) d\mathbf{s} \tag{23}$$

$$= c_2(\mathbf{x}) (2\pi |\mathbb{V}[\mathbf{s}|\mathbf{x}]|)^{\frac{1}{2}} \tag{24}$$

replacing $c_2(\mathbf{x})$ by its expression and taking the log, the expected negative log-likelihood is (up to constants) given by:

$$\begin{aligned}\mathbb{E}[-\log(p(\mathbf{x}))] &= \sum_i \frac{v}{2} \log(\sigma_i^2) + \frac{1}{2} \log(|\Sigma_s|) - \frac{1}{2} \log(|\mathbb{V}[\mathbf{s}|\mathbf{x}]|) \\ &+ \sum_i \frac{1}{2} \frac{1}{\sigma_i^2} \mathbb{E}[\|\mathbf{x}_i\|^2] - \frac{1}{2} \mathbb{E}[\langle \mathbb{E}[\mathbf{s}, \mathbf{x}], \mathbb{V}[\mathbf{s}|\mathbf{x}]^{-1} \mathbb{E}[\mathbf{s}|\mathbf{x}] \rangle]\end{aligned} \tag{25}$$

B Identifiability results on Probabilistic SRM available in [8] appendix D

Proposition 3. *Probabilistic SRM with positive definite Σ_s is not identifiable*

Proof. Let us assume the data $\mathbf{x}_i \;\; i = 1, \dots, m$ follow the Probabilistic SRM model with parameters $\Sigma_s, A_i, \sigma_i^2 \;\; i = 1, \dots, m$ and Σ_s definite positive.

Let us consider an orthogonal matrix $O \in \mathcal{O}_k$. We call $A_i' = A_i O$ and $\Sigma_s' = O^\top \Sigma_s O$. Σ_s' is trivially symmetric positive definite.

Then the data also follows the SRM model with different parameters $\Sigma_s', A_i', \sigma_i^2 \;\; i = 1, \dots, m$. □

Proposition 4. *We consider the Probabillistic SRM model where Σ_s is a positive* diagonal *matrix. We further assume that the values in Σ are all distinct and ranked in ascending order. Then, the Probabilistic SRM model is identifiable up to sign indeterminacies on the columns of $\begin{bmatrix} A^1 \\ \vdots \\ A^m \end{bmatrix}$.*

Proof. The Probabilistic SRM can be written

$$\mathbf{x}_i \sim \mathcal{N}(0, A_i \Sigma_s {A_i}^\top + \sigma_i^2 I_v) \text{ with } {A_i}^\top A_i = I_p$$

where Σ_s is a positive diagonal matrix with distincts values ranked in ascending order.

Let us assume the data $\mathbf{x}^i \;\; i = 1, \dots, m$ follow the Probabilistic SRM model with parameters $\Sigma_s, A_i, {\sigma_i}^2 \;\; i = 1, \dots, m$. Let us further assume that the data $\mathbf{x}^i \;\; i = 1, \dots, m$ follow the Probabilistic SRM model with an other set of parameters $\Sigma_s', A_i', {\sigma_i'}^2 \;\; i = 1, \dots, m$.

Since the model is Gaussian, we look at the covariances. We have for $i \neq j$

$$\mathbb{E}[\mathbf{x}_i \left(\mathbf{x}_j\right)^\top] = A_i \Sigma_s {A_j}^\top = A'_i \Sigma'_s {A'_j}^\top \quad ,$$

The singular value decomposition is unique up to sign flips and permutation. Since eigenvalues are positive and ranked the only indeterminacies left are on the eigenvectors. For each eigenvalue a sign flip can occur simultaneously on the corresponding left and right eigenvector.

Therefore we have $\Sigma'_s = \Sigma_s$, $A_i = A'_i D_{ij}$ and $A_j = A'_j D_{ij}$ where $D_{ij} \in \mathbb{R}^{k,k}$ is a diagonal matrix with values in $\{-1, 1\}$. This analysis holds for every $j \neq i$ and therefore $D_{ij} = D$ is the same for all subjects.

We also have for all i

$$\mathbb{E}[\mathbf{x}_i \left(\mathbf{x}_i\right)^\top] = A_i \Sigma_s {A_i}^\top + \sigma_i^2 I_v = A'_i \Sigma'_s {A'_i}^\top + \sigma_i^2 I_v$$

We therefore conclude $\sigma_i^2 = \sigma_i^2, i = 1 \dots m$.

Note that if the diagonal subject specific noise covariance $\sigma_i^2 I_v$ is replaced by any positive definite matrix, the model still enjoys identifiability. □

C Proofs of Proposition 1

Proof. Updates of the mixing matrices A_i in deterministic SRM equation (5) can be written:

$$A_i \leftarrow \mathcal{P}\left(\frac{1}{n} X_i S^\top\right) = U_{X_i} \mathcal{P}\left(\frac{1}{n} Z_i S^\top\right) \tag{26}$$

where $\mathcal{P}$ is the projection on the Stiefel manifold: $\mathcal{P}(M) = M(M^\top M)^{-\frac{1}{2}}$.

Therefore we can look for A_i as $A_i = U_{X_i} \tilde{A}_i$. We can show that $\tilde{A}_i$ is orthogonal. Indeed,

$$\begin{aligned} & A_i^\top A_i = I_p && (27) \\ \implies & \tilde{A}_i^\top U_{X_i}^\top U_{X_i} \tilde{A}_i = I_p && (28) \\ \implies & \tilde{A}_i^\top \tilde{A}_i = I_p && (29) \end{aligned}$$

Then, we use the fact that

$$\|X_i - A_i S\|^2 = \|U_{X_i} Z_i - U_{X_i} \tilde{A}_i S\|^2 = \|Z_i - \tilde{A}_i S\|^2 \tag{30}$$

so that $A'_i = \tilde{A}_i$.

Therefore, the solution of deterministic SRM on data $(\mathbf{z}_i)_{i=1}^m$ and $(\mathbf{x}_i)_{i=1}^m$ are linked by the change of parameters $A_i = U_{\mathbf{x}_i} A'_i$ and $S = S'$. This concludes the proof.

□

D Proofs of Proposition 2

Proof. Updates of the mixing matrices A_i in probabilistic SRM equation (21) can be written:

$$A_i \leftarrow U_{\mathbf{x}_i}\mathcal{P}(\mathbb{E}[\mathbf{z}_i\mathbb{E}[\mathbf{s}|\mathbf{x}_i]^T]) \tag{31}$$

so we can look for A_i as $A_i = U_{\mathbf{x_i}}\tilde{A}_i$ and, as in the deterministic case, $\tilde{A}_i$ is orthogonal. Therefore equality (30) holds.

Then we consider the expected negative log-likelihood of probabilistic srm:

$$\begin{aligned}\mathcal{L} &= \sum_i \frac{1}{2}v\log(\sigma_i^2) + \frac{1}{2}\log(|\Sigma_s|) + \mathbb{E}[\int_{\mathbf{s}}\sum_i \frac{1}{2\sigma_i^2}\|\mathbf{x}_i - A_i\mathbf{s}\|^2 \\ &\quad + \frac{1}{2}\langle\mathbf{s}, \Sigma_s^{-1}\mathbf{s}\rangle d\mathbf{s}] &(32)\\ &= \sum_i \frac{1}{2}v\log(\sigma_i^2) + \frac{1}{2}\log(|\Sigma_s|) + \mathbb{E}[\int_{\mathbf{s}}\sum_i \frac{1}{2\sigma_i^2}\|\mathbf{z}_i - \tilde{A}_i\mathbf{s}\|^2 \\ &\quad + \frac{1}{2}\langle\mathbf{s}, \Sigma_s^{-1}\mathbf{s}\rangle d\mathbf{s}] &(33)\end{aligned}$$

where we use equality (30). Optimizing the log-likelihood via expectation maximization yields the exact same updates as probabilistic SRM on data $\mathbf{z}_i$ except that updates

$$\sigma_i^2 \leftarrow \frac{1}{n}(\mathbb{E}[\|\mathbf{z}_i - \tilde{A}_i\mathbb{E}[\mathbf{s}|\mathbf{z}]\|^2] + \|\mathrm{diag}(\mathbb{V}[\mathbf{s}|\mathbf{z}])\|^2) \tag{34}$$

are replaced by updates

$$\sigma_i^2 \leftarrow \frac{1}{v}(\mathbb{E}[\|\mathbf{z}_i - \tilde{A}_i\mathbb{E}[\mathbf{s}|\mathbf{z}]\|^2] + \|\mathrm{diag}(\mathbb{V}[\mathbf{s}|\mathbf{z}])\|^2) \tag{35}$$

so that $\tilde{A}_i = A'_i$. Therefore, the updates in both algorithms are linked by $A_i = U_{\mathbf{x}_i}A'_i$, $\sigma'_i = \sigma_i$ and $\Sigma'_s = \Sigma_s$. This concludes the proof. □