A Sparse Integrative Cluster Analysis for Understanding Soybean Phenotypes

Jinbo Bi, Jiangwen Sun, Tingyang Xu, Jin Lu
Department of Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
jinbo@engr.uconn.edu

Yansong Ma, Lijuan Qiu
Institute of Crop Science
Chinese Academy of Agricultural Sciences
Beijing, China
qiu_lijuan@263.net

***Abstract*—Soybean is one of the most important crops for food, feed and bio-energy world-wide. The study of soybean phenotypic variation at different geographical locations can help the understanding of soybean domestication, population structure of soybean, and the conservation of soybean biodiversity. We investigate if soybean varieties can be identified that they differ from other varieties on multiple traits even when growing at different geographical locations. When a collection of traits are observed for the same soybean type at different locations (different views), joint analysis of the multiple-view data is required in order to identify the same soybean clusters based on data from different locations. We employ a new multi-view singular value decomposition approach that simultaneously decomposes the data matrix gathered at each location into sparse singular vectors. This approach is able to group soybean samples consistently across the different locations and simultaneously identify the phenotypes at each location on which the soybean samples within a cluster are the most similar. Comparison with several latest multi-view co-clustering methods demonstrates the superior performance of the proposed approach.**

***Keywords*-multi-view data analysis, multi-view clustering, soybean population structure, soybean trait analysis**

I. Introduction

Soybean is cultivated globally, in part because it produces among the highest gross oil output - with the highest protein content - of any vegetable crop [1]. There is cytological, biochemical and molecular evidence that supports the domestication of soybean from *Glycine soja*, a wild annual species that is native throughout China, and parts of Korea, Japan and Russia. Wild soybean is the closest wild relative of the cultivated soybean (*Glycine max*). A long history of domestication, cultivation and breeding has narrowed the genetic basis of cultivated soybean, limiting further improvement of crop yield and quality. In contrast, wild soybeans, which inhabit a wide range of eco-geographic regions in East Asia, have diverse genetic variability in pest and disease resistance genes and other useful agricultural and ecological characteristics [2]. However, global climate change and the destruction of the ecological balance have sped up the extinction rate of the wild species [3]. Comprehensive and extensive investigation of the population genetic structure and the phenotypic variability of wild and cultivated soybean is important. Besides several early works that studied the population genetic structure of both cultivated and wild soybean [4], [5], in this work, we focus on the understanding of the phenotypic variability and similarity of soybean populations via an advanced clustering method - multi-view bi-clustering where we identify subgroups of soybean varieties according to their phenotypes observed at different eco-geographical locations.

In the existing statistic and machine learning literature, multi-view data analysis methods include supervised/semi-supervised co-training [6], [7], [8], unsupervised co-clustering [9], [10], [11], [12], [13], [14] or multi-view feature learning [15], [16] where samples are characterized or viewed in multiple ways, thus creating multiple sets of input variables. When the majority of the data is unlabeled, co-training improves the classification accuracy by enforcing consistency between the classification decisions of the unlabeled data determined by the models learned independently from each of the views. For co-clustering, there are two types of methods: (1) biclustering [14], [17], [18], also called two-mode clustering [11], simultaneously clusters the rows and columns of a data matrix; (2) multi-view co-clustering [19], [12], [13], [15] seeks clusterings that are consistent across different views. The first type of co-clustering is similar to another set of algorithms [20], [9] that search subspaces, each of which corresponds to a view of the data and gives different clusters in different subspaces. Biclustering and subspace searching essentially find subspaces to define clusters only in one view of data or one source of data. Another set of multi-view subspace learning algorithms search for a low dimensional representation of data that enables accurate reconstruction [16].

Our problem, most similar to multi-view co-clustering, seeks a grouping of subjects that is in accordance with each other between different views. However, existing multi-view clustering methods all assume that an underlying partition exists and all given variables in each view are used to reveal this underlying partition. In our problem, clusters may exist in different subspaces, and an underlying partition consistent across all views may be revealed by the identification of the features or subspaces that specify the clusters. If a data matrix has rows represent subjects and columns represent features. This problem can be viewed as performing bi-

978-1-4799-5669-2/14/$31.00 ©2014 IEEE

clustering in each view to identify both row clusters and column clusters simultaneously but the row clusters from the different views should be the same. Figure 1 demonstrates the problem in two views with two data matrices. In an ideal clustering solution, each group of subjects (rows) shows high similarity over a subset of variables in view 1 as well as a subset of variables in view 2.

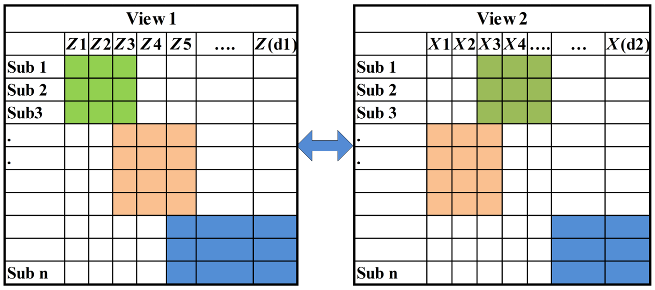

Figure 1. Sparse co-clustering: rows are grouped in the same way across the two matrices. The subjects in each row cluster are homogeneous over a subset of variables from each of the views.

II. Multi-view data matrix decomposition

Given a single data matrix $\mathbf{X}$ of size n-by-d, a subgroup of its rows and a subgroup of its columns can be simultaneously achieved by the sparse singular value decomposition (SSVD) of $\mathbf{X}$ [21]. The SSVD requires both the left and right singular vectors to be sparse. Let $\mathbf{u}$ of size n and $\mathbf{v}$ of size d be two singular vectors resulted from the SSVD. Their outer product forms a sparse low-rank approximation of the original matrix, $\mathbf{X} \approx \sigma \mathbf{u}\mathbf{v}^T$ where σ is the corresponding singular value. Then, rows in $\mathbf{X}$ corresponding to non-zero components in $\mathbf{u}$ form a row subgroup and columns in $\mathbf{X}$ corresponding to non-zero components in $\mathbf{v}$ form a column subgroup. The resulted row and column clusters help to define each other. The SSVD finds all singular vectors sequentially by repeatedly solving the following problem:

$$\begin{aligned} \min_{\sigma,\mathbf{u},\mathbf{v}} \quad & \|\mathbf{X} - \sigma\mathbf{u}\mathbf{v}^T\|_F^2 + \lambda_u\|\mathbf{u}\|_1 + \lambda_v\|\mathbf{v}\|_1 \\ \text{subject to} \quad & \|\mathbf{u}\|_2 = 1, \|\mathbf{v}\|_2 = 1, \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, the first item in the objective function reflects the low-rank approximation error, $\|\cdot\|_1$ is the ℓ_1 vector norm to enforce the sparsity of $\mathbf{u}$ and $\mathbf{v}$, and λ_u and λ_v are two tuning parameters to balance off the approximation error and sparsity regularizers. To obtain subsequent singular vectors, the SSVD solves Eq.(1) repeatedly using a new $\mathbf{X}$ which excludes subjects already identified in a row cluster.

Now we extend the SSVD to two or more data matrices denoted by $\mathbf{X}^k$ of size n-by-d_k, $k = 1, \cdots, m$. These m data matrices characterize the same set of subjects from m different views. We can obtain $\mathbf{u}^k$ and $\mathbf{v}^k$ for each matrix $\mathbf{X}^k$ by the sparse singular value decomposition of each individual $\mathbf{X}^k$, separately. However, it will not guarantee the row clusters specified, respectively, by $\mathbf{u}^k$ be consistent. To make them consistent, it requires all $\mathbf{u}^k$, $k = 1, \cdots, m$, to have non-zero components at the same positions. Notice that the $\mathbf{u}^k$ vectors are not necessarily the same given they may be derived from very different features in the different views, such as real-valued features in gene expression data but discrete features in genetic markers.

We propose to use a binary vector $\boldsymbol{\omega}$ of size n that serves as a common factor to link the different views. We multiply each component of $\mathbf{u}^k$ by the corresponding component of $\boldsymbol{\omega}$, i.e., $u_i^k = u_i^k \omega_i$. In other words, we represent each vector $\mathbf{u}^k$ by $\text{diag}(\boldsymbol{\omega})\mathbf{u}^k$ where $\text{diag}(\boldsymbol{\omega})$ is a diagonal matrix with diagonal entries equal to $\boldsymbol{\omega}$. When $\omega_i = 0$, the i-th components of all $\mathbf{u}^k$'s will be 0, and consequently, the i-th subject will be excluded from the subgroup in all views. We hence require the sparsity of $\boldsymbol{\omega}$ instead of individual $\mathbf{u}$'s in the optimization problem as follows:

$$\begin{aligned} \min_{\boldsymbol{\omega},\sigma_k,\mathbf{u}^k,\mathbf{v}^k,k=1,\cdots,m} \quad & \sum_{k=1}^{m} \|\mathbf{X}^k - \sigma_k \text{diag}(\boldsymbol{\omega})\mathbf{u}^k\mathbf{v}^{kT}\|_F^2 \\ & + \lambda\|\boldsymbol{\omega}\|_1 + \sum_{k=1}^{m}\lambda_k\|\mathbf{v}^k\|_1 \\ \text{subject to} \quad & \|\mathbf{u}^k\|_2 = 1, \ \|\mathbf{v}^k\|_2 = 1, \\ & k = 1, \cdots, m, \\ & \boldsymbol{\omega} \in \mathcal{B}_n. \end{aligned} \tag{2}$$

where $\mathcal{B}_n$ is the set that contains all binary vectors of length n, λ, and λ_k are tuning parameters to balance the errors and sparsity regularizers, and λ_k's, $k = 1, \cdots, m$ can be used to balance between different views if certain views are more sparse than others.

As an alternative, a restricted version of Eq(2) may require all $\mathbf{u}$'s to be the same, and then use the same $\mathbf{u}$ in the approximation of all matrices $\mathbf{X}^k$ in Eq.(2). By requiring $\mathbf{u}$ to be sparse, it can also identify consistent row clusters across all views. Although the resultant optimization problem is easier to solve without integer variables in $\boldsymbol{\omega}$, it imposes an unnecessarily stringent constraint to limit the search space only to those that satisfy $\mathbf{u}^1 = \mathbf{u}^2 = \cdots = \mathbf{u}^m$, which rules out many potential solutions that may include the optimal row clusters. Another alternative is to minimize the pairwise differences between $\mathbf{u}^i$ and $\mathbf{u}^j$, which suffers from the same over-constrained problem as the exact values of the difference are not concerned. Our problem only seeks the indicators of whether or not a component of $\mathbf{u}$ is zero.

III. A fast and effective algorithm

The proposed formulation (2), although is a mixed-integer program, can be effectively solved after proper relaxations. We design an alternating optimization algorithm to solve problem (2) by splitting the variables into three working sets: one set consists of $\mathbf{u}$'s; one set consists of $\mathbf{v}$'s; and the last set consists of the binary variables in $\boldsymbol{\omega}$. We optimize the variables in one working set at a time alternatively whereas fixing the others.

(1) Find the optimal $\mathbf{u}^k$ and $\mathbf{v}^k$ with fixed $\boldsymbol{\omega}$

When $\boldsymbol{\omega}$ is fixed, Problem (2) can be decomposed to optimize with respect to each individual view. For view k, we obtain $\mathbf{u}^k$ and $\mathbf{v}^k$ by solving the following optimization problem:

$$\begin{aligned}\min_{\sigma_k,\mathbf{u}^k,\mathbf{v}^k} \quad & \|\mathbf{X}^k - \sigma_k(\text{diag}(\boldsymbol{\omega})\mathbf{u}^k)\mathbf{v}^{kT}\|_F^2 + \lambda_k\|\mathbf{v}^k\|_1 \\ \text{subject to} \quad & \|\mathbf{u}^k\|_2 = 1, \|\mathbf{v}^k\|_2 = 1\end{aligned} \tag{3}$$

which can be solved by solving the following two sub-problems in alternative iterations.

(a) Solve for $\mathbf{v}^k$ when $\mathbf{u}^k$ is fixed

We solve the following problem for the optimal $\tilde{\mathbf{v}}^k$ by relaxing the unit length constraint on $\mathbf{v}^k$. Then, we set $\tau = \|\tilde{\mathbf{v}}^k\|_2$ and obtain $\mathbf{v}^k = \tilde{\mathbf{v}}^k/\tau$. The singular value σ_k also needs to be updated by $\tau\sigma_k$.

$$\min_{\tilde{\mathbf{v}}^k} \quad \|\mathbf{X}^k - \sigma_k\text{diag}(\boldsymbol{\omega})\mathbf{u}^k\tilde{\mathbf{v}}^{kT}\|_F^2 + \lambda_k\|\tilde{\mathbf{v}}^k\|_1.$$

Following the same derivation in the single-view SSVD, each component $\tilde{v}_j^k$ in $\tilde{\mathbf{v}}^k$ can be analytically computed by soft-thresholding as discussed in [21]. Let $\mathbf{X}_{(i,\cdot)}^k$ and $\mathbf{X}_{(\cdot,j)}^k$ denote the i-th row and j-th column of the matrix $\mathbf{X}^k$, respectively. The closed-form solution of $\tilde{\mathbf{v}}^k$ is written as:

$$\tilde{v}_j^k = \begin{cases} \alpha_j - \beta, & \alpha_j > \beta \\ 0, & |\alpha_j| \le \beta \\ \alpha_j + \beta, & \alpha_j < -\beta \end{cases}, \; j = 1, \cdots, d. \tag{4}$$

where $\alpha_j = (\text{diag}(\boldsymbol{\omega})\mathbf{u}^k)^T\mathbf{X}_{(\cdot,j)}^k/\sigma_k$ and $\beta = \lambda_k/(2\sigma_k)$.

(b) Solve for $\mathbf{u}^k$ when $\mathbf{v}^k$ is fixed

We now optimize Problem (3) with respect to $\mathbf{u}^k$. Again, we relax the unit length constraint on $\mathbf{u}^k$ first and then rescale it back to unit length. We solve the following problem for the best $\tilde{\mathbf{u}}^k$. Then, we obtain $\mathbf{u}^k = \tilde{\mathbf{u}}^k/\tau$ where $\tau = \|\tilde{\mathbf{u}}^k\|_2$ and also update σ_k by $\tau\sigma_k$.

$$\min_{\tilde{\mathbf{u}}^k} \quad \|\mathbf{X}^k - \sigma_k\text{diag}(\boldsymbol{\omega})\tilde{\mathbf{u}}^k\mathbf{v}^{kT}\|_F^2.$$

Each component $\tilde{u}_i^k$ of $\tilde{\mathbf{u}}^k$ can be independently and analytically computed as follows:

$$\tilde{u}_i^k = \begin{cases} \dfrac{\mathbf{X}_{(i,\cdot)}^k\mathbf{v}^k}{\omega_i\sigma_k}, & \text{if } \omega_i \ne 0 \\ 0, & \text{if } \omega_i = 0. \end{cases}, \; i = 1, \cdots, n. \tag{5}$$

(2) Find the optimal $\boldsymbol{\omega}$ with fixed $\mathbf{u}$'s and $\mathbf{v}$'s

When all $\mathbf{u}$'s and $\mathbf{v}$'s are fixed in Problem (2), the optimization problem becomes:

$$\min_{\boldsymbol{\omega}\in\mathcal{B}_n} \quad \sum_{k=1}^{m} \|\mathbf{X}^k - \sigma_k\text{diag}(\boldsymbol{\omega})\mathbf{u}^k\mathbf{v}^{kT}\|_F^2 + \lambda\|\boldsymbol{\omega}\|_1$$

We initially solve the relaxed $\tilde{\boldsymbol{\omega}}$ which takes real values, and then calculate the binary $\boldsymbol{\omega}$ from $\tilde{\boldsymbol{\omega}}$ by proper re-scaling. First, the above optimization problem can be re-written into the following equivalent form when $\boldsymbol{\omega}$ is relaxed:

$$\min_{\tilde{\boldsymbol{\omega}}} \quad \|\mathbf{X} - \text{diag}(\boldsymbol{\omega})\mathbf{E}\|_F^2 + \lambda\|\tilde{\boldsymbol{\omega}}\|_1$$

where $\mathbf{X} = [\mathbf{X}^1 \;\; \mathbf{X}^2 \;\; \cdots \;\; \mathbf{X}^m]$ is obtained by concatenating the data matrices in columns, $\mathbf{E} = [\sigma_1\mathbf{u}^1\mathbf{v}^{1T} \;\; \sigma_2\mathbf{u}^2\mathbf{v}^{2T} \cdots \sigma_m\mathbf{u}^m\mathbf{v}^{mT}]$ by concatenating the low-rank approximation matrices in columns. Then, each component $\tilde{\omega}_i$ of $\tilde{\boldsymbol{\omega}}$ can be independently and analytically computed as

$$\tilde{\omega}_i = \begin{cases} \alpha_i - \beta, & \alpha_i > \beta \\ 0, & |\alpha_i| \le \beta \\ \alpha_i + \beta, & \alpha_i < -\beta \end{cases}, \; i = 1, \cdots, n. \tag{6}$$

where $\alpha_i = \frac{\mathbf{E}_{(i,\cdot)}\mathbf{X}_{(i,\cdot)}^T}{\|\mathbf{E}_{(i,\cdot)}\|_2^2}$ and $\beta = \frac{\lambda}{2\|\mathbf{E}_{(i,\cdot)}\|_2^2}$. Formula (6) is derived based on the same scheme in [21] as how Eq.(4) is derived.

After obtaining $\tilde{\boldsymbol{\omega}}$, the binary vector $\boldsymbol{\omega}$ can be calculated as:

$$\omega_i = \begin{cases} 1, & \text{if } \tilde{\omega}_i \ne 0 \\ 0, & \text{if } \tilde{\omega}_i = 0 \end{cases}. \tag{7}$$

In order to keep the objective of Eq.(2) unchanged, we need to update $\mathbf{u}^k$, $k = 1, \cdots, m$, accordingly as follows:

$$u_i^k = \begin{cases} u_i^k/\tilde{\omega}_i, & \text{if } \tilde{\omega}_i \ne 0, \\ 0, & \text{if } \tilde{\omega}_i = 0, \end{cases}, \quad i = 1, \cdots, n. \tag{8}$$

The singular values σ_k will be recalculated as: $\sigma_k = \sigma_k\|\mathbf{u}^k\|_2$, and we normalize $\mathbf{u}^k$ by $\mathbf{u}^k = \mathbf{u}^k/\|\mathbf{u}^k\|_2$ for all $k = 1, \cdots, m$.

Algorithm 1 summarizes the main steps in our algorithm. The proposed algorithm alternates between solving the above sub-problems until a local minimizer is reached. For fixed λ parameters, the objective function of Eq.(2) is bounded from below by a constant. This objective is monotonically non-increasing when minimizing each sub-problem, and hence the convergence of this iterative process is guaranteed. In our experiments, when the ℓ_2 norm of the difference vector between two consecutive $\boldsymbol{\omega}$ falls below a pre-defined threshold of $\epsilon = 10^{-4}$, we set the algorithm to terminate. On both synthetic and real world datasets, this iterative process reached a stationary point in about 10 iterations.

At each iteration, only simple closed-form solutions need to be computed for each working group of variables, so the algorithm is fast and scalable. When optimizing the m pairs of $\mathbf{u}^k$ and $\mathbf{v}^k$, the algorithm requires a computation cost of $O(nmd)$. For optimizing $\boldsymbol{\omega}$, the most costly steps are the vector product of $\mathbf{E}_{(i,\cdot)}\mathbf{X}_{(i,\cdot)}^T$ and the evaluation of $\|\mathbf{E}_{(i,\cdot)}\|_2^2$, which requires a computation cost of $O(md)$. Hence this

Algorithm 1 Multi-view Data Matrix Decomposition

```
Input: $\mathbf{X}^k$, $\lambda_k$, $k = 1, \cdots, m$, and $\lambda$
Output: $\boldsymbol{\omega}$, $\sigma_k$, $\mathbf{u}^k$, $\mathbf{v}^k$, $k = 1, \cdots, m$
Initialize $\boldsymbol{\omega}$ with a vector of all ones
Initialize $\mathbf{u}^k$'s by the corresponding left singular vectors of $\mathbf{X}^k$, $k = 1, \cdots, m$
repeat
  for $k = 1$ to $m$ do
      Compute $\tilde{\mathbf{v}}^k$ by Eq.(4)
      Compute $\mathbf{v}^k$ from $\tilde{\mathbf{v}}^k$ and update $\sigma_k$
      Compute $\tilde{\mathbf{u}}^k$ by Eq.(5)
      Compute $\mathbf{u}^k$ from $\tilde{\mathbf{u}}^k$ and update $\sigma_k$
  end for
  Compute $\boldsymbol{\omega}$ by Eqs. (6)-(7)
  Update $\sigma_k$, $\mathbf{u}^k$ by Eq.(8), $k = 1, \cdots, m$ accordingly.
until The vector $\boldsymbol{\omega}$ reaches a fixed point
```

step takes a cost of $O(nmd)$ as well. Overall, this algorithm takes computation time of $O(nmd)$, which is in the linear order of the problem dimensions. Moreover, notice that (i) when $\boldsymbol{\omega}$ is fixed, the optimization of $\mathbf{u}^k$, $\mathbf{v}^k$ is independent from each other among the views; (ii) when calculating any of the $\mathbf{u}$, $\mathbf{v}$ and $\boldsymbol{\omega}$ vectors, each component of the vector can be computed independently from other components of the vector. Hence, this algorithm is readily parallelizable and can be distributed if more processors are available to further reduce the computation time.

To derive the subsequent row and column subgroups, we repeat Algorithm 1 using new matrices $\mathbf{X}^k$ that exclude the rows corresponding to the subjects in the identified subgroups. By repeating this procedure, the desired number of subject (row) subgroups can be achieved.

IV. Computational Results

We implemented the multi-view data matrix decomposition algorithm and validated it using a multi-site soybean dataset. This study aims to examine whether or not our algorithm can reveal the underlying variables, from each data matrix (each site), that are associated with the clusters. We compared the proposed approach against several most recent multi-view co-clustering methods as follows.

- **Single view biclustering (SVB):** Clusters were included in the comparison by running the method of SSVD-based biclustering [21] using one view of data. We reported the best performance when experimenting with different views.
- **Co-trained spectral (CTS):** This method was proposed in [12]. It also finds consistent row clusters based on spectral clustering where eigenvector representations of each view are co-trained or modified by the clustering results from other views.
- **Co-regularized spectral (CRS):** This method was proposed in [13] for finding consistent row clusters across multiple views. It applies spectral clustering to each view together with a co-regularization factor applied to the eigenvector representations of different views. We used the pairwise co-regularized formulation in [13].
- **Kernel addition (KA), Kernel product (KP)** These three baseline methods were formulated in [13] by summation or component-wise multiplication of two kernel matrices for use in spectral clustering. We used the same procedure as in [13] in our experiments. We adopted the widely used Gaussian kernel to compute the similarity between each pair of soybean samples grown at a location.

A. Soybean data

A total of 123 soybean species were considered. Their morphological characters (nine of them) were evaluated at four geographical locations in 2011 in Heilongjiang province and Jilin province of China. For each population, mature seeds were collected from each individual, with an interval of >5m between individuals. Seeds were obtained by the Institute of Crop Science of the Chinese Academy of Agricultural Sciences and the Wuhan botanical Garden of the Chinese Academy of Sciences. Randomized blocks design and two replications were executed and nine adjacent individual characters, which included plant height, number of branch, number of main stem nod, number of pod, number of individual seed, 100-seed weight, yield of blocks, content of protein and content of oil, were recorded during mature period in every replication. The average of records was calculated and presented in the cultivar observation at each location. Hence, in our data, although totally there were 36 characters/traits, they were grouped into four sets, each corresponding to a geographical location.

B. Tuning of our algorithm

An important aspect to obtain good clustering performance by our algorithms is the tuning of the hyperparameters. In the proposed approach, there are trade-off parameters: λ for sparsity of the common factor $\boldsymbol{\omega}$, which leads to the row clusters consistent across the views, and λ_k for sparsity of the column clusters in each view (i.e., feature selection in each view). Since in our experiment all the four locations of phenotypic data were equally important, we chose to use the same value for all λ_k, $k = 1, \cdots, 4$.

We experimented with various values of λ and λ_k's and observed their effects on the clustering performance. Ideally, we want our clusters to contain a reasonable number of samples. The clusters should not be empty but should not contain the full set of sample either. Similiarly, for feature selection, for each cluster, we would like to avoid the case to select either all features or zero feature. These were the basic and naïve criteria used in our experiments to choose appropriate values of the hyperparameters. Figure 2 shows the performance change when we varied the values of λ and λ_k for identifying the first cluster. From the figure, we can see many choices of λ and λ_k are not appropriate as they either end up a cluster of all subjects or zero subjects or use no or all features. Based on the figure, the values in the bright blue color area would be good choices. We hence chose $\lambda = 0.18$ and $\lambda_k = 0.65$ when we ran the algorithm to identify the first cluster. When the second cluster was to be identified, the hyperparametes were chosen in the same way by drawing two figures similiar to Figure 2, and we obtained $\lambda = 0.16$ and $\lambda_k = 0.6$. After two clusters were sequentially identified, the remaining subjects were treated as in the third cluster.

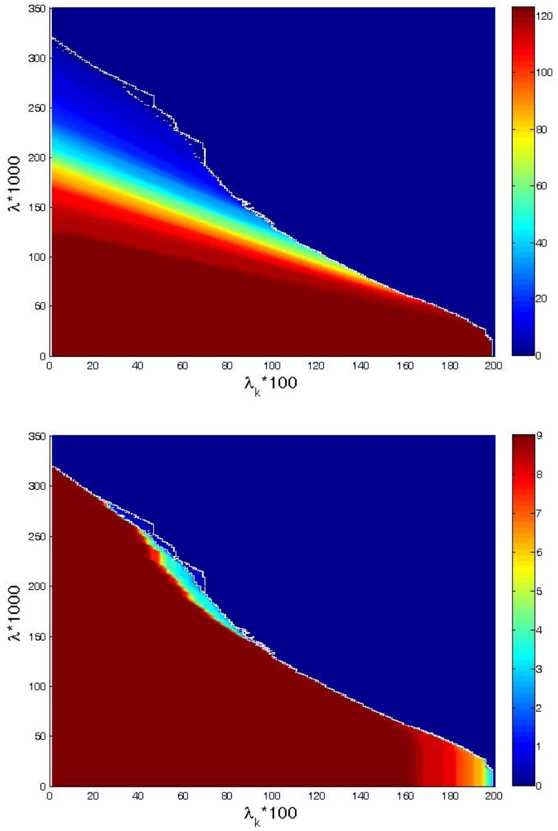

Figure 2. Cluster performance when varying λ and λ_k. Top: the number of subjects in the first cluster; Bottom: the number of features on which subjects in the first cluster differ from other subjects.

C. Clustering results

In our experiments with the soybean dataset, three clusters were created: Cluster 1 consisted of only 16 soybean samples; Cluster 2 was the largest and contained 77 soybean samples; and the last cluster contained the remaining 30 samples. Our algorithm automatically selected features/variables based on which the samples were grouped into Cluster 1 or Cluster 2 (the rest went to Cluster 3). Traits were selected from each of the locations for Cluster 1: 5 traits at location 1 (Branch, Pod, Seed, Yield and Percentage of Oil); 5 traits at location 2 (Height, Branch, Nod, Pod and Seed); 5 traits at location 3 (Height, Branch, Nod, Pod, and Seed); and 5 traits at location 4 (Branch, Nod, Pod, Seed, and Percentage of Oil). Based on the selected features for Cluster 2, samples in Cluster 2 were similar on 5 traits at location 1 (Nod, Yield, Weight, Percentage of Protein, Percentage of Oil), on the Branch trait at location 2 and location 4, and on 4 traits at location 3 (Branch, Pod, Seed and Yield).

To further demonstrate the similarity within clusters and dissimilarity between clusters, we draw the bar plots for the mean values of the 9 traits at each of the four locations in Figure 3. Based on Figure 3, the most distinguishable characteristics we can see is that samples in Cluster 1 differ significantly from other clusters on the the traits of Branch, Pod and Seed at locations 1 and 3. Samples in Cluster 2 differ significantly from other clusters on the trait of Branch at locations 2 and 4. Samples in Clusters 1 and 2 differ from the remaining cluster on the Yield trait at all locations. All of these differences were at the significance level of $p < 0.01$ (with χ^2-test).

D. Comparison results

All of the compared methods were used to obtain three soybean subgroups. To evaluate the clustering performance, we built three classifiers using each view of the data to separate samples in one cluster from the rest. Totally, twelve classifiers were built for each cluster solution resulted from a comparison method. Receiver operating characteristic (ROC) curves were plotted and the area under the ROC curve (AUC) values were used to compare the different cluster solutions, in other words, to compare the different methods. The average AUC values obtained from each of the compared methods averaged over the three clusters and over the four locations are plotted in Figure 4.

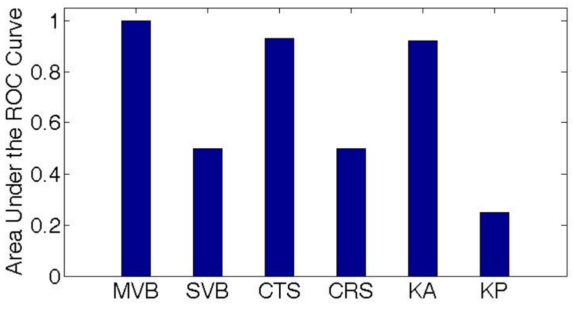

Figure 4. The AUC comparison of the classifiers built to evaluate the separability of the clusters resulted from each of the comparison methods. The higher an AUC value, the better separation among the clusters. The different methods include the proposed multi-view bicluster (MVB), single view biclustering (SVB), co-trained spectral (CTS) clustering, co-regulaized spectral (CRS) clustering, kernel addition (KA) and kernel product (KP).

The averaged AUC value can be treated as a measure for separability among the clusters based on each single view of the data. The proposed method achieved the best performance on this measurement whereas the co-training spectral

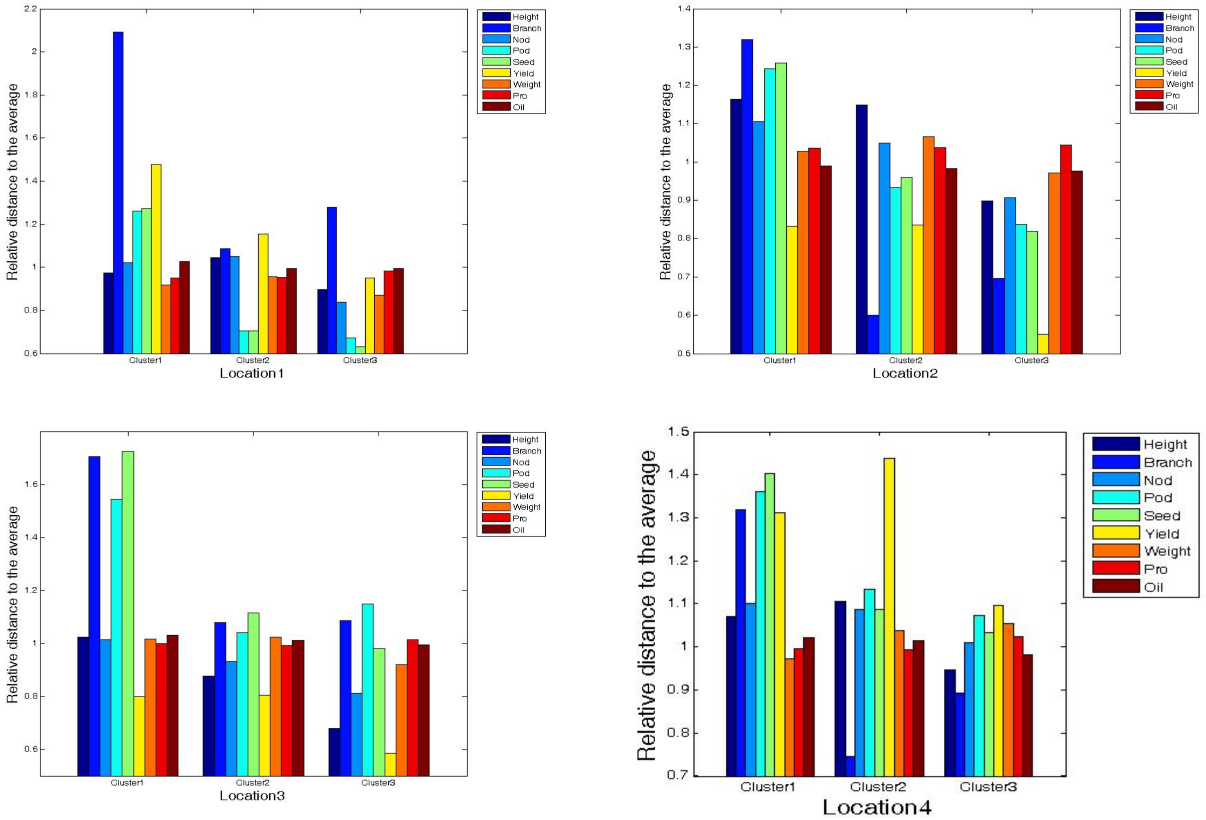

Figure 3. The characteristics of the three clusters viewed at each of the four locations. The average values of each trait at a location are computed for each cluster, and plotted in four grouped bar plots, each corresponding to a location.

clustering and kernel addition methods achieved slightly worse separability than our method. The co-regularized spectral clustering, although a state of the art, performed similarly to the single view biclustering. It is likely because the co-regularization factor was applied to the eigenvector representations of different views where eigenvectors were those of the Gaussian kernel. This method hence did not select features from the raw feature space rather it mapped to an eigenvector space. Kernel product was obviously an improper choice for the soybean data.

V. Conclusion

In this paper, we have proposed a multi-view sparse clustering approach based on matrix decomposition of multiple data matrices simultaneously into sparse singular vectors. This approach links different views of data by a binary vector that is used to enforce the row clusters from all views to be consistent. Surprisingly, the resultant optimization problem is efficiently solvable using only closed-form formulas in alternating minimization steps. To the best of our knowledge, our work is the first approach that extends *sparse* matrix decomposition to multi-view data. As matrix decomposition methods are the fundamental tools for many learning tasks, the capability of extending them to learn jointly from multiple views of data will enhance many applications not only in co-clustering. For instance, unsupervised dimension reduction, such as PCA, can directly benefit from the proposed multi-view matrix decomposition approach (e.g., PCA from multi-view SVD and kernel PCA from multi-view eigen-decomposition).

There are a few directions for future work. It is possible to extend the proposed approach to the case when missing values are present in any of the views. A simple idea is to recover the missing values in one view based on information from other views. Theoretical analysis of co-clustering in general has not been fully explored. Consistency analysis of multi-view SVD-based or eigen-decomposition-based co-clustering will provide insights into the rate of convergence as the sample size increases. If partial data is labeled, generalization of the proposed framework to the semi-supervised setting will also be important. Although our algorithm is computationally efficient, more empirical evaluations on large-scale datasets might be needed to examine its speed and scalability.

Acknowledgments

This work was supported by the research grants of National Science Foundation of the United States DBI-1356655, IIS-1320586 and IIS-1447711. Correspondence should be addressed to Jinbo Bi for the details of the

analytics, and should be addressed to Lijuan Qiu for the details of the soybean data set.

References

[1] A. I. Mohamed and M. Rangappa, "Nutrient composition and anti-nutritional factors in vegetable soybean: Ii. oil, fatty acids, sterols, and lipoxygenase activity," *Food Chemistry*, vol. 44, no. 4, pp. 277–282, 1992.

[2] R. Hajjar and T. Hodgkin, "The use of wild relatives in crop improvement: a survey of developments over the last 20 years," *Euphytica*, vol. 156, no. 1, pp. 1–13, 2007.

[3] C. Thomas, A. Grainger, L. Hannah, L. Hughes, B. Huntley, A. van Jaarsveld, G. Midgley, L. Miles, M. Ortega-Huerta, A. Peterson, O. Phillips, S. Williams, A. Cameron, R. Green, M. Bakkenes, L. Beaumont, Y. Collingham, B. Erasmus, and M. de Siqueira, "Biodiversity conservation - uncertainty in predictions of extinction risk - effects of changes in climate and land use - climate change and extinction risk - reply," *Nature*, vol. 430, no. 6995, pp. 1–2, 2004.

[4] J. Guo, Y. Liu, Y. Wang, J. Chen, Y. Li, H. Huang, L.-j. Qiu, and Y. Wang, "Population structure of the wild soybean (glycine soja) in china: implications from microsatellite analyses," *Annals of botany*, vol. 110, no. 4, pp. 777–785, 2012.

[5] Y.-H. Li, W. Li, C. Zhang, L. Yang, R.-Z. Chang, B. S. Gaut, and L.-J. Qiu, "Genetic diversity in domesticated soybean (glycine max) and its wild progenitor (glycine soja) for simple sequence repeat and single?nucleotide polymorphism loci," *New Phytologist*, vol. 188, no. 1, pp. 242–253, 2010.

[6] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory*. New York, NY, USA: ACM, 1998, pp. 92–100.

[7] M.-F. Balcan, A. Blum, and K. Yang, "Co-training and expansion: Towards bridging theory and practice," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 89–96.

[8] S. Yu, B. Krishnapuram, R. Bharat Rao, and R. Rosales, "Bayesian co-training," *Journal of Machine Learning Research*, vol. 12, pp. 2649–2680, 2011.

[9] Y. Guan, J. Dy, and M. Jordan, "A unified probabilistic model for global and local unsupervised feature selection," in *Proceedings of the International Conference on Machine Learning 2011*, 2011, pp. 1073–1080.

[10] K. Sohn and E. Xing, "A hierarchical dirichlet process mixture model for haplotype reconstruction from multi-population data," *Annals of Applied Statistics*, vol. 3, no. 2, pp. 791–821, 2009.

[11] I. Van Mechelen, H.-H. Bock, and P. De Boeck, "Two-mode clustering methods: a structured overview," *Statistical methods in medical research*, vol. 13, no. 5, pp. 363–394, 2004.

[12] A. Kumar and H. Daume III, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th International Conference on Machine Learning*, L. Getoor and T. Scheffer, Eds. New York, NY, USA: ACM, 2011, pp. 393–400.

[13] A. Kumar, P. Rai, and H. Daume III, "Co-regularized multi-view spectral clustering," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 1413–1421.

[14] S. Ji, W. Zhang, and J. Liu, "A sparsity-inducing formulation for evolutionary co-clustering," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 334–342.

[15] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 129–136.

[16] M. White, Y. Yu, X. Zhang, and D. Schuurmans, "Convex multi-view subspace learning," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1682–1690.

[17] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 89–98.

[18] H. Shan and A. Banerjee, "Bayesian co-clustering," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ser. ICDM '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 530–539. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2008.91

[19] M. Culp and G. Michailidis, "A co-training algorithm for multi-view data with applications in data fusion," *J. Chemometr. Journal of Chemometrics*, vol. 23, no. 6, pp. 294–303, 2009.

[20] D. Niu, J. G. Dy, and M. I. Jordan, "Multiple Non-Redundant Spectral Clustering Views," in *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

[21] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron, "Biclustering via sparse singular value decomposition," *Biometrics*, vol. 66, no. 4, pp. 1087–95, 2010.