Abstract—Prediction of annual crop yields at a county granularity is important for national food production and price stability. In this paper, towards the goal of better crop yield prediction, leveraging recent graph signal processing (GSP) tools to exploit spatial correlation among neighboring counties, we denoise relevant features via graph spectral filtering that are inputs to a deep learning prediction model. Specifically, we first construct a combinatorial graph with edge weights that encode county-to-county similarities in soil and location features via metric learning. We then denoise features via a maximum a posteriori (MAP) formulation with a graph Laplacian regularizer (GLR). We focus on the challenge to estimate the crucial weight parameter $\mu$, trading off the fidelity term and GLR, that is a function of noise variance in an unsupervised manner. We first estimate noise variance directly from noise-corrupted graph signals using a graph clique detection (GCD) procedure that discovers locally constant regions. We then compute an optimal $\mu$ minimizing an approximate mean square error function via bias-variance analysis. Experimental results from collected USDA data show that using denoised features as input, performance of a crop yield prediction model can be improved noticeably.

Index Terms—Graph spectral filtering, unsupervised learning, bias-variance analysis, crop yield prediction

I. INTRODUCTION

As weather patterns become more volatile due to unprecedented climate change, accurate crop yield prediction—forecast of agriculture production such as corn or soybean at a county / state granularity—is increasingly important in agronomics to ensure a robust and reliable national food supply [1]. A conventional crop yield prediction scheme gathers relevant features that influence crop production—e.g., soil composition, precipitation, temperature—as input to a deep learning (DL) model such as convolutional neural network (CNN) [2] and long short-term memory (LSTM) [3] to estimate yield per county / state in a supervised manner. While this is feasible when the training dataset is sufficiently large, the trained model is nonetheless susceptible to noise in feature data, typically collected by USDA from satellite images and farmer surveys1. In this paper, we focus on the problem of pre-denoising relevant features prior to DL model training to improve crop yield prediction performance.

Given that basic environmental conditions such as soil makeup, rainfall and drought index at one county are typically similar to nearby ones, one would expect crucial features directly related to crop yields, such as normalized difference vegetation index (NDVI) and enhanced vegetation index (EVI) [4], at neighboring counties to be similar as well. To exploit these inter-county similarities for feature denoising, leveraging recent rapid progress in graph signal processing (GSP) [5], [6] we pursue a graph spectral filtering approach. While graph signal denoising is now well studied in many contexts, including general band-limited graph signals [7], 2D images [8], [9], and 3D point clouds [10], [11], our problem setting for crop feature denoising is particularly challenging because of its unsupervised nature. Specifically, an obtained feature $\mathbf{y}$ \in $\mathbb{R}^{N}$ for $N$ counties is typically noise-corrupted, and one has no access to ground truth data $\mathbf{x}^o$ nor knowledge of the noise variance $\sigma^2$. Thus, the important weight parameter $\mu$ that trades off the fidelity term $\|\mathbf{y} - \mathbf{x}^o\|_2^2$ against the graph signal prior such as the graph Laplacian regularizer $\mathbf{x}^T \Lambda \mathbf{x}$ [8] or graph total variation (GTV) [12] in a maximum a posteriori (MAP) formulation cannot be easily derived [13] or trained end-to-end [9] as previously done.

In this paper, we focus on the unsupervised estimation of the weight parameter $\mu$ in a GLR-regularized MAP formulation for relevant feature pre-denoising to improve crop yield prediction. Specifically, we first construct a combinatorial graph $\mathcal{G}$ with edge weights $w_{i,j}$ encoding similarities between counties (nodes) $i$ and $j$. $w_{i,j}$ is inversely proportional to the Mahalanobis distance $d_{i,j} = (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j)$, where $\mathbf{f}_i$ is a vector for node $i$ composed of soil and location features, and $\mathbf{M}$ is an optimized metric matrix [14]. We then estimate noise variance $\sigma^2$ directly from noise-corrupted features using our proposed graph clique detection (GCD) procedure, generalized from noise estimation in 2D imaging [15]. Finally, we derive equations analyzing the bias-variance tradeoff [13] to minimize the resulting MSE of our MAP estimate and compute the optimal weight parameter $\mu$. See Fig. 1 for an illustration of a similarity graph connecting neighboring counties in Iowa with undirected edges, where the set of feature values per county is shown as a discrete signal on top of the graph.

Using USDA corn data from 10 states in the corn belt (Iowa, Illinois, Indiana, Ohio, Nebraska, Minnesota, Wisconsin, Michigan, Michigan, Michigan, Michigan, Michigan, Michigan), 2\text{L} is the combinatorial graph Laplacian matrix; definitions are formally defined in Section III-A.

Gene Cheung acknowledges the support of the NSERC grants RGPIN-2019-06271, RGPAS-2019-00110.

1https://www.usda.gov/
Missouri, and Kentucky) containing 938 counties, experimental results show that using our GLR-regularized denoiser with optimized \( \mu \) to denoise two important EVI features led to improved performance in an DL model [16]: reduction of root mean square error (RMSE) [17] by 0.434% in crop yield prediction compared to the baseline when the features were not pre-denoised.

The paper is organized as follows. We first overview our crop yield prediction model in Section II. We then present our unsupervised feature denoising algorithm in Section III. Finally, we present experimental results and conclusion in Section IV and V, respectively.

II. PREDICTION FRAMEWORK

We overview a conventional crop yield prediction framework [2], [3]. First, relevant features at a county level such as silt / clay / sand percentage in soil, accumulated rainfall, drought index, and growing degree days (GDD) are collected from various sources, including USDA and satellite images. Features of the same county are inputted to a DL model like CNN or LSTM for future yield prediction, trained in a supervised manner using annual county-level yield data provided by USDA. Note that existing yield prediction schemes [2], [3] focus mainly on exploiting temporal correlation (both short-term and long-term) to predict future crop yields.

Depending on the type of features, the acquired measurements may be noise-corrupted. This may be due to measurement errors by faulty mechanical instruments, human errors during farmer surveys, etc. Given that environmental variables are likely similar in neighboring counties, one would expect similar basic features in a local region. To exploit this spatial correlation for feature denoising, we employ a graph spectral approach to be described next.

III. FEATURE DENOISING

A. Preliminaries

An \( N \)-node undirected positive graph \( G(\mathcal{N}, \mathcal{E}, \mathbf{W}) \) can be specified by a symmetric adjacency matrix \( \mathbf{W} \in \mathbb{R}^{N \times N} \), where \( W_{i,j} = w_{i,j} > 0 \) is the weight of an edge \( (i, j) \in \mathcal{E} \) connecting nodes \( i, j \in \mathcal{N} = \{1, \ldots, N\} \), and \( W_{i,j} = 0 \) if there is no edge \( (i, j) \notin \mathcal{E} \). Here we assume there are no self-loops, and thus \( W_{i,i} = 0, \forall i \). Diagonal degree matrix \( \mathbf{D} \in \mathbb{R}^{N \times N} \) has diagonal entries \( D_{i,i} = \sum_{j \neq i} W_{i,j} \). We can now define the combinatorial graph Laplacian matrix \( \mathbf{L} \triangleq \mathbf{D} - \mathbf{W} \), which is positive semi-definite (PSD) for a positive graph \( G \) (i.e., \( W_{i,j} \geq 0, \forall i, j \)) [6].

An assignment of a scalar \( x_i \) to each graph node \( i \in \mathcal{N} \) composes a graph signal \( \mathbf{x} \in \mathbb{R}^N \). Signal \( \mathbf{x} \) is smooth with respect to (w.r.t.) graph \( G \) if its variation over \( G \) is small. A popular graph smoothness measure is the graph Laplacian regularizer (GLR) \( \mathbf{x}^\top \mathbf{L} \mathbf{x} \) [8], i.e., \( \mathbf{x} \) is smooth iff \( \mathbf{x}^\top \mathbf{L} \mathbf{x} \) is small. Denote by \( (\lambda_k, \mathbf{v}_k) \) the \( k \)-th eigen-pair of matrix \( \mathbf{L} \) and \( \mathbf{V} \) the eigen-matrix composed of eigenvectors \( \{\mathbf{v}_k\}_{k=1}^N \) as columns. \( \mathbf{V}^\top \) is known as the Graph Fourier Transform (GFT) [6] that converts a graph signal \( \mathbf{x} \) to its graph frequency representation \( \mathbf{x} = \mathbf{V}^\top \mathbf{x} \). GLR can be expanded as

\[
\mathbf{x}^\top \mathbf{L} \mathbf{x} = \sum_{(i,j) \in \mathcal{E}} w_{i,j} (x_i - x_j)^2 = \sum_{k=1}^N \lambda_k \mathbf{x}_k^2.
\] (1)

Thus, a small GLR means that a connected node pair \( (i, j) \in \mathcal{E} \) with large edge weight \( w_{i,j} \) has similar sample values \( x_i \) and \( x_j \) in the nodal domain, and most signal energy resides in low graph frequency coefficients \( \mathbf{x}_k \) in the spectral domain—\( \mathbf{x} \) is a low-pass (LP) signal.

B. Graph Metric Learning

Assuming that each node \( i \in \mathcal{N} \) is endowed with a length-\( K \) feature vector \( \mathbf{f}_i \in \mathbb{R}^K \), one can compute edge weight \( w_{i,j} \) connecting nodes \( i \) and \( j \) in \( G \) as

\[
w_{i,j} = \exp \left\{ - (f_i - f_j)^\top \mathbf{M}(f_i - f_j) \right\}
\] (2)

where \( \mathbf{M} \geq 0 \) is a PSD metric matrix that determines the square Mahalanobis distance (feature distance) \( d_{i,j} = (f_i - f_j)^\top \mathbf{M}(f_i - f_j) \geq 0 \) between nodes \( i \) and \( j \). There exist metric learning schemes [14], [18] that optimize \( \mathbf{M} \) given an objective function \( f(\mathbf{M}) \) and training data \( \mathcal{X} = \{x_1, \ldots, x_T\} \). For example, we can define \( f(\mathbf{M}) \) using GLR and seek \( \mathbf{M} \) by minimizing \( f(\mathbf{M}) \):

\[
\min_{\mathbf{M} \geq 0} f(\mathbf{M}) = \sum_{t=1}^T \mathbf{w}_t^\top \mathbf{L}(\mathbf{M})\mathbf{x}_t.
\] (3)

In this paper, we adopt an existing metric learning scheme [14], and use soil- and location-related features—clay percentage, available water storage estimate (AWS), soil organic carbon stock estimate (SOC) and 2D location features—to compose \( \mathbf{f}_i \in \mathbb{R}^5 \). These features are comparatively noise-free and thus reliable. We use also these features as training data \( \mathcal{X} \) to optimize \( \mathbf{M} \), resulting in graph \( \mathcal{G} \). Node pair \( (i, j) \) with distance \( d_{i,j} \) larger than a threshold has no edge \( (i, j) \notin \mathcal{E} \). We will use \( \mathcal{G} \) to denoise two EVI features that are important for yield prediction.

C. Denoising Formulation

Given a constructed graph \( \mathcal{G} \) specified by a graph Laplacian matrix \( \mathbf{L} \), one can denoise a target input feature \( \mathbf{y} \in \mathbb{R}^N \) using a MAP formulation regularized by GLR [8]:

\[
\min_{\mathbf{x}} \| \mathbf{y} - \mathbf{x} \|_2^2 + \mu \mathbf{x}^\top \mathbf{L} \mathbf{x}
\] (4)

where \( \mu > 0 \) is a weight parameter trading off the fidelity term and GLR. Given \( \mathbf{L} \) is PSD, objective (4) is convex with a system of linear equations as solution:

\[
(\mathbf{I} + \mu \mathbf{L}) \mathbf{x}^* = \mathbf{y}.
\] (5)

Given that matrix \( \mathbf{I} + \mu \mathbf{L} \) is symmetric, positive definite (PD) and sparse, (5) can be solved using conjugate gradient (CG) [19] without matrix inverse. We focus on the selection of \( \mu \) in (4) next.

D. Estimating Noise Variance

In our feature denoising scenario, we first estimate the noise variance \( \sigma^2 \) directly from noisy feature (signal) \( \mathbf{y} \), using which weight parameter \( \mu \) in (4) is computed. We propose a noise estimation procedure called graph clique detection (GCD) when a graph \( \mathcal{G} \) encoded with inter-node similarities is provided.

We generalize from a noise estimation scheme for 2D images [15]. First, we identify locally constant regions (LCRs) \( \mathcal{R}_m \) where signal samples are expected to be similar, i.e., \( x_i \approx x_j, \forall i, j \in \mathcal{R}_m \). Then, we compute mean \( \bar{x}_m = \frac{1}{\left| \mathcal{R}_m \right|} \sum_{i \in \mathcal{R}_m} x_i \) and variance \( \sigma^2_m = \frac{1}{\left| \mathcal{R}_m \right|} \sum_{i \in \mathcal{R}_m} (x_i - \bar{x}_m)^2 \) for each \( \mathcal{R}_m \). Finally, we compute the global noise variance as the weighted average:

\[
\sigma^2 = \sum_m \frac{\left| \mathcal{R}_m \right|}{\sum_k \left| \mathcal{R}_k \right|} \sigma^2_m.
\] (6)

The crux thus resides in the identification of LCRs in a graph. Note that this does not imply conventional graph clustering [20]; grouping of all graph nodes to two or more non-overlapping sets. There is no requirement here to put every node in a LCR.
Fig. 2: (a) Example of a 10-node graph, where edges with weights less than threshold \( \hat{w} \) are colored in green; (b) The resulting k-hop connected graph (KCG) for \( k = 2 \), after removing green edges and creating edges (colored in magenta) by connecting 2-hop neighbors. Two maximal cliques (out of 5) in KCG are highlighted.

We describe our proposal based on cliques. A clique is a (sub-)graph where every node is connected with every other node in the (sub-)graph. Thus, a clique implies a node cluster with strong inter-node similarities, which we assume is roughly constant. Given an input graph \( \mathcal{G} \), we identify cliques in \( \mathcal{G} \) as follows.

1) k-hop Connected Graph: We first sort \( M = |E| \) edges in \( \mathcal{G} \) in weights from smallest to largest. For a given threshold weight \( \hat{w} \) (to be discussed) and \( k \in \mathbb{Z}^+ \), we remove all edges \((i,j) \in E\) with weights \( w_{i,j} < \hat{w} \) and construct a k-hop connected graph (KCG) \( \mathcal{G}^{(k)} \) with edges connecting nodes \( i \) and \( j \) that are \( k \)-hop neighbors in \( \mathcal{G} \). If \( \mathcal{G}^{(k)} \) has at least a target \( \hat{E} \) number of edges, then it is a feasible KCG, with minimum connectivity \( C(\mathcal{G}^{(k)}) = \hat{w}^k \). \( C(\mathcal{G}^{(k)}) \) is the weakest possible connection between two connected nodes in \( \mathcal{G}^{(k)} \), interpreting edge weights \( w_{i,j} \) as conditional probabilities as done in Gaussian Markov Random Field (GMRF) [21].

To find threshold \( \hat{w} \) for a given \( k \), we seek the largest \( \hat{w} \) for feasible graphs \( \mathcal{G}^{(k)} \) (with minimum \( \hat{E} \) edges) via binary search among \( M \) edges in complexity \( O(\log M) \). We initialize \( k = 1 \), compute threshold \( \hat{w} \), then increment \( k \) and repeat the procedure until we identify a maximal \( \mathcal{G}^{(k)} \) with \( \hat{E} \) edges for \( k = 1, 2, \ldots \).

See Fig. 2(b) for an example of a KCG \( \mathcal{G}^{(2)} \) given original graph \( \mathcal{G} \) in Fig. 2(a). We see, for example, that edge \((3, 10)\) is removed from \( \mathcal{G} \), but edge \((4, 10)\) is added in \( \mathcal{G}^{(2)} \) because nodes 4 and 10 are 2-hop neighbors in \( \mathcal{G} \). The idea is to identify strongly similar pairs in original \( \mathcal{G} \) and connect them with explicit edges in \( \mathcal{G}^{(k)} \). Then the maximal cliques\(^3\) are discovered using algorithm in [22], as shown in Fig. 2(b). The cliques in the resulting graph \( \mathcal{G}^{(k)} \) are LCRs used to calculate the noise variance via (6).

2) Target \( \hat{E} \) Edges: The last issue is the designation of targeted \( \hat{E} \) edges. \( \hat{E} \) should be chosen so that each clique \( m \) discovered in KCG \( \mathcal{G}^{(k)} \) has enough nodes to reliably compute mean \( \hat{x}_m \) and variance \( \sigma^2_m \). We estimate \( \hat{E} \) as follows. Given input graph \( \mathcal{G} \), we first compute the average degree \( d \). Then, we target a given clique to have an average of \( n_c \) nodes—large enough to reliably compute mean and variance. Thus, the average degree of the resulting graph can be approximated as \( d + n_c - 1 \). Finally, we compute \( \hat{E} \approx N(d + n_c - 1) \), where \( N \) is the number of nodes in the input graph \( \mathcal{G} \).

E. Deriving Weight Parameter

Having estimated a noise variance \( \sigma^2 \), we now derive the optimal weight parameter \( \mu \) for MAP formulation (4). Following the derivation in [13], given \( \sigma^2 \), the mean square error (MSE) of the MAP estimate \( x^\ast \) from (4) computed using ground truth signal \( x \) as function of \( \mu \) is

\[
\text{MSE}(\mu) = \sum_{i=1}^{N} \psi_i^2(\nu_i^\top x^\ast)^2 + \sigma^2 \sum_{i=1}^{N} \phi_i^2
\]

where \( \psi_i = \frac{1}{1+\mu \nu_i^\top \nu_i} \) and \( \phi_i = \frac{1}{1+\mu \nu_i^\top \nu_i} \). The first term \( B(\mu) \) corresponds to the bias of estimate \( x^\ast \), which is a differentiable, concave and monotonically increasing function of \( \mu > 0 \). In contrast, the second term \( V(\mu) \) corresponds to the variance of \( x^\ast \), and is a differentiable, convex monotonically decreasing function of \( \mu > 0 \). When combined, MSE is a differentiable and provably pseudo-convex function of \( \mu > 0 \) [23], i.e.,

\[
\nabla \text{MSE}(\mu_1) \cdot (\mu_2 - \mu_1) \geq 0 \rightarrow \text{MSE}(\mu_2) \geq \text{MSE}(\mu_1),
\]

\[\forall (\mu_1, \mu_2) > 0. \text{ See Fig. 3 for an example of bias } B(\mu), \text{ variance } V(\mu) \text{ and } \text{MSE}(\mu) \text{ for a specific graph signal } x \text{ and a graph } \mathcal{G}, \text{ and Appendix A for a proof of pseudo-convexity.} \]

In [13], the authors derived a corollary where MSE(\mu) is replaced by a convex upper bound \( \text{MSE}^+(\mu) \) that is more easily computable. The optimal \( \mu \) is then computed by minimizing the convex function \( \text{MSE}^+(\mu) \) using conventional optimization methods. However, this upper bound is too loose in practice to be useful.

Instead, we take an alternative approach: we approximate (7) by modeling the distributions of eigenvalues \( \lambda_i \)’s of \( \mathbf{L} \) and signal energies \( \alpha_i^2 = (\nu_i^\top x^\ast)^2 \) at graph frequencies \( i \) as follows. We model \( \lambda_i \)’s as an exponentially increasing function \( f(i) \), and model \( \alpha_i^2 \) as an exponentially decreasing function \( g(i) \), namely

\[
\lambda_i \approx f(i) = q \exp\{\gamma i\}; \quad \alpha_i^2 \approx g(i) = r \exp\{-\theta i\},
\]

Fig. 3: Bias \( B(\mu) \), variance \( V(\mu) \), and MSE(\mu) as functions of weight parameter \( \mu \), for a signal with respect to a graph constructed in Section III-B. The underlying graph is constructed by connecting adjacent counties in Iowa. Graph signal \( x^\ast \) is the clay percentages in each county. We assume \( \sigma^2 = 1 \) when computing \( B(\mu), V(\mu), \text{ and } \text{MSE}(\mu) \) in (7).

Fig. 4: (a) Modeling \( \lambda_i \)’s as an exponentially increasing function \( f(i) \); (b) modeling \( \alpha_i^2 \) as an exponentially decreasing function \( g(i) \).
where \( q, r, \theta \) are parameters. See Fig. 4 for illustrations of both approximations. To compute those parameters, we first compute extreme eigen-pairs \((\lambda_i, \nu_i)\) for \( i \in \{2, N\} \) in linear time using LOBPCG [24]. Hence we have following expressions from (9),

\[
\lambda_2 \approx q \exp\{2\gamma\}; \quad \lambda_N \approx q \exp\{N\gamma\},
\]

\[
\alpha_2^2 \approx r \exp\{-2\theta\}; \quad \alpha_N^2 \approx r \exp\{-N\theta\}.
\] (10)

By solving these equations, one can obtain the four parameters as

\[
\gamma = \frac{\ln \frac{\lambda_2}{\lambda_N}}{N - 2}; \quad q = \lambda_2 \exp\{-2 \ln \frac{\lambda_2}{\lambda_N} (N - 2)\},
\]

\[
\theta = \frac{\ln \frac{\alpha_2^2}{\alpha_N^2}}{N - 2}; \quad r = \alpha_2^2 \exp\{-2 \ln \frac{\alpha_2^2}{\alpha_N^2} (N - 2)\}.
\] (11)

One can thus approximate MSE in (7) as

\[
\text{MSE}^a(\mu) = \sum_{i=2}^{N} \frac{g(i)}{1 + \mu f(i)}^2 + \sigma^2 \sum_{i=2}^{N} \frac{1}{(1 + \mu f(i))^2}.
\] (12)

IV. EXPERIMENTATION

A. Experimental Setup

To test the effectiveness of our proposed feature pre-denoising algorithm, we conducted the following experiment. We used the corn yield data at the county level between year 2010 and 2019 provided by USDA and the National Agricultural Statistics Service\(^3\) to predict yields in 2020. We performed our experiments in 938 counties in 10 states (Iowa, Illinois, Indiana, Kentucky, Michigan, Minnesota, Missouri, Nebraska, Ohio, Wisconsin) in the corn belt. As discussed in Section III-B, we used five soil- and location-related features to compose feature \( f_i \), \( i \in \mathbb{R}^2 \) for each county \( i \) and metric learning algorithm in [14] to compute metric matrix \( M \), in order to build a similarity graph \( G \). For feature denoising, we targeted enhanced Vegetation Index (EVI) for the months of June and July, EVI\(_{June}\) and EVI\(_{July}\). In a nutshell, EVI quantifies vegetation greenness per area based on captured satellite images, and is an important feature for yield prediction. EVI is noisy for a variety of reasons: low-resolution satellite images, cloud occlusion, etc. We built a DL model for yield prediction based on XGBoost [16] as the baseline, using which different versions of EVI\(_{June}\) and EVI\(_{July}\) were injected as input along with other relevant features.

B. Experimental Results

First, we computed the optimum weight parameter \( \mu \) for both EVI\(_{June}\) and EVI\(_{July}\), which was \( \mu = 2.2 \). Table I shows the crop yield prediction performance using noisy features versus denoised features with different weight parameters, under three metrics in the yield prediction literature: root-mean-square error (RMSE), Mean Absolute Error (MAE) and \( R^2 \) score (larger the better) [17]. Results in Table I demonstrate that our optimal weight parameter (i.e., 2.2) has the best results among other \( \mu \) values. Specifically, our denoised features can reduce RMSE by 0.434%.

In addition to the metrics in Table I, we measured the permutation feature importance [25] for both EVI\(_{June}\) and EVI\(_{July}\) before and after denoising. Fig. 5 shows that the importance of these features increases after denoising, demonstrating the positive effects of our unsupervised feature denoiser. Specifically, the result for the optimal \( \mu = 2.2 \) induced the most feature importance.

| TABLE I: Performance Metrics with different weight parameters |
|---------------------------------------------------------------|
| Metric          | Original | \( \mu = 0.001 \) | \( \mu = 0.01 \) | \( \mu = 2.2 \) |
| RMSE (bu/acre)  | 14.139   | 14.1966          | 14.2042         | 14.0776         |
| MAE (bu/acre)   | 11.225   | 11.2635          | 11.2674         | 10.9839         |
| \( R^2 \)       | 0.3894   | 0.3860           | 0.3856          | 0.3929          |

Further, we calculated the correlation between the original / denoised feature EVI\(_{June}\) and EVI\(_{July}\) and actual crop yield. Fig. 6 shows that the denoised features with the optimal \( \mu = 2.2 \) has the largest correlation with the crop yield feature. In comparison, using previous method in [13] to estimate \( \mu = 3.35 \) resulted in a weaker correlation. Lastly, to visualize the effect of our denoising algorithm, Fig. 7 shows the yield prediction error for different counties in the 10 states in the corn belt. We observe that with the exception of a set of counties in southern Iowa devastated by a rare strong wind event (called derecho) in 2020, there were very few noticeably large yield prediction errors.

\(^3\)https://quickstats.nass.usda.gov/
V. Conclusion

Conventional crop yield prediction schemes exploit only temporal correlation to estimate future yields per county given input relevant features. In contrast, to exploit inherent spatial correlations among neighboring counties, we perform graph spectral filtering to pre-denoise input features for a deep learning model prior to network parameter training. Specifically, we formulate the feature denoising problem via a MAP formulation with the graph Laplacian regularizer (GLR). We derive the weight parameter \( \mu \) trading off the fidelity term against GLR in two steps. We first estimate noise variance directly from noisy observations using a graph clique detection (GCD) procedure that discovers locally constant regions. We then compute an optimal \( \mu \) minimizing an MSE objective via bias-variance analysis. Experiments show that using denoised features as input can improve a DL models' crop yield prediction.

Appendix A

Proof of Pseudo-convexity for (7)

We rewrite (7) as:

\[
\text{MSE}(\mu) = \sum_{i=2}^{N} \frac{\mu^2 \lambda_i \sigma_i^2 + \sigma_i^2}{(1 + \mu \lambda_i)^2} + \sigma_i^2, \tag{15}
\]

where \( \sigma_i^2 = (\mathbf{v}_i^T \mathbf{x})^2 \). For simplicity, we only provide the proof for \( N = 1 \). In this case, we can write,

\[
\nabla \text{MSE}(\mu) = \frac{2 \mu \lambda_1 \sigma_1^2}{(1 + \mu \lambda_1)^3} - \frac{2 \lambda_1 \sigma_1^2}{(1 + \mu \lambda_1)^3}, \tag{16}
\]

Thus, the following expressions follow naturally:

\[
\mu \geq \frac{\sigma_1^2}{\lambda_1} > 0 \Rightarrow \nabla \text{MSE}(\mu) \geq 0; \tag{17a}
\]

\[
0 < \mu < \frac{\sigma_1^2}{\lambda_1} \Rightarrow \nabla \text{MSE}(\mu) < 0. \tag{17b}
\]

Further, according to (17), for \( \mu_1 \geq \frac{\sigma_2^2}{\lambda_2 \sigma_2^2} > 0, \)

\[
(\mu_2 - \mu_1) \geq 0 \Rightarrow (\text{MSE}(\mu_2) - \text{MSE}(\mu_1)) \geq 0, \tag{18}
\]

and for \( 0 < \mu_1 < \frac{\sigma_2^2}{\lambda_2 \sigma_2^2}, \)

\[
(\mu_2 - \mu_1) < 0 \Rightarrow (\text{MSE}(\mu_2) - \text{MSE}(\mu_1)) \geq 0. \tag{19}
\]

Now, by combining (17), (18), and (19), one can write (8), which concludes the proof.

References

[1] Y. Cai, “Crop Yield Predictions – High Resolution Statistical Model for Intra-season Forecasts Applied to Corn in the US,” in AGU Fall Meeting Abstracts, vol. 2017, Dec. 2017, pp. GC31G–47.
[2] S. Khaki, L. Wang, and S. V. Archontoulis, “A CNN-RNN framework for crop yield prediction,” *Frontiers in Plant Science*, vol. 10, 2020.
[3] J. Sun, L. Di, Z. Sun, Y. Shen, and Z. Lai, “County-level soybean yield prediction using deep cnn-lstm model,” *Sensors*, vol. 19, no. 20, 2019 [Online]. Available: https://www.mdpi.com/1424-8220/19/20/4363
[4] B. Matsushita, W. Yang, J. Chen, Y. Onda, and G. Qiu, “Sensitivity of the enhanced vegetation index (evi) and normalized difference vegetation index (ndvi) to topographic effects: a case study in high-densely forest,” *Sensors*, vol. 7, no. 11, pp. 2636–2651, 2007.
[5] A. Ortega et al., “Graph signal processing: Overview, challenges, and applications,” in *Proceedings of the IEEE*, vol. 106, no.5, May 2018, pp. 808–828.
[6] G. Cheung et al., “Graph spectral image processing,” in *Proceedings of the IEEE*, vol. 106, no.5, May 2018, pp. 907–930.
[7] S. Chen, A. Sandryhaila, J. Moura, and J. Kovacevic, “Signal recovery on graphs: Variation minimization,” in *IEEE Transactions on Signal Processing*, vol. 63, no.17, September 2015, pp. 4609–4624.
[8] J. Pang and G. Cheung, “Graph Laplacian regularization for inverse imaging: Analysis in the continuous domain,” in *IEEE Transactions on Image Processing*, vol. 26, no.4, April 2017, pp. 1770–1785.
[9] H. Vu et al., “Unrolling of deep graph total variation for image denoising,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2050–2054.
[10] J. Zeng et al., “3D point cloud denoising using graph Laplacian regularization of a low dimensional manifold model,” *IEEE Transactions on Image Processing*, vol. 31, pp. 4374–4389, 2020.
[11] C. Dinesh, G. Cheung, and J. V. Bajic, “Point cloud denoising via feature graph Laplacian regularization,” *IEEE Transactions on Image Processing*, vol. 32, pp. 4143–4158, 2020.
[12] Y. Bai, G. Cheung, X. Liu, and W. Gao, “Graph-based blind image deblurring from a single photograph,” in *IEEE Transactions on Image Processing*, vol. 31, no.3, March 2019, pp. 1404–1418.
[13] P.-Y. Chen and S. Liu, “Bias-variance tradeoff of graph Laplacian regularizer,” *IEEE Signal Processing Letters*, vol. 47, no. 8, pp. 1118–1122, 2017.
[14] W. Hu et al., “Feature graph learning for 3D point cloud denoising,” *IEEE Trans. Signal Process.*, vol. 68, pp. 2841–2856, 2020.
[15] C. H. Wu and H. H. Chang, “Superpixel-based image noise variance estimation with local statistical assessment,” *EURASIP Journal on Image and Video Processing*, vol. 2015, no. 1, p. 38, 2015.
[16] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785
[17] D. Pasquel, S. Roux, J. Richetti, D. Cammarano, B. Tisseyre, and J. Taylor, “A review of methods to evaluate crop model performance at multiple and changing spatial scales: Precision Agriculture, 2022.
[18] C. Yang, G. Cheung, and W. Hu, “Signed graph metric learning via Gershgorin disc perfect alignment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
[19] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain,” CMU, Tech. Rep., 1994.
[20] S. E. Schaeffer, “Graph clustering,” *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007.
[21] H. Rue and L. Held, *Gaussian Markov random fields: theory and applications*. CRC press, 2005.
[22] F. Cavazis and C. Karande, “A note on the problem of reporting maximal cliques,” *Theoretical computer science*, vol. 407, no. 1-3, pp. 564–568, 2008.
[23] O. L. Mangasarian, “Pseudo-convex functions,” in *Stochastic optimization models in finance*. Elsevier, 1975, pp. 23–32.
[24] A. V. Knyazev, “Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method,” *SIAM journal on scientific computing*, vol. 23, no. 2, pp. 517–541, 2001.
[25] A. Altman, L. Tolocsi, O. Sander, and T. Lengauer, “Permutation importance: a corrected feature importance measure,” *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.