LEARNING SILHOUETTES WITH GROUP SPARSE AUTOENCODERS

Emmanouil Theodosis and Demba Ba

School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138

ABSTRACT

Sparse coding has been extensively used in neuroscience to model brain-like computation by drawing analogues between neurons' firing activity and the nonzero elements of sparse vectors. Contemporary deep learning architectures have been used to model neural activity, inspired by signal processing algorithms; however sparse coding architectures are not able to explain the higher-order categorization that has been empirically observed at the neural level. In this work, we propose a novel model-based architecture, termed group-sparse autoencoder, that produces sparse activity patterns in line with neural modeling, but showcases a higher-level order in its activation maps. We evaluate a dense model of our architecture on MNIST and CIFAR-10 and show that it learns dictionaries that resemble silhouettes of the given class, while its activations have a significantly higher level order compared to sparse architectures. Source code is available at: https://github.com/manosth/silhouette-learning.

***Index Terms*—** Unrolled optimization, group sparsity, neural categorical representation, silhouette learning

1. INTRODUCTION

Neuroscientists have observed that human's inferior temporal (IT) cortex activity patterns tend to exhibit a higher level order and there exists an inherently categorical representation of objects [1, 2]. In particular, objects that belong in the same conceptual class elicit similar neural responses; interestingly, the categorical representation seems to be hierarchical and there is a conceptual ordering of the classes [2]. The neural activities that generate the activity patterns are sparse and sparse coding has been extensively used in neuroscience [3] to model the firing activity of the neurons; however, the computationally recovered activity patterns do not exhibit the categorization that is empirically observed in human subjects.

Dictionary learning [4] has been extensively employed in neuroscience in order to learn optimal atoms to model brain activity. Computational advancements led to the use of deep learning models and the recently developed model-based learning approaches [5, 6] offer a theoretically motivated approach to deep learning, tackling one of its fundamental problems: current, high-performing architectures lack explainability. The model-based learning paradigm addresses this by invoking domain knowledge in order to constrain the neural architectures, thus making them amenable to interpretation. Within this context, *unrolled networks* [7], popularized by the seminal work of [8], unroll the steps of iterative optimization algorithms to form a neural network. LISTA [8], which enforces sparsity on the units of deep layers, is based on the unfolding of the Iterative Shrinkage Thresholding Algorithm (ISTA), a sparse coding optimization algorithm. Sparsity-focused generative models [9] are most frequently employed due to their experimentally and theoretically proven generalization power [10, 11]. In addition, because they can significantly reduce the number of nonzero coefficients—units active at a given layer—sparse models have also been used to speed up inference in deep neural networks [12, 13].

Recent research has deviated from the traditional sparse coding model; certain works reconsidered the sparsity-promoting minimization [14], where others focused on exploring different generative models [15, 16]. Within the latter class, works studying *group sparsity* [17, 18] have been rather prolific. In addition to minimizing the number of non-zero coefficients, group sparsity forces them to occur in blocks. Inputs that share similar activity patterns can be interpreted as belonging to the same class or cluster. The groupings manifest themselves either as a direct arrangement of the hidden units of neural networks into blocks [19, 20], or as a clustering of data that, a priori, share similar characteristics, such as patches of natural images [21]. Enforcing group structure has proved practical in applications and outperforms approaches based on the traditional notion of sparsity.

We propose a novel unrolled architecture based on group sparsity in order to encode stimuli to neural representations that maintain the performance of sparse coding models but exhibit a higher-level categorization of neural activity that is consistent with the empirical ordering that is observed in IT. In Section 2 we introduce the group-sparse generative model and group-sparse dictionary learning. Section 3 introduces the unrolled architecture of group-sparse autoencoders and experiments are presented in Section 4. Finally, we conclude in Section 5.

2. GROUP-SPARSITY AND DICTIONARY LEARNING

In model-based approaches the observed data $\{\boldsymbol{y}_i\}_{i=1}^N \in \mathcal{Y}$[1] are assumed to adhere to a generative model. Formally, we assume that the data satisfy

$$\boldsymbol{y}_i = f_{\theta^*}(\boldsymbol{x}_i^*), \tag{1}$$

where $\boldsymbol{x}_i^* \in \mathcal{X}$ is a latent vector and f_θ, parametrized by θ, comes from a function class $\mathcal{F}$ that describes the relation between the data $\boldsymbol{y}_i$ and the latent variables $\boldsymbol{x}_i^*$. Most frequent are models of *linear* relations, where the function f_θ is of the form $f_\theta \colon \boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$, parametrized by $\theta = \{\boldsymbol{A}\}$. Note that even when f is linear, the inverse problem of recovering $\boldsymbol{x}$ from observations is generally not linear.

2.1. Group-sparse generative model

Consider a generative model where each observation[2] $\boldsymbol{y}$ belongs to the union of one, or more, subspaces [22]. In this general group-sparse model the observed data satisfy

$$\boldsymbol{y} = \boldsymbol{A}^*\boldsymbol{x}^* = \sum_{g\in S} \boldsymbol{A}_g^*\boldsymbol{x}_g^*, \tag{2}$$

where $S \subset [\Gamma]$ denotes the *group support* (i.e. which of the Γ groups are active), and the latent vector has the form $\boldsymbol{x}^* = [\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_\Gamma^*]^T$. Gaussian mixture models, sparse models, and nonnegative sparse models [23] can readily be derived as special cases of the highly-expressive generative model from (2). The group-sparse prior assumes that the latent representation $\boldsymbol{x}$ is sparse, and that nonzero entries occur in blocks (groups). The model also implies a decomposition of $\boldsymbol{A}^*$ into sub-matrices $\boldsymbol{A}_1^*, \boldsymbol{A}_2^*, \ldots, \boldsymbol{A}_\Gamma^*$ such that $\boldsymbol{A}^* = [\boldsymbol{A}_1^*\boldsymbol{A}_2^* \ldots \boldsymbol{A}_\Gamma^*]$, where we assume that each group $\boldsymbol{A}_g^*$ has exactly d elements. Without additional structure, the generative model may not yield a unique solution; for example, [18] impose orthonormality on $\boldsymbol{A}_g^*$ to ensure uniqueness.

An analogue to the *coherence* of a dictionary in sparse models (defined as $\mu = \max_{i\neq j}|\boldsymbol{a}_i^{*T}\boldsymbol{a}_j^*|$; the inner-product with the largest magnitude in $\boldsymbol{A}^*$) is the *block coherence* of $\boldsymbol{A}^*$

$$\mu_B = \max_{g\neq h} \frac{1}{d}\|\boldsymbol{A}_g^{*T}\boldsymbol{A}_h^*\|_2. \tag{3}$$

Intuitively, coherence metrics give a sense of how correlated the different columns, or groups, of $\boldsymbol{A}^*$ are and directly affect the ability to recover latent vectors. Assuming normalized groups, as we will in this work, it holds that $0 \leq \mu_B \leq \mu \leq 1$.

[1] We intentionally do not write $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, as the setting we are considering is *strictly* unsupervised.

[2] For the rest of the text, we drop the index i to reduce clutter.

2.2. Group-sparse dictionary learning

Assuming a linear underlying generative model, dictionary learning sets out to learn a dictionary $\boldsymbol{A}$ such that every vector $\boldsymbol{y}$ in a data set adopts a sparse representation as a linear combination of the columns of $\boldsymbol{A}$ using a vector $\boldsymbol{x}$. In group-sparse settings, given the dictionary $\boldsymbol{A}$, *group-sparse coding* lets us find $\boldsymbol{x}$ as the solution to the optimization problem

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad \|\boldsymbol{x}\|_{\ell_0/\ell_2}, \qquad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{4}$$

The ℓ_0/ℓ_2 norm, expressed as the ℓ_0 "norm" of the vector of ℓ_2 norms $[\|\boldsymbol{x}_1\|_2, \|\boldsymbol{x}_2\|_2, \ldots, \|\boldsymbol{x}_\Gamma\|_2]^T$, minimizes the number of active groups. The combinatorial nature of ℓ_0 "norm" makes this optimization intractable in practice. A popular approach utilizes the ℓ_1 norm instead, as a tractable convex relaxation of the optimization of (4), yielding

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad \|\boldsymbol{x}\|_{\ell_1/\ell_2}, \qquad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{5}$$

where $\|\boldsymbol{x}\|_{\ell_1/\ell_2} = \sum_{g\in S}\|\boldsymbol{x}_g\|_2$. Both the optimizations of (4) and (5) require the recovery of latent codes $\boldsymbol{x}$ that lead to an exact reconstruction of the data $\boldsymbol{y}$. The following *unconstrained* optimization problem enables a trade-off between exact recovery and the group-sparsity of the latent codes

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\sum_{g\in S}\|\boldsymbol{x}_g\|_2. \tag{6}$$

Optimization objectives of the form $\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\Omega(\boldsymbol{x})$ have been studied extensively in the literature and can be directly solved via the theory of proximal operators. Pertinent to the current discussion, the proximal operator promoting group-sparse structures can be derived as

$$\sigma_\lambda(\boldsymbol{x}_g) = \left(1 - \frac{\lambda}{\|\boldsymbol{x}_g\|_2}\right)_+ \boldsymbol{x}_g, \tag{7}$$

where $(\cdot)_+ = \max(\cdot, 0)$. Note that this proximal operator bears a striking similarity to $\mathrm{ReLU}(x) = \max(x, 0)$. Indeed, we can consider (7) as a generalization of ReLU (informally termed "Group ReLU"), where the thresholding is applied in a structured way, instead of an element-wise fashion. Dictionary learning can then be performed by solving

$$(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{x}}) = \operatorname*{arg\,min}_{\boldsymbol{A}\in\mathcal{A}, \boldsymbol{x}\in\mathcal{X}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\sum_{g\in S}\|\boldsymbol{x}_g\|_2, \tag{8}$$

a nonconvex optimization problem. A popular approach, termed *alternating minimization* [24], cycles between group-sparse coding and dictionary update steps. Formally, the group-sparse coding step considers the dictionary $\widehat{\boldsymbol{A}}^{(t)}$ fixed and solves

$$\widehat{\boldsymbol{x}}^{(t+1)} = \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathcal{X}} \quad \frac{1}{2}\|\boldsymbol{y} - \widehat{\boldsymbol{A}}^{(t)}\boldsymbol{x}\|_2^2 + \lambda\sum_{g\in S}\|\boldsymbol{x}_g\|_2, \tag{9}$$

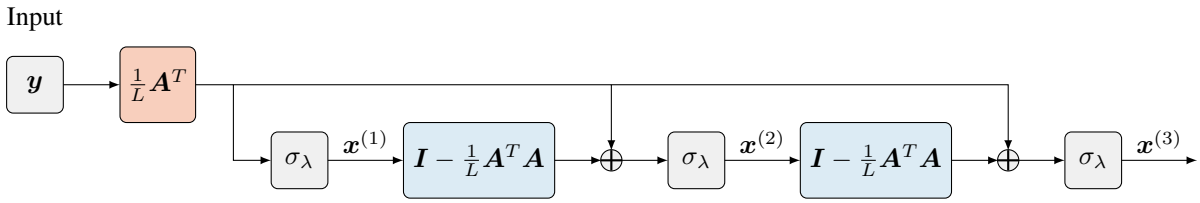


Fig. 1: The unfolded architecture of (11) for 3 layers.

followed by an optimization to find the optimal dictionary $\widehat{\boldsymbol{A}}^{(t+1)}$ given an estimate of the latent code $\widehat{\boldsymbol{x}}^{(t+1)}$

$$\widehat{\boldsymbol{A}}^{(t+1)} = \underset{\boldsymbol{A}\in\mathcal{A}}{\arg\min} \quad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}^{(t+1)}\|_2^2, \tag{10}$$

where (9) and (10) are performed in an alternating manner until convergence, yielding $(\widehat{\boldsymbol{A}}^T, \widehat{\boldsymbol{x}}^T)$.

3. ARCHITECTURE

Unrolling the optimization of (8) results in the representation at layer $l+1$ to be given by

$$\boldsymbol{x}^{(t+1)} = \sigma_\lambda\left(\boldsymbol{x}^{(t)} + \frac{1}{L}\boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(t)})\right), \tag{11}$$

where (t) was dropped from $\boldsymbol{A}$ as we consider a tied architecture. Similar to the optimization algorithm, given enough iterations (in our case, layers), the latent representation $\boldsymbol{x}^{(t)}$ will be a good approximation of the group-sparse codes. The dictionary learning of the group dictionaries happens implicitly as we train the architecture via backpropagation.

4. EXPERIMENTS

In this section we experimentally evaluate the proposed architecture. We design experiments and evaluate the performance of the group-sparse autoencoder versus sparse autoencoders and showcase that the former is significantly more aligned with empirical findings in neuroscience, exhibiting a categorical organization in its latent space. In all our experiments, the sparse autoencoder is trained using $\lambda = 0.5$ and the group-sparse one using $\lambda = 7.5$.

4.1. Pairwise activations

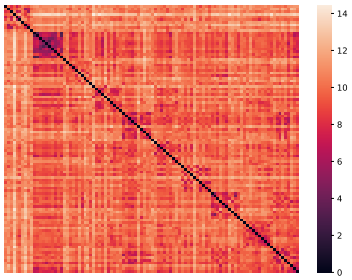
Figure 2 shows pairwise distances between 100 MNIST test images of each of the 10 digit classes with respect to the pixel basis, the learned dictionary of using a sparse autoencoder, and the learned dictionary using a group-sparse autoencoder. We observe that the similarity matrix of the raw data (pixel basis), as well as the latent representation of the sparse autoencoder, do not exhibit any particular structure. This is expected as a categorical organization is not enforced through the optimization procedure in the case of the latter and not expected in the case of the former. We report that the similarity structure of the group-sparse dictionary lends itself most readily to standard similarity-based clustering algorithms and strongly resembles the higher-level order and hierarchical categorical representation that is empirically found in neural data [2]. Indeed, we observe that the latent representations of samples of the same class are similar to one another and dissimilar to those of other classes.

4.2. Activity patterns

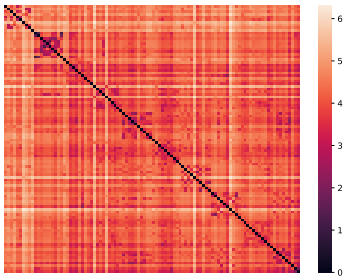
Figure 3 shows the class-specific mean activity maps of the latent representations using the different architectures for both MNIST and CIFAR10. The figure only compares one class per dataset due to space constraints, but consistent results are reported for every class. We observe that, as expected by the group-sparse prior, the group-sparse autoencoder's latent representation exhibits significant structure at the class level. This was already hinted by, and is consistent with, the results of Figure 2. Each learned sub-dictionary of $\boldsymbol{A}$ tends to be aligned with some of the classes and that is also reflected through their learned atoms. In stark contrast, the latent representations of sparse autoencoders seem uniform and this implies that classes do not have preferences for specific neurons. Note that the non-sparse means we observe for the sparse autoencoder are expected as different samples have no optimizational incentive to use the same atoms in their reconstruction. The amount of sparsity for both architectures can be tuned via λ in order to get sparser activations and was chosen so that both models would have similar classification performance.

4.3. Dictionary atoms

Finally, we present dictionaries learned with sparse and group-sparse autoencoders in Figure 4 to highlight their differences. For both datasets, we observe that sparse autoencoders learn local features that resemble strokes or edges. This validates the findings of the previous subsections, as these atoms have a universal flavor and are not particularly aligned with a specific class. In contrast, the atoms learned using the group-sparse architecture are distinctly aligned with a specific class and are *silhouettes*, or averages, of the samples of that class.



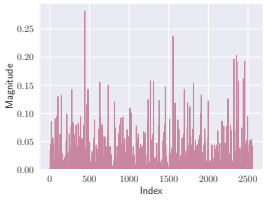
(a) Similarity matrix of MNIST images when representing the data using their raw format (pixel basis).



(b) Similarity matrix of MNIST images on the latent representation learned by a sparse autoencoder.

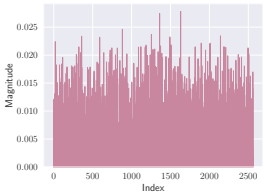
(c) Similarity matrix of MNIST images on the latent representation learned by a group-sparse autoencoder

Fig. 2: Pairwise distances between the representations of MNIST test images for different latent representations.

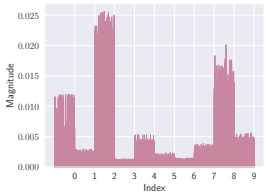


(a) MNIST, digit 7, sparse autoencoder.

(b) MNIST, digit 7, group-sparse autoencoder.

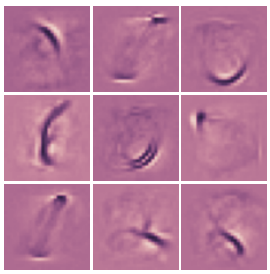


(c) CIFAR-10, class "airplanes", sparse autoencoder.

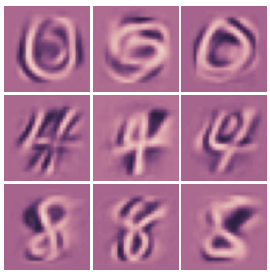


(d) CIFAR-10, class "airplanes", group-sparse autoencoder.

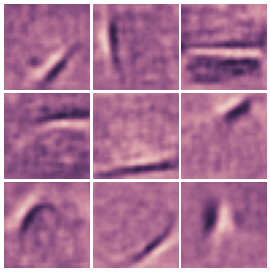
Fig. 3: Mean activity patterns of test samples for MNIST (top row) and CIFAR10 (bottom row) using sparse versus group-sparse autoencoders. One class per dataset shown due to space constraints.



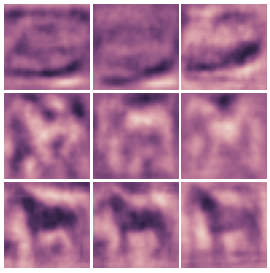
(a) MNIST, atoms, sparse autoencoder.



(b) MNIST, digit 0, 4, and 8, group-sparse autoencoder.



(c) CIFAR10, atoms, sparse autoencoder.



(d) CIFAR10, classes "car", "dog", and "horse", group-sparse autoencoder.

Fig. 4: Dictionary atoms learns on MNIST (top row) and CIFAR10 (bottom row) using sparse versus group-sparse autoencoders. For the group-sparse autoencoders each row corresponds to a different class.

5. CONCLUSIONS

In this work we proposed a novel unrolled architecture that produces activity maps that exhibit higher order. Motivated by the misalignment of sparse coding with empirically observed phenomena in neural data, we considered a group-sparse prior. We developed an unrolled autoencoder for group-sparse coding, showcased that it generates latent representations that exhibit categorical organization, leads to similarity maps that resemble those empirically observed in neuroscience, and learns dictionary atoms that are characteristic silhouettes of classes.

6. REFERENCES

[1] Roozbeh Kiani, Hossein Esteky, Koorosh Mirpour, and Keiji Tanaka, "Object category structure in response patterns of neuronal population in monkey inferior temporal cortex," *Journal of Neurophysiology*, vol. 97, 2007.

[2] Marieke Mur, Mirjam Meys, Jerzy Bodurka, Rainer Goebel, Peter Bandettini, and Nikolaus Kriegeskorte,

"Human object-similarity judgments reflect and transcend the primate-it object representation," *Frontiers in Psychology*, vol. 4, no. 128, 2013.

[3] Bruno Olshausen and David Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, 2004.

[4] Bruno Olshausen and David Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1¿," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1994.

[5] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros Dimakis, "Compressed sensing using generative models," in *International Conference on Machine Learning*, 2017.

[6] Nir Shlezinger, Jay Whang, Yonina Eldar, and Alexandros Dimakis, "Model-based deep learning," in *arXiv*, 2020.

[7] John Hershey, Jonathan Le Roux, and Felix Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," in *arXiv*, 2014.

[8] Karol Gregor and Yann LeCun, "Learning fast approximations of sparse coding," in *International Conference on Machine Learning*, 2010.

[9] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, 1996.

[10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online dictionary learning for sparse coding," in *International Conference on Machine Learning*, 2009.

[11] Nishant Mehta and Alexander Gray, "Sparsity-based generalization bounds for predictive sparse coding," in *International Conference on Machine Learning*, 2013.

[12] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Penksy, "Sparse convolutional neural networks," in *Conference on Computer Vision and Pattern Recognition*, 2015.

[13] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel Emer, *Efficient processing of deep neural networks*, Morgan & Claypool Publishers, 2020.

[14] Shuai Huang and Trac Tran, "Sparse signal recovery via generalized entropy functions minimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1322–1337, 2018.

[15] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini, "Group sparse regularization for deep neural networks," *Neurocomputing*, vol. 241, pp. 81–89, 2017.

[16] Pol del Aguila Pla and Joakim Jaldén, "Cell detection by functional inverse diffusion and non-negative group sparsity–part ii: Proximal optimization and performance evaluation," *IEEE Transactions on Signal Processing*, vol. 20, no. 20, pp. 5422–5437, 2018.

[17] Ming Yuan and Yi Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society*, vol. 68, no. 1, pp. 49–67, 2006.

[18] Yonina Eldar, Patrick Kuppinger, and Helmut B´olcskei, "Block-sparse signals: uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.

[19] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems*, 2016.

[20] Jaehong Yoon and Sung Ju Hwang, "Combined group and exclusive sparsity for deep neural networks," in *International Conference on Machine Learning*, 2017.

[21] Bruno Lecouat, Jean Ponce, and Julien Mairal, "Fully trainable and interpretable non-local sparse models for image restoration," in *European Conference on Computer Vision*, 2020.

[22] Reémi Gribonval and Morten Nielsen, "Sparse representations in unions of bases," *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3320–3325, 2003.

[23] Thanh Nguyen, Raymond Wong, and Chinmay Hegde, "On the dynamics of gradient descent for autoencoders," in *International Conference on Artificial Intelligence and Statistics*, 2019.

[24] Paul Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal of Control and Optimization*, vol. 29, no. 1, pp. 119–138, 1991.