THE PICASSO ALGORITHM FOR BAYESIAN LOCALIZATION VIA PAIRED COMPARISONS IN A UNION OF SUBSPACES MODEL

Gregory Canal, Marissa Connor, Jihui Jin, Namrata Nadagouda, Matthew O'Shaughnessy, Christopher J. Rozell, and Mark A. Davenport

School of Electrical & Computer Engineering
Georgia Institute of Technology

ABSTRACT

We develop a framework for localizing an unknown point $\boldsymbol{w}$ using paired comparisons of the form "$\boldsymbol{w}$ is closer to point $\boldsymbol{x}_i$ than to $\boldsymbol{x}_j$" when the points lie in a union of known subspaces. This model, which extends a broad class of existing methods to exploit union of subspaces structure, provides a powerful framework for using the types of structure found in many practical applications. We divide the problem into two phases: (1) determining which subspace $\boldsymbol{w}$ lies in, and (2) localizing $\boldsymbol{w}$ within the identified subspace using existing techniques. We introduce two algorithms for determining the subspace in which an unknown point lies: the first admits a sample complexity guarantee demonstrating the advantage of the union of subspaces model, and the second improves performance in practice using an adaptive Bayesian strategy. We demonstrate the efficacy of our method with experiments on synthetic data and in an image search application.

Index Terms— Paired comparisons, Bayesian, stochastic, perceptual compression, PICASSO

1. INTRODUCTION

In many applications we are interested in learning a preference function over a set of points $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^d$ using *paired comparisons* of the form "$\boldsymbol{w}$ is closer to point $\boldsymbol{x}_i$ than to $\boldsymbol{x}_j$." Algorithms for learning preference functions or rank orderings from this type of observation have been explored in a wide variety of settings [1–8]. However, the majority of this literature imposes no structural assumptions on the points $\{\boldsymbol{x}_i\}$. As a result, the number of paired comparisons required typically scales *at least* linearly with the size n of the set. Unfortunately, obtaining this volume of training data may often be infeasible, especially when the paired comparisons are supplied by human subjects.

To address this challenge, a powerful approach is to exploit latent structure in the $\{\boldsymbol{x}_i\}$ to reduce the required number of comparisons. As an example, suppose that the $\{\boldsymbol{x}_i\}$ can be embedded in $\mathbb{R}^k$ where $k \ll d$ and that our comparisons are of the form $\|\boldsymbol{w} - \boldsymbol{x}_i\| \gtrless \|\boldsymbol{w} - \boldsymbol{x}_j\|$, where $\boldsymbol{w} \in \mathbb{R}^k$ is the target point of a search. In this case our problem reduces

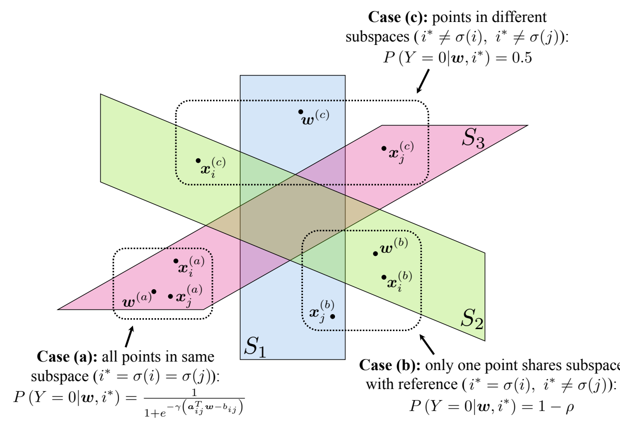


Fig. 1. The measurement probability model (1) differs based on the subspace membership of $\boldsymbol{w}$, $\boldsymbol{x}_i$, and $\boldsymbol{x}_j$.

to that of estimating $\boldsymbol{w}$ and the query complexity can be reduced from depending on the number of items n to depending instead only on the dimension k [4, 9, 10].

This model, however, is inappropriate for the many practical applications where points lie not in all of $\mathbb{R}^d$, but rather in a *union of subspaces* $\{S_i\}_{i=1}^K \subset \mathbb{R}^d$ [11]. The union of subspaces model gives us the flexibility to model settings where distances are well-defined only between subsets of points [12]. For example, in Section 4 we consider an application of our method to localizing a point in the space of images of various classes (e.g., cars, horses, and dogs) where each class will correspond to a subspace in the union. Here, we expect that the distances (in feature space) between images of cars are meaningful, and that the distance between two cars will be much less than the distance between a car and horse. However, in general comparing the distance between a car and horse and the distance between a horse and dog will not be meaningful.

The fundamental fact exploited by our algorithm is that comparisons involving points within the *same* subspace and comparisons between points in *different* subspaces will be governed by different probability models: when compared

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 10,2020 at 15:25:27 UTC from IEEE Xplore. Restrictions apply.

points are in the same subspace, the comparison is meaningful and standard algorithms for localization apply, but comparisons between different subspaces are not meaningful and return random results. We use this asymmetry to *infer the true subspace* S_{i^*} that $\boldsymbol{w}$ belongs to, allowing us to exploit standard methods to *localize* $\boldsymbol{w}$ within this subspace.

Our main contributions in this paper are two variants of an algorithm, **P**airw**I**se **C**omparisons for **A**ctive **S**ubspace **S**electi**O**n (PICASSO), to identify the subspace S_{i^*} that $\boldsymbol{w}$ lies in using paired comparisons between points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \cup_{i=1}^{K} S_i$. The first variant, *Distilled PICASSO*, admits sample complexity bounds that describe the number of measurements needed to correctly identify the true subspace S_{i^*}. The second variant, *Bayesian PICASSO*, tracks the posterior $P(i = i^*)$ and uses it to adaptively select measurements, improving performance in practice.

To the best of our knowledge, this is the first work to apply a union of subspaces model to point localization using paired comparisons. Once PICASSO identifies a subspace, the algorithm moves into a second phase where a state-of-the-art Bayesian embedding search method is deployed to localize $\boldsymbol{w}$ within that subspace [13].

2. THE PICASSO ALGORITHM

We first describe the probability model and general strategy underlying both PICASSO variants, then outline the specifics of each algorithm.

2.1. Probability model

The (potentially noisy) response $Y(\boldsymbol{x}_i, \boldsymbol{x}_j)$ indicates which of the points $\boldsymbol{x}_i$ or $\boldsymbol{x}_j$ the target search point $\boldsymbol{w}$ is closer to. In our notation convention, $Y(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ indicates the response that x_i is perceived to be closer, and $Y(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1$ indicates the response that x_j is perceived to be closer.

In our model, the probability of obtaining the measurement $Y(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ depends on the subspace membership of $\boldsymbol{w}$, $\boldsymbol{x}_i$, and $\boldsymbol{x}_j$:

- If $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, and $\boldsymbol{w}$ are all in the same subspace, the measurement is well-defined and depends on the relative distances of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from $\boldsymbol{w}$. In this case, $Y = 0$ is more likely when $\boldsymbol{x}_i$ is closer to $\boldsymbol{w}$ than $\boldsymbol{x}_j$ is.
- If $\boldsymbol{w}$ and $\boldsymbol{x}_i$ are contained in the same subspace but $\boldsymbol{x}_j$ is in a different subspace, only the distance between $\boldsymbol{w}$ and $\boldsymbol{x}_i$ is well defined and we expect to obtain $Y = 0$. Similarly, if $\boldsymbol{w}$ and $\boldsymbol{x}_j$ are contained in the same subspace but $\boldsymbol{x}_i$ is in a different subspace, we expect to obtain $Y = 1$.
- If $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, and $\boldsymbol{w}$ are in three different subspaces, the distances from $\boldsymbol{w}$ to both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are poorly defined, and we expect to obtain $Y = 0$ and $Y = 1$ with equal probability.

We operationalize this knowledge by denoting the subspace that point $\boldsymbol{x}_i$ lives in as $\sigma(x_i) = m\colon \boldsymbol{x}_i \in S_m$ and defining the measurement model

$$P\left(Y\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = 0 \mid \boldsymbol{w}, i^*\right) = \\ = \begin{cases} \frac{1}{1+e^{-\gamma\left(\boldsymbol{a}_{ij}^T \boldsymbol{w} - b_{ij}\right)}}, & \sigma(x_i) = \sigma(x_j) = i^* \\ 1-\rho, & \sigma(x_i) = i^*,\ \sigma(x_j) \neq i^* \\ \rho, & \sigma(x_i) \neq i^*,\ \sigma(x_j) = i^* \\ 0.5, & \sigma(x_i) \neq i^*,\ \sigma(x_j) \neq i^*. \end{cases} \tag{1}$$

Here, when $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are in the same subspace as $\boldsymbol{w}$ we use a logistic model where $\boldsymbol{a}_{ij} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2}$ and $b_{ij} = \frac{\|\boldsymbol{x}_i\|_2^2 - \|\boldsymbol{x}_j\|_2^2}{2\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2}$ define the hyperplane bisecting $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and γ represents a noise constant describing the reliability of the measurements. This type of distance-based response probability is a standard psychometric model used previously in paired comparison queries [13]. When only one of $\boldsymbol{x}_i$ or $\boldsymbol{x}_j$ is in the same subspace as $\boldsymbol{w}$, the small constant ρ represents the probability of the point not located in the same subspace as $\boldsymbol{w}$ erroneously being selected as closer to $\boldsymbol{w}$.

2.2. Algorithm overview

Our proposed PICASSO algorithm proceeds in two phases. In the first phase, we exploit the difference in response models to infer the subspace $\widehat{i^*}$ that $\boldsymbol{w}$ lives in. With this inferred subspace $\widehat{i^*}$ in hand, the second phase applies the adaptive Bayesian strategy of [13] to localize $\boldsymbol{w} \in S_{\widehat{i^*}}$.

The first variant of PICASSO collects a fixed number of queries per subspace, and the second variant starts with the full set of candidate subspaces $\mathcal{I} = \{1, \ldots, K\}$ and iteratively removes unlikely subspaces as measurements are adaptively collected. Our first variant, which we call Distilled PICASSO because of its resemblance to the distilled compressive sensing strategy of [14], admits a sample complexity bound that describes the number of queries needed to correctly identify the subspace i^* that $\boldsymbol{w}$ lies in. Our second variant, which we call Bayesian PICASSO, uses an adaptive Bayesian scheme to select measurements, improving performance in practice.

2.3. Variant 1: Distilled PICASSO

Our first subspace identification algorithm (listed in Algorithm 1) poses a scheduled sequence of paired comparisons between points in different subspaces to estimate i^*. For each candidate subspace, a fixed batch of comparisons is constructed such that one of the items lies in the subspace while the other item lies in a different subspace. The following theorem bounds the number of comparisons needed to identify the correct subspace using this strategy.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 10,2020 at 15:25:27 UTC from IEEE Xplore. Restrictions apply.

Theorem 2.1. *With probability at least $1-\delta$, Distilled PICASSO identifies the correct subspace i^* from K possible subspaces using T paired comparisons per subspace if*

$$T > \frac{2(K-1)^2}{K^2(\frac{1}{2}-\rho)^2} \ln \frac{K}{\delta},$$

resulting in $O(K \log \frac{K}{\delta})$ total paired comparisons required for accurate subspace recovery.

Importantly, note that the number of queries to identify the correct subspace scales only slightly worse than linearly in the number of subspaces and is independent of the number of possible query points $\{\boldsymbol{x}_i\}$. The proof follows from an application of concentration inequalities and a union bound using the probability response model defined in (1).

Proof. Without loss of generality, let $i^* = 1$. We have that $P(Y(\boldsymbol{x}_1, \boldsymbol{x}_j) = 0) = 1 - \rho \; \forall \; j > 1$, and $P(Y(\boldsymbol{x}_j, \boldsymbol{x}_k) = 0) = q$ for $j > 1$ where $q \equiv \frac{2\rho+K-2}{2(K-1)}$ and $\boldsymbol{x}_k$ is randomly selected as in Algorithm 1. Since each y_t is a Bernoulli random variable, by Hoeffding's inequality we have for $\varepsilon > 0$, $\forall \; j > 1$

$$P(r_1 < 1-\rho-\varepsilon) \leq e^{-2\varepsilon^2 T} \quad P(r_j > q+\varepsilon) \leq e^{-2\varepsilon^2 T}$$

From the union bound,

$$\begin{aligned} &P((r_1 < 1-\rho-\varepsilon) \vee (r_2 > q+\varepsilon) \vee (r_3 > q+\varepsilon) \dots) \\ \leq &P(r_1 < 1-\rho-\varepsilon) + P(r_2 > q+\varepsilon) + P(r_3 > q+\varepsilon) \dots \\ \leq &K e^{-2\varepsilon^2 T} \end{aligned}$$

$$\Longrightarrow P\left((r_1 > 1-\rho-\varepsilon) \bigwedge_{j=2}^{K} (r_j < q+\varepsilon) \right) \geq 1 - K e^{-2\varepsilon^2 T}$$

Let $\varepsilon = \frac{K(\frac{1}{2}-\rho)}{2(K-1)}$. Then with high probability, r_1 is greater than r_j for all $j > 1$, meaning that i^* is recovered accurately. To guarantee that i^* is recovered with probability at least $1-\delta$, we can set $T > \frac{2(K-1)^2}{K^2(\frac{1}{2}-\rho)^2} \ln \frac{K}{\delta}$. □

2.4. Variant 2: Bayesian PICASSO

Our second subspace identification algorithms (listed in Algorithm 2) starts with the full set of candidate subspaces $\mathcal{I} = \{1, \dots, K\}$ having the distribution $P(i = i^*) \sim \text{Unif}(\mathcal{I})$ and iteratively removes the unlikely ones. To form each paired comparison, points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are selected from the remaining candidate subspaces $\mathcal{I}$ such that $\sigma(\boldsymbol{x}_i) \neq \sigma(\boldsymbol{x}_j)$. After collecting the t^{th} paired comparison of the form (1), the posterior over the subspaces can be updated as

$$\begin{aligned} &P\left(i = i^* \mid y_{1:t-1}, Y(\boldsymbol{x}_j, \boldsymbol{x}_k) = y\right) \\ = &\frac{P\left(Y(\boldsymbol{x}_j, \boldsymbol{x}_k) = y \mid i = i^*, y_{1:t-1}\right) P\left(i = i^* \mid y_{1:t-1}\right)}{\sum\limits_{z \in \{0,1\}} P\left(Y(\boldsymbol{x}_j, \boldsymbol{x}_k) = z | i = i^*, y_{1:t-1}\right) P\left(i = i^* \mid y_{1:t-1}\right)}. \quad (2) \end{aligned}$$

Algorithm 1 Distilled PICASSO Algorithm

Require: T pairs per subspace

```
r[i] ← 0 ∀ i ∈ {1...K}
for i = 1,...K do
    for t = 1,...,T do
        Sample x_1 uniformly from S_i
        Sample j uniformly from {1 ≤ j ≤ K, j ≠ i}
        Sample x_2 uniformly from S_j
        Obtain measurement y_t ~ P(Y(x_1, x_2) | w, i*)
        r[i] ← r[i] + (1 − y_t)/T
    end for
end for
Return estimate î* = argmax_i r[i].
```

Algorithm 2 Bayesian PICASSO Algorithm

```
while P(i | y_{1:t}) < 1 − τ, i = 1,...,K do
    Sample i uniformly from I
    Sample x_1 uniformly from S_i
    Sample j uniformly from I \ {i}
    Sample x_2 from subspace S_j
    Obtain measurement y_t ~ P(Y(x_1, x_2) | w, i*).
    Update posterior P(i* | y_{1:t}) using (2).
    Update candidate subspaces I = {i: P(i|y_{1:t}) > τ}.
    t = t + 1.
end while
Return estimate î* = argmax_i P(i | y_{1:t}).
```

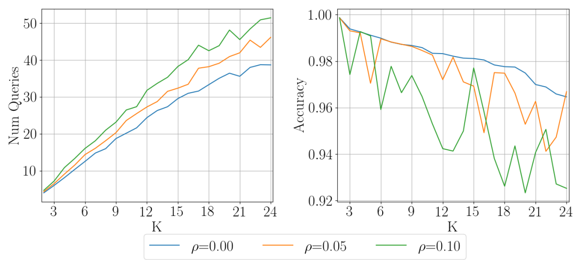
3. EXPERIMENTS

In this section, we demonstrate the performance of our proposed method using synthetic data with known ground truth. We proceed in two stages: First, we apply Bayesian PICASSO to determine the correct subspace S_{i^*} using paired comparisons. Second, we exploit this knowledge to localize $\boldsymbol{w} \in S_{i^*}$ using the *mean-cut, maximum variance* (MCMV) algorithm of [13].

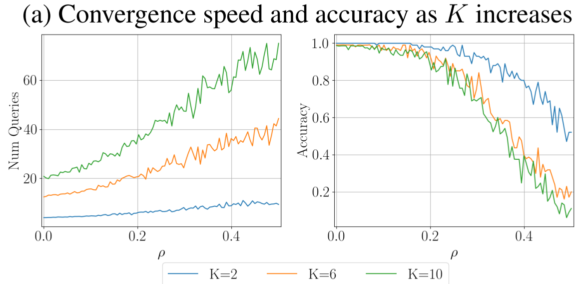
In each synthetic experiment, we randomly generate data in $\mathbb{R}^d$ lying in one of the K subspaces. Each subspace is defined by a random covariance matrix of rank $r = \frac{d}{2}$ and the data is generated by sampling from a Gaussian distribution with the defined covariance. We then run Bayesian PICASSO until convergence for different values of K and ρ. In our experiments, we set $d = 10$, $\tau = 0.01$

Figure 2 shows the performance of the Bayesian PICASSO algorithm as the problem difficulty increases in one of two ways. In Figure 2(a), we increase the number of candidate subspaces K. In Figure 2(b), we increase ρ, making it more difficult for the algorithm to determine whether $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are in the same subspace. We observe that PICASSO can recover the true subspace accurately but that its performance degrades as K and ρ increase. Notably, in Figure 2(a) the number of required comparisons roughly follows the

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 10,2020 at 15:25:27 UTC from IEEE Xplore. Restrictions apply.



(a) Convergence speed and accuracy as K increases



(b) Convergence speed and accuracy as ρ increases

Fig. 2. Performance of Bayesian PICASSO as difficulty increases with increases in the number of subspaces (K) and the value of the constant (ρ). "Num Queries" denotes the number of queries required for the algorithm to converge; accuracy denotes the rate of success in identifying the correct subspace.

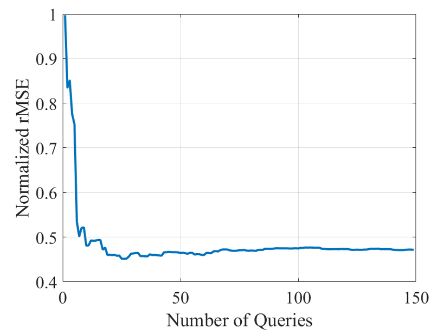


Fig. 3. Performance of point localization using MCMV as the number of queries increases.

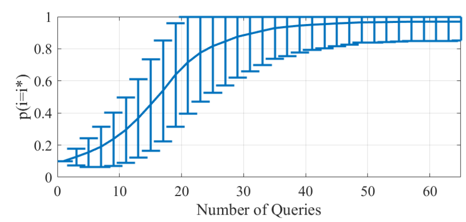


Fig. 4. Performance of subspace identification using Bayesian PICASSO on the CIFAR dataset versus number of queries.

$O(K \log K)$ trend described in Theorem 2.1. Figure 3 shows the accuracy of the second phase of the algorithm, which uses the MCMV algorithm of [13] to localize the point within the subspace identified in the first phase. The point localization error decreases sharply after just a few steps.

4. APPLICATION TO IMAGE SEARCH

We demonstrate the utility of PICASSO by applying it to the task of localizing an image in an embedding of the CIFAR-10 dataset [15], which consists of 32×32 images from 10 classes. The advantage of our method in this application is that by using our union of subspaces model with image classes representing subspaces, the search procedure is able to exploit class relationships between images during inference.

To generate an initial item embedding, we implement a VGG neural network classifier [16] and classify 5,000 images from the CIFAR test set. We use the classifier to generate both an estimated subspace label for each data point (represented by the classifier output), and a feature representation from an intermediate layer in the neural network, which we use as item locations and to calculate similarity. Here, we adapt the oracle model to work with the neural network by defining the function $\sigma(\cdot)$ in (1) as the classifier output and defining $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as the feature representations from an intermediate layer of a pre-trained VGG network.

Figure 4 shows the subspace identification accuracy averaged over 1000 trials. We observe that Bayesian PICASSO accurately estimates the correct subspace S_{i^*} as the number of queries reaches approximately 50.

5. DISCUSSION

In this paper we extended methods for using paired comparisons to localize a point in $\mathbb{R}^d$ to the setting where points instead lie in a union of subspaces. We introduced two algorithms for identifying which subspace an unknown point lies in given paired comparisons to points in known subspaces: Distilled PICASSO (which admits sample complexity bounds showing that the algorithm exploits the union of subspaces model to require provably fewer measurements than a naive unstructured model), and Bayesian PICASSO (which uses a fully Bayesian model and adaptive sampling scheme to identify the correct subspace with even fewer measurements).

While we have focused on the most challenging aspect of inference in the union of subspaces model (identifying the subspace the unknown point lies in), further improvements may be possible by utilizing the subspace structure in the localization phase of the algorithm. For example, the adaptive Bayesian strategy of [13] could be made more efficient in our framework by projecting points onto the identified subspaces. Other possible avenues for future work include an extension to the time-varying setting (e.g., using the Bayesian framework of [17]), extending analytical results to the Bayesian algorithm, and applying the union of subspaces model to perceptual compression in the latent space of generative models.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 10,2020 at 15:25:27 UTC from IEEE Xplore. Restrictions apply.

6. REFERENCES

[1] B. Carterette, P. Bennett, M. Chickering, and S. Dumais, “Here or there: Preference judgments for relevance,” in *Proc. European Conf. on Inf. Retrieval (ECIR)*, Glasgow, Scotland, Apr. 2008.

[2] E. Hullermeier, J. Fürnkranz, W. Cheng, and K. Brinker, “Label ranking by learning pairwise preferences,” *Artif. Intell.*, vol. 172, no. 16-17, pp. 1897–1916, 2008.

[3] Nir Ailon, “Active learning ranking from pairwise preferences with almost optimal query complexity,” in *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Grenada, Spain, Dec. 2011.

[4] K. Jamieson and R. Nowak, “Active ranking using pairwise comparisons,” in *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Granada, Spain, Dec. 2011.

[5] S. Negahban, S. Oh, and D. Shah, “Iterative ranking from pair-wise comparisons,” in *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Lake Tahoe, California, Dec. 2012.

[6] F. Wauthier, M. Jordan, and N. Jojic, “Efficient ranking from pairwise comparisons,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, Jun. 2013.

[7] B. Eriksson, “Learning to Top-K search using pairwise comparisons,” in *Proc. Int. Conf. Art. Intell. Stat. (AIStats)*, Scottsdale, AZ, Apr. 2013.

[8] N. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. Wainwright, “Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence,” in *Proc. Int. Conf. Art. Intell. Stat. (AIStats)*, San Diego, CA, May 2015.

[9] A. Massimino and M. Davenport, “The geometry of random paired comparisons,” in *Proc. IEEE Conf. Acoustics, Speech, Signal Processing (ICASSP)*, New Orleans, Louisiana, Mar. 2017.

[10] A. Massimino and M. Davenport, “As you like it: Localization via paired comparisons,” *submitted to J. Mach. Learn. Res.*, 2018.

[11] Ehsan Elhamifar and Rene Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[12] John Lipor and Laura Balzano, “Leveraging union of subspace structure to improve constrained clustering,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2130–2139.

[13] G. Canal, A. Massimino, M. Davenport, and C. Rozell, “Active embedding search via noisy paired comparisons,” in *Proc. Int. Conf. on Machine Learning (ICML)*, 2019, pp. 902–911.

[14] J. Haupt, R. Baraniuk, R. Castro, and R. Nowak, “Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements,” in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*. IEEE, 2009, pp. 1551–1555.

[15] A. Krizhevsky, G. Hinton, et al., “Learning multiple layers of features from tiny images,” Tech. Rep., Citeseer, 2009.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[17] G. Canal, M. O’Shaughnessy, C. Rozell, and M. Davenport, “Joint estimation of trajectory and dynamics from paired comparisons,” in *Proc. IEEE Int. Work. on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Guadeloupe, West Indies, 2019.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 10,2020 at 15:25:27 UTC from IEEE Xplore. Restrictions apply.