Multi-Task Learning for Subspace Segmentation

Yu Wang * YW323@CAM.AC.UK
David Wipf † DAVIDWIP@MICROSOFT.COM
Qing Ling ♯ QINGLING@MAIL.USTC.EDU.CN
Wei Chen * WC253@CAM.AC.UK
Ian Wassell * IJW24@CAM.AC.UK
∗ Computer Laboratory, University of Cambridge, Cambridge, UK
† Microsoft Research, Beijing, China
♯ University of Science and Technology of China, Hefei, Anhui, China

Abstract

Subspace segmentation is the process of clustering a set of data points that are assumed to lie on the union of multiple linear or affine subspaces, and is increasingly being recognized as a fundamental tool for data analysis in high dimensional settings. Arguably one of the most successful approaches is based on the observation that the sparsest representation of a given point with respect to a dictionary formed by the others involves nonzero coefficients associated with points originating in the same subspace. Such sparse representations are computed independently for each data point via ℓ_1-norm minimization and then combined into an affinity matrix for use in a final spectral clustering step. The downside of this procedure is two-fold. First, unlike canonical compressive sensing scenarios with ideally-randomized dictionaries, the data-dependent dictionaries here are unavoidably highly structured, disrupting many of the favorable properties of the ℓ_1 norm. Secondly, by treating each data point independently, we ignore useful relationships between points that can be leveraged for jointly computing such sparse representations. Consequently, we motivate a multi-task learning-based framework for learning coupled sparse representations leading to a segmentation pipeline that is both robust against correlation structure and tailored to generate an optimal affinity matrix. Theoretical analysis and empirical tests are provided to support these claims.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

1. Introduction

Principal component analysis is a classical method for finding a low-dimensional linear subspace that captures the majority of variance in a particular centered data set. Estimating this subspace and projecting onto it are trivial matters using a simple eigen-decomposition. However, suppose the data instead lie on or near the union of multiple low-dimensional subspaces. In this revised scenario the goal is to both estimate each of these subspaces and segment every data point into the closest one, a considerably more challenging proposition with no closed-form solution. Note that this problem is quite different than traditional clustering, where the objective is to find groups of points that are closest to one another, rather than closest to or members of some subspace. This so-called subspace clustering or subspace segmentation problem (these terms are used interchangeably) is relevant to numerous machine learning and computer vision applications, including image representation and compression (Hong et al., 2006), motion segmentation (Rao et al., 2010), and face clustering (Liu et al., 2013).

We define the subspace segmentation problem more formally as follows. Let $\{\mathcal{S}_k\}_{k=1}^K$ represent a collection of K linear subspaces in $\mathbb{R}^D$, where $\dim[\mathcal{S}_k] = D_k$ for all k. Now consider that we have a set of N points $\{\boldsymbol{x}_j\}_{j=1}^N$ that have been sampled from this union of subspaces, with N_k samples drawn from subspace k. We then let $\boldsymbol{X}_k$ denote the $D \times N_k$ matrix of points associated with the respective subspace. The entire constellation of points can be expressed as

$$\boldsymbol{X} \triangleq [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K]\boldsymbol{P} \in \mathbb{R}^{D \times N}, \quad (1)$$

where $\boldsymbol{P} \in \mathbb{R}^{N \times N}$ is some unknown permutation matrix. The ultimate objective of subspace segmentation is to learn the subspace membership of each $\boldsymbol{x}_j$, without any prior knowledge of the cardinality or dimensionality of the underlying union of subspaces. Extensions to affine subspaces is straightforward and will be discussed later.

Spectral clustering represents arguably the most popular and robust recent method for subspace segmentation. The basic idea proceeds by forming an affinity matrix $\boldsymbol{A}$, where the ij-th element a_{ij} quantifies the strength of the relationship between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. In traditional clustering, this affinity is commonly computed using a Gaussian kernel $\exp[-\alpha\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2]$ with $\alpha > 0$, but this ignores the subspace structure we seek to capture.

In contrast, a more effective construction of $\boldsymbol{A}$ exploits the *self-expressiveness property* of $\boldsymbol{X}$ (Elhamifar & Vidal, 2013), namely that any $\boldsymbol{x}_j$ can be expressed as a linear combination of other data points in $\boldsymbol{X}$ within the same subspace, assuming suitable sampling of each subspace (i.e., each N_k is sufficiently large with points in general position). In general, with $N > D$ there will exist an infinite number of such representations; however, in forming a viable affinity matrix it is paramount that we find representations that heavily favor *only* using points from the same subspace. From an optimization standpoint, this involves solving

$$\min_{\boldsymbol{Z}} f(\boldsymbol{Z}) \;\text{ s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z}, \tag{2}$$

over the $N \times N$ coefficient matrix $\boldsymbol{Z}$. Ideally the penalty function f will encourage an optimal solution $\boldsymbol{Z}^*$ to avoid the degeneracy $\boldsymbol{Z}^* = \boldsymbol{I}$ and be such that $\boldsymbol{P}^{-1}\boldsymbol{Z}^*$ is block diagonal, with blocks aligned and sized according to the columns of each $\boldsymbol{X}_k$.

We may then form a symmetric affinity matrix as

$$\boldsymbol{A} = |\boldsymbol{Z}^*| + |\boldsymbol{Z}^*|^\top \tag{3}$$

and apply traditional spectral clustering (Luxburg, 2007) to the normalized Laplacian of $\boldsymbol{A}$ to recover the underlying subspace segmentation. In brief, if we view data points $\boldsymbol{x}_j$ as nodes on a graph and elements of $\boldsymbol{A}$ as edge weights between them, then spectral clustering estimates the number of connected components in the graph, as well as which data points are connected.[1] Assuming $\boldsymbol{Z}^*$ produces the correct block-diagonal structure, or approximately so, similar points will naturally be grouped together, and we can expect to correctly learn which data points belong to each $\mathcal{S}_k$. If noise or outliers are present, we can also relax the equality constraint in (2) with an appropriate, application-specific data-fidelity term (Elhamifar & Vidal, 2013; Liu et al., 2013).

The differentiating factor in state-of-the-art spectral clustering algorithms applied to subspace segmentation is primarily in how the function f is chosen. One influential class of algorithms called *sparse subspace clustering* (SSC) selects the sparsity-promoting penalty such as $f(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_1 \triangleq \sum_j \|\boldsymbol{z}_j\|_1$, along with the additional constraint $\text{diag}[\boldsymbol{Z}] = 0$ (Elhamifar & Vidal, 2013). The latter is required to explicitly prevent $\boldsymbol{Z} = \boldsymbol{I}$ (it can also be embedded in the function f instead if preferred). Drawing on related ideas from compressive sensing, the motivation here is that the sparsest solution to each individual constraint $\{\boldsymbol{x}_j = \boldsymbol{X}\boldsymbol{z}_j, z_{jj} = 0\}$ will involve all or most nonzero coefficients belonging to the same subspace, or equivalently, $\boldsymbol{P}^{-1}\boldsymbol{z}_j$ will be block-sparse with favorable alignment.

An alternative proposal, frequently referred to as *low-rank represention* (LRR), suggests penalizing $\text{rank}[\boldsymbol{Z}]$, in part because $\text{rank}[\boldsymbol{Z}]$ will be small if each subspace dimension D_k is sufficiently low permitting a low-rank feasible solution, and the degenerate full-rank solution $\boldsymbol{Z}^* = \boldsymbol{I}$ can naturally be avoided (Liu et al., 2013). Alternatively, relaxation of the rank to the convex nuclear norm $f(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_*$ is a popular surrogate. Notably, if the subspaces are independent,[2] then $\boldsymbol{P}^{-1}\boldsymbol{Z}^*$ will provably display the desired block-diagonal structure (Liu et al., 2013). Beyond LRR and SSC, more recent methods have been proposed based on the Frobenius norm (Lu et al., 2012), the Trace Lasso (Lu et al., 2013), and non-convex constraints applied directly to the affinity matrix $\boldsymbol{A}$ to rigidly enforce block sparsity (Feng et al., 2014). Although not our central focus here, many additional variants of these have been suggested for tackling outliers.

All of the above have pros and cons, and there remain important potential directions for improving the state-of-the-art in what amounts to a widely-applicable data analysis tool. Similar to SSC we will herein investigate a penalty function that explicitly favors block sparsity in each column of $\boldsymbol{Z}$. However, unlike any existing SSC algorithm which learns each column completely independently, we propose to apply a Bayesian multi-task learning formulation such that columns from the same subspace are linked during the estimation process given their obvious similarities. Against this backdrop our primary contributions are threefold:

- Analysis of intrinsic limitations of existing spectral clustering approaches to subspace segmentation (Section 2).
- Reframing of subspace segmentation as a principled multi-task learning (MTL) problem (Section 3)
- Theoretical and empirical examination of a novel Bayesian MTL pipeline (Sections 4 and 5).

2. Limitations of Current Methods

Sparse Subspace Clustering (SSC): The canonical motivating form of SSC involves solving

$$\min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_0 \;\text{ s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z},\; \text{diag}[\boldsymbol{Z}] = 0, \tag{4}$$

[1] Two nodes of the graph are connected if there exists a path with nonzero edge weights between them, which will typically occur in the present context due to the self-expressiveness property described above.

[2] A union of subspaces is independent if $\dim[\bigoplus_{k=1}^K \mathcal{S}_k] = \sum_{k=1}^K \dim[\mathcal{S}_k]$, where $\bigoplus$ denotes the direct sum.

where $\|\boldsymbol{Z}\|_0$ is the matrix ℓ_0 norm. As long as each individual subspace satisfies $D_k < D$ for all k, and sampled points are sufficiently dense in general position, then the solution to (4) will be block diagonal and aligned with the true clusters as desired. From a practical standpoint, based on theoretical analysis from (Elhamifar & Vidal, 2013; Mahdi & Candes, 2012), in certain cases as long as the angles between subspaces are not too small, then we can replace the NP-hard matrix ℓ_0-norm minimization in (4) with $\|\boldsymbol{Z}\|_1$ and still expect this same desirable block-diagonal structure.

However, as well-established in the literature on compressive sensing, the equivalence between the maximally sparse ℓ_0-norm solution and the ℓ_1-norm substitution is quite sensitive to correlations in the data. When we are free to choose a design matrix with randomized sampling as in compressive sensing, this is not a problem; however, in the subspace segmentation problem $\boldsymbol{X}$ is likely to have rich structure and strong correlations, *which is why we want to cluster it in the first place*. Therefore, it seems premature to adopt the popular convex relaxation mantra in a problem domain that deviates substantially from the theoretical realm where ℓ_1-norm-based sparse estimation is motivated to begin with.

A second difficulty with SSC applies even if the original ℓ_0 norm is used to solve (4), and similarly with any other standard, element-wise separable sparsity-promoting penalty function. The problem is that even if we achieve a perfect block-diagonal structure in some estimated $\boldsymbol{P}^{-1}\hat{\boldsymbol{Z}}$, we still are not guaranteed to arrive at the optimal segmentation after the final spectral clustering step. This is because *within* each block there can exist latent disconnected components such that intrinsic cluster memberships are falsely estimated. More quantitatively, the affinity matrix defines weighted edges between N nodes (data-points) as stated previously. To guarantee the correct subspace segmentation by spectral clustering, we require that within each block, all points are fully connected, implying that there is a path with non-zero edge weights between *every* node within the block (Luxburg, 2007). The core issue regarding SSC is that, although sparsity (or all zero-valued elements of $\boldsymbol{Z}$) *between* blocks is desirable, the lack of sparsity *within* blocks can potentially disrupt spectral clustering as pointed out in (Elhamifar & Vidal, 2013).

Low Rank Representations (LRR): Although practically successful, from a high-level conceptual standpoint LRR suffers in the sense that low-rank solutions are often provably ineffectual. For example, consider the scenario where $M \triangleq \dim[\bigoplus_{k=1}^{K} \mathcal{S}_k] > D$, meaning that the direct sum of all subspaces is greater than the ambient dimension. Assuming sufficient sampling in each subspace, this implies that $\text{rank}[\boldsymbol{X}] = D$, or full row rank. Then the optimal solution to the LRR problem

$$\min_{\boldsymbol{Z}} \text{rank}[\boldsymbol{Z}] \;\; \text{s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z} \tag{5}$$

will typically *not* (even approximately) produce a block diagonal $\boldsymbol{P}^{-1}\boldsymbol{Z}^*$. In fact, we can always find some optimal $\boldsymbol{Z}^*$ with $\text{rank}[\boldsymbol{Z}^*] = D$. However, any feasible block-diagonal solution $\boldsymbol{Z}'$ aligning with the true clusters must satisfy $\text{rank}[\boldsymbol{Z}'] \geq M > D$, and hence cannot be optimal. Therefore, solving (5) will not lead to a useful result. In practice however, LRR algorithms typically relax $\text{rank}[\boldsymbol{Z}]$ to $\|\boldsymbol{Z}\|_*$. It can be shown that the resulting optimal closed-form solution then becomes $\boldsymbol{Z}^* = \boldsymbol{V}\boldsymbol{V}^\top$, where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the abbreviated svd of $\boldsymbol{X}$ (Liu et al., 2013). Interestingly, this is likewise the optimal solution to (5) when $\|\boldsymbol{Z}\|_{\mathcal{F}}$ is used instead.[3] Given that the Frobenius norm is well known to encourage non-sparse solutions, we may expect that either selection will lead to undesirable, strong off-block-diagonal elements in $\boldsymbol{P}^{-1}\boldsymbol{Z}^*$. There are of course several modifications of the LRR paradigm, including the inclusion of non-convex surrogate rank functions (Babacan et al., 2012), but they possess similar limitations, and closer approximations to the rank function may actually perform worse. Overall, while the ℓ_0 norm represents a viable, if not directly-computable target for SSC, the rank function does not occupy a similar role with LRR.

Other Regularization Methods: Although the selection $f(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_{\mathcal{F}}^2 = \sum_j \|\boldsymbol{z}_j\|_2^2$ does not produce block-sparsity as discussed above, it does nonetheless exploit a grouping effect that can act as a significant advantage over SSC. In brief, the ℓ_2 norm tends to assign similar values to coefficients associated with correlated columns of a design matrix (in this case $\boldsymbol{X}$). Therefore, assuming some degree of correlation between the columns of each $\boldsymbol{X}_k$, we may expect that coefficients associated with the same subspace are likely to share significant, nonzero magnitudes (at least to the extent that intra-subspace correlations are appreciable), leading to full intra-subspace connections on the associated graph as desired. To the extent which this dense representation within blocks can outweigh the lack of sparsity outside of blocks, this so-called *least squares represenation* (LSR) method can potentially outperform SSC (equivalently for LRR in the noiseless case).

To directly address the trade-off between the grouping effect and block sparsity, it has been proposed (Lu et al., 2013) to replace the objective in (4) with

$$f(\boldsymbol{Z}) = \sum_j \|\boldsymbol{z}_j\|_{TL}, \quad \|\boldsymbol{z}\|_{TL} \triangleq \|\boldsymbol{X}\text{diag}[\boldsymbol{z}]\|_*. \tag{6}$$

Here $\|\cdot\|_{TL}$ denotes the *Trace Lasso* norm which, roughly speaking, interpolates between the ℓ_1 and ℓ_2 norms depend-

[3] Note however that this equivalence no longer holds once we relax to the equality constraint and allow for differentiating error penalties.

ing on the correlation structure of $\boldsymbol{X}$ (Grave et al., 2011). If $\boldsymbol{X}$ has highly correlated columns, the Trace Lasso norm behaves like the ℓ_2 norm, while for nearly uncorrelated columns it is similar to ℓ_1. In the context of subspace clustering, it is argued that this refined penalization then can hopefully support both the grouping effect and block sparse structure, assuming points within the same subspace are more correlated than others.

The resulting algorithm, called *correlation-adaptive subspace segmentation* (CASS) unfortunately has a few lingering problems. First, unlike all of the algorithms described above, CASS is extremely sensitive to transformations of the data. In particular, if $\boldsymbol{X} \rightarrow \boldsymbol{Q}\boldsymbol{X}$ for an arbitrary invertible matrix $\boldsymbol{Q}$, then the feasible set $\boldsymbol{Q}\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{X}\boldsymbol{Z}$ is unchanged, and so any optimization problem of the form of (2) is unchanged provided that f is independent of $\boldsymbol{X}$. However, when f is given by (6), the effective penalty function will change substantially as $\boldsymbol{Q}$ is varied. A second related issue with CASS is that in many important problem domains the latent subspaces are *all* highly correlated due to the intrinsic geometry of the underlying applications, e.g., face clustering and motion segmentation (Elhamifar & Vidal, 2013). In this regime, CASS behaves very much like LSR as the Trace Lasso converges to nearly the ℓ_2 norm across all columns of $\boldsymbol{X}$.

Lastly, we close this section with one additional algorithm that implicitly attempts to enforce both block-sparsity and the grouping effect simultaneously (Feng et al., 2014). This procedure is presented as a modified form of either LRR or SSC; in both cases an additional non-convex constraint is applied enforcing the resulting affinity matrix $\boldsymbol{A}$ to have exactly K connected components (this disallows disconnected components within a subspace block). While this proposal is interesting, unlike all of the methods described above, it requires explicit prior knowledge of the cluster number K, and moreover, the provided code also assumes that $D_k = C$ for some constant C (all subspaces have the same dimension), and that this C be provided to the algorithm as well. Additionally, if $\boldsymbol{X}$ does not satisfy some RIP-like conditions, which arguably will not hold for subspace segmentation, the resulting algorithm is not even guaranteed to reduce the underlying cost function at each iteration.

3. Multi-Task Learning for Subspace Segmentation

Subspace clustering algorithms and analysis have largely been driven by the compressive sensing and signal processing communities, as well as certain computer vision applications. However, while not previously explored in this context, the gist of subspace segmentation, and the search for dense block-diagonal structure in the relevant affinity matrix, can be viewed as a multi-task learning (MTL) problem, treating each data point $\boldsymbol{x}_j$ as an individual task.

Motivating Principles for MLT: Define $\bar{\boldsymbol{X}}_j$ as $\boldsymbol{X}$ with the j-th column set to zero; this modification effectively allows us to remove the constraint $\text{diag}[\boldsymbol{Z}] = 0$ defined previously for SSC (assuming any penalty that favors zero). Then for each $\boldsymbol{x}_j$ we first consider the following task-specific optimization problem

$$\min_{\boldsymbol{z}_j} g(\boldsymbol{z}_j) \;\; \text{s.t. } \boldsymbol{x}_j = \bar{\boldsymbol{X}}_j \boldsymbol{z}_j. \tag{7}$$

Current SSC algorithms (as well as LSR and CASS) all decompose into (7) for each task $\boldsymbol{x}_j$, and the solutions are computed completely independently of one another and then later merged together to form $\boldsymbol{A}$. However, for tasks associated with the same subspace, solutions to (7) should ideally be very similar. More concretely, if $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ are both from $\mathcal{S}_k$, then we prefer that they both rely primarily on columns from $\boldsymbol{X}_k$ for their respective block-sparse representations. While we do not *a priori* know which tasks should be clustered, we can however exploit the existence of *some* underlying clustering, and it is here that MTL becomes especially relevant. In fact, as we will demonstrate shortly, MTL can jointly serve two distinct purposes.

1. By jointly estimating task representations $\boldsymbol{z}_j$, MTL can increase the chances that ideal block-sparse representations are found (no tasks from different subspaces are connected on the graph).

2. MTL can help to ensure that within a block all tasks are fully connected, facilitating the final spectral clustering step (dense representations within blocks).

We then motivate MTL as follows. Suppose we somehow knew the number of clusters K. Then let the set $\{\Omega_k\}_{k=1}^K$, with each $\Omega_k \subset \{1, \ldots, N\}$, denote a partitioning such that $\bigcup_{k=1}^K \Omega_k = \{1, \ldots, N\}$ and $\Omega_k \cap \Omega_{k'} = \emptyset$ for all pairs $\{k, k'\}$. Also let $\boldsymbol{Z}_{\Omega_k}$ represent the columns of $\boldsymbol{Z}$ indexed by Ω_k. Now consider the joint optimization over all tasks

$$\min_{\boldsymbol{Z}, \{\Omega_k\}} \sum_k h(\boldsymbol{Z}_{\Omega_k}) \;\; \text{s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z}, \text{diag}[\boldsymbol{Z}] = 0, \tag{8}$$

where the function h returns the number of nonzero *rows* in $\boldsymbol{Z}_k$ (a row-wise generalization of the ℓ_0 norm) and the set $\{\Omega_k\}$ is optimized over all possible index partitions per the above description. Under very minor conditions (the *generic subspace model* as defined in the next section), it can be shown that the resulting optimal solution $\boldsymbol{Z}^*$ and associated partition $\{\Omega_k^*\}$ will be such that, for all k, $h(\boldsymbol{Z}^*_{\Omega_k^*}) = D_k + 1$, with all nonzero rows aligned with the respective columns of $\boldsymbol{X}_k$. Moreover, all N_k indices within each Ω_k^* are connected in the resulting affinity matrix.[4]

[4]This occurs because a minimum of D_k points are needed to

The above is sufficient to guarantee that spectral clustering using the resulting $\boldsymbol{A}$ will produce the correct subspace segmentation (Luxburg, 2007).

Although solving (8) is NP-hard, and K is typically not known, a Bayesian MTL algorithm based on Dirichlet process priors (DP) has previously been designed to accomplish something related, albeit in a more general context (Qi et al., 2008). The advantage of this procedure, when adapted to the subspace segmentation problem structure, is that the DP prior putatively allows the algorithm to implicitly learn the value of K. The downside though is that the model is justified purely based on the validity of a presumed hierarchical probabilistic structure and the qualitative effectiveness of subsequent variational mean-field approximations required for inference purposes. Consequently, while high-level motivating principles may be similar, the actual connection to (8) remains tenuous.

Approximation for Subspace Segmentation: In the context of generic compressive sensing, we have previously demonstrated that the DP-based algorithm from above can be recast using alternative variational techniques into a much simpler form that is both considerably more transparent and amenable to analysis (Wang et al., 2015). Ultimately this reformulation will allow us to reveal a deeper connection with (8) and lead to a principled algorithmic adaptation specialized to subspace segmentation problems.

This revised model begins with a Gaussian likelihood

$$p(\boldsymbol{X}|\boldsymbol{Z}) \propto \prod_j \exp\left[-\frac{1}{2\nu}\left\|\boldsymbol{x}_j - \bar{\boldsymbol{X}}_j \boldsymbol{z}_j\right\|_2^2\right]. \quad (9)$$

Here we will assume that the noise variance ν is known (although it can be learned as well). For the prior distribution on $\boldsymbol{Z}$ we build upon the basic sparse Bayesian learning (SBL) framework from (Tipping, 2001), which in the present circumstances would involve a zero-mean Gaussian with an independent diagonal covariance for each column $\boldsymbol{z}_j$; however, this would not allow for task clustering. Instead we assume the prior distribution

$$p(\boldsymbol{Z}|\boldsymbol{\Lambda}, \boldsymbol{W}) \propto \prod_j \exp\left[-\frac{1}{2}\boldsymbol{z}_j^\top \Gamma_j^{-1} \boldsymbol{z}_j\right], \quad (10)$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ are hyperparameter matrices; $\boldsymbol{\Lambda}$ is constrained to have all non-negative elements, $\boldsymbol{W} \in \Omega$ is defined such that each column $\boldsymbol{w}_j$ is an element of the probability simplex, i.e.,

$$\Omega \triangleq \{\boldsymbol{w}_j : \sum_i w_{ij} = 1, w_{ij} \in [0,1]\}. \quad (11)$$

With some abuse of notation, we say that $\boldsymbol{W} \in \Omega$ if every column $\boldsymbol{w}_j \in \Omega$. Finally, $\boldsymbol{\Gamma}_j$ is the diagonal covariance matrix produced via

$$\boldsymbol{\Gamma}_j^{-1} = \sum_i w_{ij} \boldsymbol{\Lambda}_i^{-1}, \quad (12)$$

where $\boldsymbol{\Lambda}_i$ is defined as a diagonal matrix formed from the i-th column of matrix $\boldsymbol{\Lambda}$. Although the unknown $\boldsymbol{z}_j$ from each task are assumed to be independent via the above distributions, they are nonetheless linked via the common set of hyperparameters that will subsequently be learning from the data in a multi-task fashion. Additionally, from (12) we are expressing the j-th precision matrix as a linear combination of N diagonal precision matrix basis functions. This precision-based representation is chosen for algorithmic reasons.

Given this likelihood and prior, the posterior distribution $p(\boldsymbol{z}_j|\boldsymbol{x}_j; \boldsymbol{\Lambda}, \boldsymbol{W})$ is also a Gaussian with mean

$$\hat{\boldsymbol{z}}_j = \boldsymbol{\Gamma}_j \bar{X}_j^\top \left(\nu I + \bar{\boldsymbol{X}}_j \boldsymbol{\Gamma}_j \bar{\boldsymbol{X}}_j^\top\right)^{-1} \boldsymbol{x}_j. \quad (13)$$

Thus if $\boldsymbol{\Lambda}$ and $\boldsymbol{W}$ were known, we have access to a simple closed-form estimator for $\boldsymbol{z}_j$. The most challenging responsibility then becomes estimating these unknown hyperparameters. The empirical Bayesian solution to this problem is to first apply hyperpriors to $\boldsymbol{\Lambda}$ and $\boldsymbol{W}$, integrate out the unknown $\boldsymbol{Z}$, and then compute MAP estimates via

$$\max_{\boldsymbol{\Lambda}>0, \boldsymbol{W}\in\Omega} \int p(\boldsymbol{X}|\boldsymbol{Z}) p(\boldsymbol{Z}|\boldsymbol{\Lambda}, \boldsymbol{W}) p(\boldsymbol{\Lambda}) p(\boldsymbol{W}) d\boldsymbol{Z}. \quad (14)$$

For the covariance bases we simply assume a flat hyperior $p(\boldsymbol{\Lambda}) = 1$; for $\boldsymbol{W}$ we assume $p(\boldsymbol{W}) \propto \exp[-1/2\rho(\boldsymbol{W})]$, where ρ is a function (finite everywhere within the feasible set) designed to promote a clustering effect as will be described later. Given the above, applying a $-2\log$ transformation to (14) produces the equivalent problem

$$\min_{\boldsymbol{\Lambda}>0, \boldsymbol{W}\in\Omega} \sum_j \left[\boldsymbol{x}_j \Sigma_{x_j}^{-1} \boldsymbol{x}_j + \log\left|\Sigma_{x_j}\right|\right] + \rho(\boldsymbol{W}), \quad (15)$$

where

$$\Sigma_{x_j} \triangleq \nu I + \bar{\boldsymbol{X}}_j \boldsymbol{\Gamma}_j \bar{\boldsymbol{X}}_j^\top.$$

To facilitate later optimization, a convenient approximation to (15) can be formed using convex analysis and Jensen's inequality (Wang et al., 2015), leading to the multi-task objective function

$$\mathcal{L}(\boldsymbol{\Lambda}, \boldsymbol{W}) \triangleq \sum_j \left[\boldsymbol{x}_j \Sigma_{x_j}^{-1} \boldsymbol{x}_j\right] + \rho(\boldsymbol{W}) \quad (16)$$

$$+ \sum_j \log\left|\sum_i w_{ij} \boldsymbol{\Lambda}_i^{-1} + \frac{1}{\nu} \bar{\boldsymbol{X}}_j^\top \bar{\boldsymbol{X}}_j\right| + \sum_{ij} w_{ij} \log|\boldsymbol{\Lambda}_i|$$

that we will henceforth seek to optimize. This can be accomplished using standard variational bounding techniques

span a given D_k-dimensional subspace, but we require one additional coefficient such that we can honor the $\text{diag}[\boldsymbol{Z}_k] = 0$ constraint. All points will then necessarily be connected because each must be linked to D_k of these $D_k + 1$ points.

from (Wipf et al., 2011), with update rules contained in the supplementary file. Close inspection reveals a strong relationship with the updates from (Qi et al., 2008), the primary difference being how the parameter $\boldsymbol{W}$ is iterated. A significant advantage here is that (16), unlike the DP model, represents a closed-form cost function devoid of integrals and amenable to analysis in the domain of subspace segmentation. We henceforth refer to this paradigm as multi-task subspace clustering (MTSC).

4. Analysis

We now describe some analytical properties of MTSC that serve as justification for its application to subspace segmentation. Consistent with (2), we consider the limit as $\nu \rightarrow 0$. First we address the idealized scenario where the set $\{\mathcal{S}_k\}_{k=1}^K$ contains independent subspaces. In this special case, existing convex methods (e.g., SSC, LSR, LRR) have been previously shown to provide the ideal block-diagonal affinity matrix when globally optimized. However, MTSC involves a comparably complex non-convex objective. Fortunately though, it effectively still satisfies an identical criterion at any stationary point of $\mathcal{L}(\boldsymbol{\Lambda}, \boldsymbol{W})$.

Definition 1 (Ideal Block-Sparse Solution)**.** *Without loss of generality, assume that $\boldsymbol{x}_j \in \mathcal{S}_k$ for some k. We say that $\boldsymbol{z}_j$ is an* ideal block sparse solution *if, (i) $\boldsymbol{x}_j = \bar{\boldsymbol{X}}_j \boldsymbol{z}_j$ (feasibility), and (ii) the support of $\boldsymbol{z}_j$ is restricted to some subset of indices corresponding with columns of $\boldsymbol{X}_k$, i.e., $\boldsymbol{z}_j$ has no nonzero values multiplying columns of $\boldsymbol{X}$ in subspaces outside of k.*

Lemma 1. *Let columns of $\boldsymbol{X}$ be drawn from a union of K independent subspaces. Moreover, assume that the number of samples from each subspace N_k is sufficiently large and positioned such that for all $\boldsymbol{x}_j$ there exists an ideal block sparse solution. Then in the limit $\nu \rightarrow 0$, any stationary point $\{\boldsymbol{\Lambda}^*, \boldsymbol{W}^*\}$ of $\mathcal{L}(\boldsymbol{\Lambda}, \boldsymbol{W})$ will produce an estimate $\boldsymbol{z}_j^*$ via (13) that is also an ideal block sparse feasible solution for all $j = 1, \ldots, N$.*

Proof of the above is confined to the supplementary file. While the assumption of independent subspaces is somewhat restrictive, Lemma 1 nonetheless reassures us that at least MTSC is equally robust relative to convex algorithms in this regard, even though the former presumably possesses multiple local solutions. But to elucidate the real advantage of MTSC, we must consider a much more challenging and general model underlying our data set $\boldsymbol{X}$.

Definition 2 (Generic Subspace Model)**.** *Let $\mathcal{S}_k = span[\boldsymbol{B}_{\mathcal{S}_k} + \alpha \boldsymbol{R}_{\mathcal{S}_k}]$, where $\boldsymbol{B}_{\mathcal{S}_k}$ is any $D \times D_k$ matrix with $D > D_k$, $\boldsymbol{R}_{\mathcal{S}_k}$ is any random matrix with iid, continuously distributed elements, and $\alpha > 0$ is arbitrarily small. Additionally, assume that $\boldsymbol{X}_k = \boldsymbol{B}_k + \alpha \boldsymbol{R}_k$, where $\boldsymbol{B}_k$ is any $D \times N_k$ matrix with columns in $\mathcal{S}_k$, and $\boldsymbol{R}_k$ has iid columns, each of which is drawn from any continuous distribution in $\mathcal{S}_k$ such that $\boldsymbol{R}_k$ is full rank with probability one. We say that any data $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K]\boldsymbol{P}$ produced by this process follows the* generic subspace model*.*

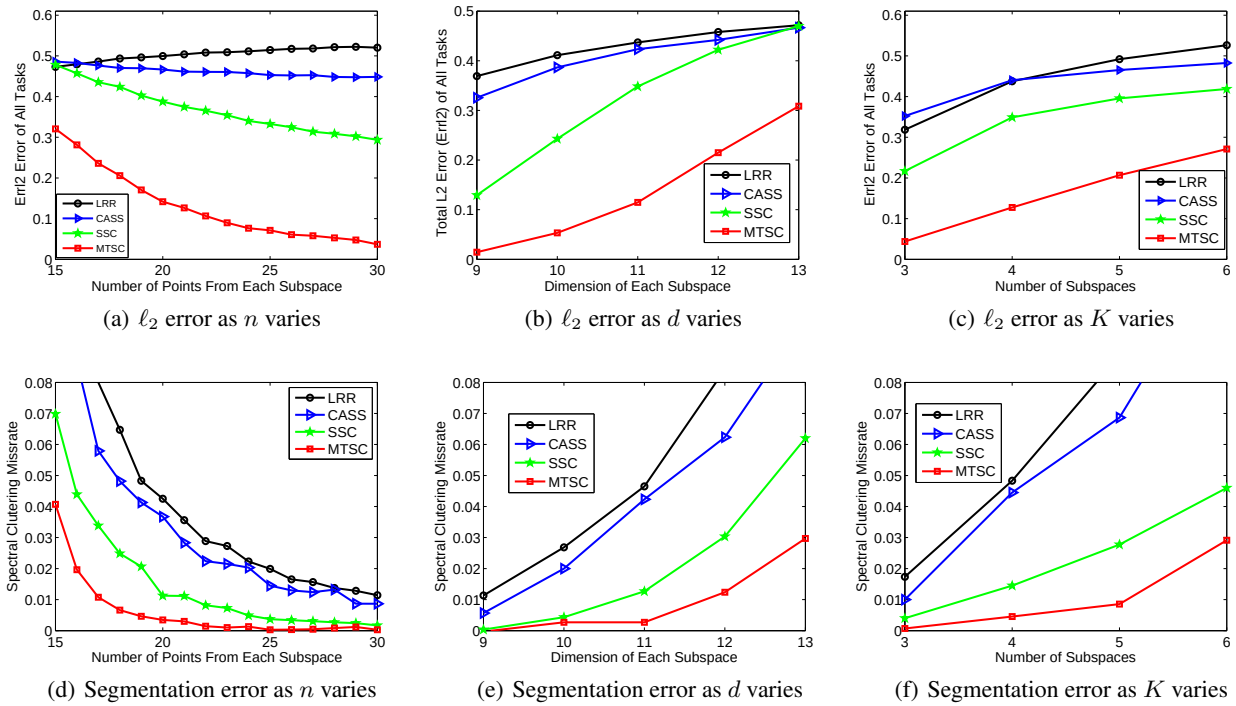
While admittedly cumbersome to present, this model essentially encapsulates all practical situations and data $\boldsymbol{X}$ of interest, and the small random components are only included as a technical necessity to avoid inconsequential adversarial co-linearities.

Lemma 2. *Suppose that $\boldsymbol{X}$ follows the generic subspace model and that $N_k > D_k$ for all k. Moreover, assume that $\rho(\boldsymbol{W}) = \beta \|\boldsymbol{W}\|_{\mathcal{F}}^2$, with $\beta > 0$. Then in the limit $\nu \rightarrow 0$, there exists a β sufficiently large such that any global optimum $\{\boldsymbol{\Lambda}^*, \boldsymbol{W}^*\}$ of $\mathcal{L}(\boldsymbol{\Lambda}, \boldsymbol{W})$ will be such that the corresponding $\boldsymbol{z}_j^*$ (computed via (13)) is an ideal block sparse solution for all j with probability one. Additionally, all points $\boldsymbol{x}_j$ which belong to the same subspace will be connected in the corresponding affinity matrix computed via (3).*

This result is interesting in that, provided β is large enough, MTSC naturally achieves what (8) ideally promotes, a globally minimizing solution which guarantees the correct final segmentation of the data. Moreover, unlike (8) MTSC manages this even without knowledge of K. The caveat of course is that we must avoid getting stuck in a suboptimal local minima. But there are certain indications that minimization of $\mathcal{L}(\boldsymbol{\Lambda}, \boldsymbol{W})$ is particularly well-suited for avoiding such adversarial extrema.

One such line of reasoning proceeds by considering the special case where $\rho(\boldsymbol{W}) = 0$. Here it can be shown that the MTSC model collapses to the SBL objective from (Tipping, 2001) with decoupled tasks. In the context of solving (7), SBL has been shown to be equivalent to adapting a particular dictionary-dependent penalty function for $g(\boldsymbol{z}_j)$ that compensates for dictionary coherence and yet unlike CASS, is still invariant to transformations of the data via $\boldsymbol{QX}$ (Wipf, 2011). In our own experiments we found that SBL was adept at finding considerably sparser solutions than the ℓ_1 norm used by SSC; however, in some sense these solutions were actually *too* sparse and occasionally led to relatively poor final spectral clustering results even though the block-sparsity profile was nearly perfect (note that SBL provably does not satisfy Lemma 2). In this respect, MTSC can be interpreted as coupling desirable local sparsity properties of SBL in an integrated framework with an eye towards the final clustering fidelity.

Affine Subspace Segmentation: The generalization to clustering affine subspaces also motivates the utility of SBL-related estimators over standard convex alternatives. Let $\boldsymbol{T} \triangleq [\boldsymbol{T}_1, \ldots, \boldsymbol{T}_K]\boldsymbol{P} \in \mathbb{R}^{D \times N}$, where $\boldsymbol{T}_k = \boldsymbol{t}_k \boldsymbol{1}_{N_k}^\top$, $\boldsymbol{t}_k \in \mathbb{R}^D$ is arbitrary, and $\boldsymbol{1}_{N_k}$ denotes a length N_k vector of ones. Here each $\boldsymbol{t}_k$ can be interpreted as a translation



(a) ℓ_2 error as n varies (b) ℓ_2 error as d varies (c) ℓ_2 error as K varies

(d) Segmentation error as n varies (e) Segmentation error as d varies (f) Segmentation error as K varies

Figure 1. ℓ_2 and segmentation error comparisons for different algorithms.

vector which will be applied to every element of the k-th subspace, with $\text{rank}[T] \leq K$. Additionally, let $\boldsymbol{\Phi}$ denote a non-negative, $N \times N$ diagonal matrix. Now consider any collection of N points $\boldsymbol{X}$ that follow the linear subspace model (1) with unit norm columns (i.e., $\|\boldsymbol{x}_j\|_2 = 1$), and then form the modified data

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\Phi} + \boldsymbol{T}. \tag{17}$$

The new data $\boldsymbol{Y}$ represent an arrangement of points in K *affine* subspaces, with points within each subspace having arbitrary scalings. The standard SSC adaptation to this case requires solving

$$\min_{\boldsymbol{z}_j} \|\boldsymbol{z}_j\|_1 \;\text{ s.t. } \boldsymbol{y}_j = \boldsymbol{Y}\boldsymbol{z}_j,\; z_{ij} = 0,\; \sum_i z_{ij} = 1, \tag{18}$$

for all j, where the additional constraint leads to a form of translation invariance accounting for the affine subspace (Elhamifar & Vidal, 2013).[5] However, it is well-known that if the columns of the associated dictionary (in this case columns of $\boldsymbol{Y}$) are not normalized, then the minimal ℓ_1 norm solution will be heavily biased. Simply put, the resulting nonzero coefficients will tend to align with the large dictionary columns even at the expense of finding maximally sparse solutions confined to the desired subspace. For present purposes this implies that coefficients from the wrong subspace may be selected at the expense of clustering accuracy.

One potential solution would be to explicitly normalize the columns of $\boldsymbol{Y}$. But this comes with a substantial cost because the normalization may be dominated by the low-rank translation component $\boldsymbol{T}$ rather than the magnitude scaling matrix $\boldsymbol{M}$. For example, if some translation $\boldsymbol{t}_k$ is large, the corresponding norms of all data points $\boldsymbol{y}_j$ in this subspace will be rescaled with roughly the same normalization factor even if these points have wildly different distances from $\boldsymbol{t}_k$. Fortunately SBL is naturally robust against such transformations (Wipf, 2011), and MTSC inherits many related attributes.

5. Experiments

In this section we present empirical results designed to highlight the utility of MTSC. For this purpose we compare with a suite of recent competing algorithms implementing SSC (Elhamifar & Vidal, 2013), LRR (Liu et al., 2013), LSR (Lu et al., 2012), and CASS (Lu et al., 2013), in each case using the authors' original code.

Simulated Data: We first describe simulation experiments that allow us to bypass the effects of noise and outliers to focus on intrinsic differences in the baseline algorithms. In this restricted setting, LSR and LRR are equivalent for rea-

[5] MTSC can be trivially modified to account for this additional constraint as well.

sons stated previously. We generate K disjoint subspaces, each with $D_k = d$ dimensions embedded with uniformaly random angles in $\mathbb{R}^D$. Within each subspace, we draw $N_k = n$ points from an iid Gaussian distribution, which are then projected to the ambient space. Results are combined to form $\boldsymbol{X}$, and each algorithm is applied to learn an affinity matrix $\boldsymbol{A}$ followed by the standard spectral clustering step to obtain the final segmentation.

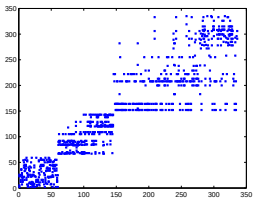
Performance is evaluated via two metrics. First, to isolate each algorithm's ability to obtain the correct block-wise structure, we evaluate the estimated $\hat{\boldsymbol{Z}}$ provided by each algorithm before spectral clustering via

$$\ell_2 \text{ error} \triangleq 1/N \sum_{j=1}^{N} \left(1 - \|\hat{\boldsymbol{z}}'_j\|_2 / \|\hat{\boldsymbol{z}}_j\|_2\right), \qquad (19)$$

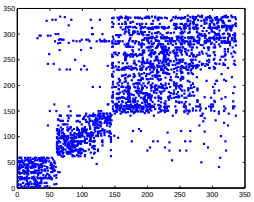
where $\hat{\boldsymbol{z}}'_j$ denotes the elements of $\hat{\boldsymbol{z}}_j$ associated with points in the same cluster as $\boldsymbol{x}_j$. This metric provides an estimate of the proportion of signal energy that is *not* along the correct block diagonal of $\hat{\boldsymbol{Z}}$. Secondly, we compute the final *segmentation error* (SE) after spectral clustering, which represents the proportion of points that have been assigned to the wrong subspace. For all simulations, $D = 26$. In Figure 1, the *left* column displays results as n is varied with $d = 13$ and $K = 4$ fixed. The *middle* column then shows results as d is varied while $n = 16$ and $K = 4$ are fixed. Finally, the *right* column varies K while $d = 11$ and $n = 16$ are fixed. In all cases, results were averaged over 100 independent trials for each curve, and MTSC shows a significant advantage.

Motion Segmentation Data: We next present evaluations using the Hopkins 155 Motion Database (Elhamifar & Vidal, 2013). We chose this data because it is a standard benchmark and mostly free of outliers, and therefore avoids the influence of outlier removal layers which must be specially tuned for each algorithm. The Hopkins data consists of 155 video sequences. Every sequence produces a data matrix $\boldsymbol{X}$, each column of which represents a two-dimensional feature point tracked throughout the video. These feature points correspond with objects moving in the scene which we would like to segment. Of the 155 videos, 120 contain two motions while the remaining contain three. We evaluate the performance of the chosen algorithms, using the noise models provided in the original code and tuning parameters adjusted from default settings.

Table 1 shows the results using the ℓ_2 error metric averaged across sequences, while Tables 2 and 3 display the corresponding segmentation errors, where again MTSC provides a distinct advantage. Note that existing methods all include some form of post-processing step, either directly in published work or embedded in the code itself. This is typically designed to remove artifacts from the affinity matrix and likely explains some of the wide variance in reported results coming from ostensibly equivalent algorithms. Because our purpose here is to explore subspace segmentation algorithms, not solve a particular application per se with domain-specific augmentations, in all cases we disabled postprocessing to place each method on an equal footing and avoid reliance on unknown quantities such as d (e.g., this value is frequently used for thresholding $\boldsymbol{A}$ as a postprocessing step).



(a) MTSC



(b) SSC

Figure 2. $\hat{\boldsymbol{Z}}$ estimated from motion segmentation data using MTSC and SSC.

Table 1. ℓ_2 error ($\times$ 100) for 2 and 3 motion data

Algo.	SSC	CASS	LRR	LSR	MTSC
2 Motion	1.60	18.06	24.88	23.27	**1.38**
3 Motion	3.04	26.23	31.8	30.07	**1.60**

Table 2. Segmentation error ($\times$ 100) for 2 motion data

Algo.	SSC	CASS	LRR	LSR	MTSC
Mean	1.92	3.30	3.63	3.14	**1.60**
Med.	**0**	**0**	0.21	0.2	**0**
Stdv.	7.1	7.7	8.77	8.06	**5.48**

Table 3. Segmentation error ($\times$ 100) for 3 motion data

Algo.	SSC	CASS	LRR	LSR	MTSC
Mean	7.15	9.22	7.56	6.53	**3.80**
Med.	0.69	3.58	3.99	2.56	**0.67**
Stdv.	13.67	11.27	11.92	**8.54**	8.66

Finally, Figure 2 illustrates the ability of MTSC to both find a representation devoid of energy outside of the desired block structure, and yet with high connectivity within each block by virtue of the observed row-sparsity structure. Note that ordinary SBL often displays a similar degree of overall sparsity as MTSC (not shown), but lacks the requisite intra-block connections that lead to an accurate final segmentation.

6. Conclusions

Our work is the first to connect MTL to the important problem of subspace segmentation. This observation then leads to a principled retrofitting of an existing MTL pipeline that is both theoretically accessible and empirically useful. In particular, our approach compensates for intrinsic limitations of traditional convex penalty functions, navigating highly coherent data sets $\boldsymbol{X}$ towards robust affinity-matrix formation, even without postprocessing steps.

7. Acknowledgement

Y. Wang is sponsored by the University of Cambridge Overseas Trust. Y. Wang and Q. Ling are partially supported by sponsorship from Microsoft Research Asia. Q. Ling is also supported in part by NSFC grant 61004137. W. Chen is supported by EPSRC Research Grant EP/K033700/1 and the Natural Science Foundation of China 61401018.

References

Babacan, S. D., Nakajima, S., and Do, M. N. Probabilistic low-rank subspace clustering. In *NIPS*, 2012.

Elhamifar, E. and Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11): 2765–2781, 2013.

Feng, J., Lin, Z., Xu, H., and Yan, H. Robust subspace segmentation with block-diagonal prior. In *CVPR*, 2014.

Grave, E., Obozinski, G., and Bach, F. Trace lasso: A trace norm regularization for correlated designs. In *NIPS*, 2011.

Hong, W., Wright, J., Huang, K., and Ma, Y. Multiscale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing*, 15(12):3655–3671, 2006.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., and Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

Lu, C., Min, H., Zhao, Z., Zhu, L., Huang, D., and Yan, S. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, 2012.

Lu, C., Feng, J., Lin, Z., and Yan, S. Correlation adaptive subspace segmentation by trace lasso. In *ICCV*, 2013.

Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Mahdi, S. and Candes, E. J. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

Qi, Y., Liu, D., Dunson, D., and Carin, L. Multi-task compressive sensing with Dirichlet process priors. In *ICML*, 2008.

Rao, S., Tron, R., Vidal, R., and Ma, Y. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

Tipping, M. E. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

Wang, Y., Wipf, D., Yun, J. M., Chen, W., and Wassell, I. J. Clustered sparse Bayesian learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

Wipf, D. P. Sparse estimation with structured dictionaries. In *NIPS*. 2011.

Wipf, D. P., Rao, B. D., and Nagarajan, S. Latent variable Bayesian models for promoting sparsity. *IEEE Transactions on Information Theory*, 57(9):6236–6255, 2011.