Analysis of Sports Statistics via Graph-Signal Smoothness Prior

Haitian Zheng*, Gene Cheung† and Lu Fang*
* University of Science and Technology China, Hefei, China
E-mail: {zhenght, fanglu}@ustc.edu.cn Tel/Fax: +86-511-63607430
† National Institute of Informatics, Tokyo, Japan
E-mail: cheung@nii.ac.jp Tel/Fax: +81-3-4212-2567

***Abstract*—Since teams in a sporting league compete head-to-head according to a structured schedule, it is natural to interpret statistics emanating from competitions as signals on a graph modeling similarities among competing entities. In this paper, we analyse available sports statistics to predict game outcomes from a graph signal processing (GSP) perspective: GSP tools are used to remove (denoise) unwanted variability to reveal underlying predictable trends, and to interpolate missing data—predicted game outcomes in terms of point differential. First, we construct a graph for the desired graph-signal (point differential for every team pair): for an N-team league, we construct N subgraphs $\mathcal{G}_j$, each containing $N-1$ nodes representing teams competing against opponent j. We next assign weight to each intra-subgraph edge based on similarity in observed statistics (e.g., total points scored, assists, etc) of the two connecting nodes (teams). We then connect nodes in different subgraphs representing the same teams, where the weight of an inter-subgraph edge connecting nodes in subgraphs $\mathcal{G}_k$ and $\mathcal{G}_l$ now reflects the similarity between opponents k and l. Finally, assuming a graph-signal smoothness prior, we compute the desired graph-signal on the constructed graph via an alternating convex programming procedure. Experimental results show that our graph-based scheme achieves better prediction than a competing k-nearest neighbor (kNN) scheme.**

I. Introduction

Professional sports is a multi-billion dollar business. The financial value of a professional team is explicitly tied to its success on the playing field. To maximize a team's success, in the past decade there is a drive towards data-driven analysis of sports statistics to predict the performance of professional teams and players[1]. In this paper, we adopt a signal processing approach to analyse publicly available head-to-head sports statistics to predict game outcome of any matchup between two competing entities in a sporting league.

In particular, we pose the data analysis problem from a *graph signal processing* (GSP) perspective [1]. GSP is the study of signals that exist naturally on structured data kernels described by graphs; examples of graph-signals include temperature collected by distributed sensors in a wireless network, and messages posed by a group of friends connected in an online social network. In a sporting league, teams compete according to a structured schedule, and thus it is natural to interpret statistics emanating from scheduled competitions as signals on graphs modeling similarities among competing entities. Observed statistics contain variability due to non-stationary game-to-game conditions, such as players' physical states on a given day, in-game reactionary decisions by players and referees, etc. We model this variability as random noise, and GSP supplies tools [2–7] to *denoise* the signal to extract the predictable components from the observations. Observed statistics are also incomplete; an N-team league typically has far fewer than $N(N-1)$ sets of reliable statistics due to scheduling constraints. GSP provides tools [8–10] to *interpolate* missing samples in a desired graph-signal—point differential for every pair of teams in our case. Interpolated samples are thus our predicted game outcomes in future matchups in terms of point differential.

Specifically, we design a graph signal analysis framework to study sports statistics as follows. First, we construct a graph *a priori* for the desired graph-signal (point differentials): for an N-team league, we construct N subgraphs $\mathcal{G}_j$, each containing $N-1$ nodes representing teams competing against common opponent j. We next assign a weight to each *intra-subgraph edge* based on similarity in observed statistics (*e.g.*, total points scored, assists, etc) of the two connecting nodes (teams); more similar team pairs will have larger edge weights. We then connect nodes in different subgraphs representing the same teams, where the weight of an *inter-subgraph edge* connecting nodes in subgraphs $\mathcal{G}_k$ and $\mathcal{G}_l$ now reflects the similarity between opponents k and l. Finally, assuming a graph-signal smoothness prior, we compute the desired graph-signal on the constructed graph via an alternating convex programming procedure. Experimental results show that our graph-based scheme achieves better prediction than a competing k-nearest neighbor (kNN) scheme.

The outline of the paper is as follows. We first overview related work in Section II. We then describe the graph-signal smoothness prior in Section III. We formulate our graph-based data analysis problem in Section IV, and present our optimization algorithm in Section V. Finally, experimental results and conclusion are presented in Section VI and VII, respectively.

II. Related Work

The majority of available literature on sport event prediction of head-to-head matches employ a statistical learning

[1] As a concrete illustration, bestseller "Moneyball" chronicled Major League baseball Oakland Aces general manager Billy Beane's reliance on analytics to field a competitive team in 2002 despite their limited payroll.

approach: generic learning models like Gaussian Markov Random Field (GMRF), Gaussian Mixture Models (GMM) etc are first chosen *a priori*, large amount of data are collected for statistical analysis to derive parameters of chosen models best fitted to the available data, and finally, desired output like players' rank, match winning probabilities, match outcomes are predicted based on developed models.

Specifically, a representative work [11] combines Bayes inference and rule-based reasoning for soccer matches prediction. Each game is represented as a series of *tides* (ball possession periods). Probabilities of some events, like possible player substitute or change of formation, are calculated on the basis of previous data in each tide and are used to fire certain rules to determine the decisions for the next tide. Another common approach is the use of artificial neural networks [12, 13] to train the multi-layer perceptron based on a number of statistic data. All of these methods suffer from unavoidable noise corruption in the observed low-level statistics without explicitly first pre-processing them [14].

Other approaches have also been used in sports games, other than the prediction of match outcomes. In the basketball domain, Polese *et al.* [15] present a pre-game and post-game analysis system based on the data mining and fuzzy evaluation, to provide a performance evaluation of the players and predict their performances in the upcoming matches. Such a system also helps the basketball coaches in making tactical/technical decisions during matches. Naïve Bayes classifiers are proposed to predict the Cy Young Award winners in American baseball in [16]. The system has the winners successfully predicted over 80% of the prizewinners in the period from 1967 to 2006.

A notable work is [17], where factor graphs are used to model interaction among players and Bayesian analysis is used to determine players' rank. The rank is then used to assess players and to predict particular match outcomes in terms of wins / loses.

In summary, the majority of the state-of-the-art studies on head-to-head matches train generic statistical models based on large quantity of available data for prediction. In contrast, in our case the potential dearth of data (*e.g.*, two NBA teams may play against one another only once a season) means statistically significant conclusions may not be possible through traditional statistical analysis. Thus we opt instead a deterministic approach, where by leveraging on the domain knowledge of our specific problem, we first construct a suitable graph structure *a priori* that reflects team similarities. We then formulate an objective function to jointly denoise a single large deterministic graph-signal and interpolate missing samples using a graph-signal smoothness prior. In doing so, our proposed method extracts the predictable components from limited noise-corrupted observations thanks to domain-specific knowledge and smoothness assumption, improving the accuracy of match outcome prediction.

III. Graph-Signal Smoothness Prior

As in any inverse problem, a signal prior for the desired signal $\mathbf{x}$ is needed for regularization, so that the optimization problem to identify the desired signal is well posed. As done in [3–7, 10], in this paper we also employ a *graph-signal smoothness prior*: a signal $\mathbf{x}$ is more probable if $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is small, *i.e.*,

$$Pr(\mathbf{x}) = \exp\left\{-\frac{\mathbf{x}^T\mathbf{L}\mathbf{x}}{\sigma_x^2}\right\} \tag{1}$$

where $\mathbf{L}$ is the *graph Laplacian* for signal $\mathbf{x}$ and σ_x^2 is a model parameter. One interpretation is that (1) is a GMRF generative model for signal $\mathbf{x}$, where $\mathbf{L}$ is the precision matrix [18]. We instead take a graph spectral interpretation, leading to a notion of smoothness for a deterministic signal on a defined graph.

Specifically, we first construct a graph $\mathcal{G}$ where the nodes in the graph correspond to signal samples in $\mathbf{x}$. Edges connect signal samples in $\mathcal{G}$, whose weights $w_{i,j}$ reflect the degree of similarity between connecting nodes i and j. Having defined edge weights, one can define the *adjacency matrix* $\mathbf{W}$ where the (i,j)-th entry is $W_{i,j} = w_{i,j}$. The *degree matrix* $\mathbf{D}$ is a diagonal matrix where the i-th diagonal entry is $D_{i,i} = \sum_j W_{i,j}$. The *combinatorial graph Laplacian* $\mathbf{L}$ is then defined as the difference between the degree matrix $\mathbf{D}$ and the adjacency matrix $\mathbf{W}$ [1]:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2}$$

It can be shown [1] that the Laplacian regularizer $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is a measure of variation in the signal $\mathbf{x}$ modulated by weights $w_{i,j}$:

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{i,j} w_{i,j}(x_i - x_j)^2 \tag{3}$$

Thus $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is small if the squared signal variation $(x_i - x_j)^2$ is small *or* the modulating weight $w_{i,j}$ is small.

Given $\mathbf{L}$ is positive semi-definite, one can perform eigen-decomposition on $\mathbf{L}$ to obtain non-negative eigen-values λ_k and eigen-vectors ϕ_k. We can then express $\mathbf{x}^T\mathbf{L}\mathbf{x}$ alternatively as:

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_k \lambda_k \alpha_k^2 \tag{4}$$

where eigen-value λ_k can be interpreted as the k-th graph frequency, and $\alpha_k = \phi_k^T\mathbf{x}$ is the coefficient for the k-th graph frequency. In this interpretation, a small $\mathbf{x}^T\mathbf{L}\mathbf{x}$ means that the energy of the signal $\mathbf{x}$ is concentrated in the low graph frequencies.

IV. Problem Formulation

A. Analysis Overview

We now formulate our graph-signal analysis (joint denoising / interpolation) problem. We assume that the aggregate *low-level statistics* of all teams and point differentials of a sufficient number of head-to-head competitions in a targeted sporting league are available for analysis. By head-to-head competition, we mean that a team i is competing *directly* against a team j. Examples include individual sports such as tennis and ping-pong and team sports such as basketball and volleyball. For team sports, we gather statistics of the entire team rather than individuals in the team. Analysis of open competitions like golf is considered out-of-scope.

By low-level statistics, we mean a quantifiable number tabulated using an accepted, sport-dependent statistical definition that reflects a team's offensive / defensive performance during the course of the season thus far. Examples of statistics in basketball are total points scored, assists, three pointers made, free throws made, etc. Examples of defensive statistics are steals, blocks, defensive rebounds, etc.

Our objective is to analyse available data, so that the *expected point differential* of team i against team j can be estimated for each pair of teams (i, j) in the league. A positive head-to-head point differential would mean that team i is expected to beat opponent j (and vice versa). Beyond a binary game outcome predictor (win / lose), our prediction in point differentials can be compared against point spread set by gambling establishments for spread betting[2]. Note that previous works like [17] consider only coarse-grained predictions of wins / loses, rather than fine-grained predictions of point differentials.

In the following sections, we describe our proposed methodology to construct a graph and compute edge weights. Finally, we formulate an objective function for optimization.

B. Graph Construction

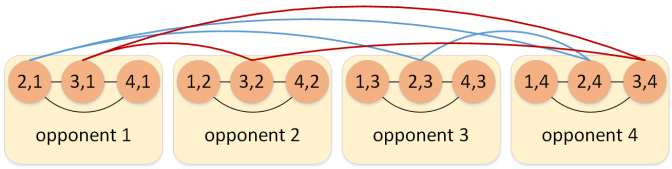


Fig. 1. Example of four subgraphs connected to a large graph for four teams. Intra-subgraph edges are black, and inter-subgraph edges are blue or red for team 2 and 3 respectively.

Towards meaningful GSP computation, we construct an appropriate graph *a priori* for the intended signal—expected point differentials in head-to-head competitions in our case. We first construct a suitable subgraph[3] $\mathcal{G}_k$ for point differentials of teams against a given opponent k. We construct one node for each team i, $i \neq k$. The desired signal $x_{i|k}$ at each node i in $\mathcal{G}_k$ is the point differential of team i against team k. Nodes i and j in $\mathcal{G}_k$ are connected by an *intra-subgraph* edge s with weight w_s that reflects the similarity between teams i and j (to be discussed in details).

There are N subgraphs $\mathcal{G}_k$ each with $N-1$ nodes—thus a total of $N(N-1)$ nodes—and we can connect them together to compose a large graph $\mathcal{G}$. Specifically, since two nodes $x_{i|k}$ and $x_{i|l}$ represent the same team i in two different subgraphs $\mathcal{G}_k$ and $\mathcal{G}_l$, one would expect point differentials $x_{i|k}$ and $x_{i|l}$ to be similar, *if* opponents k and l are similar. We can thus draw an *inter-subgraph* edge between these two nodes in two subgraphs $\mathcal{G}_k$ and $\mathcal{G}_l$, where the edge weight would reflect the degree of similarity between opponents k and l. See Fig. 1 for an illustration, where four teams competing head-to-head are represented by four subgraphs. Black edges are intra-subgraph edges connecting nodes in the same subgraphs—different teams competing against the same opponents. Red and blue edges are inter-subgraph edges connecting nodes across subgraphs; blue (red) edges connects nodes representing the same team 2 (3) competing against different opponents.

[2]In *Spread betting*, a point spread Z is established prior to a match. One wins / loses a bet if the stronger team wins by Z points or more.

[3]The idea to construct N subgraphs $\mathcal{G}_k$ each containing $N-1$ nodes—as opposed to a simpler N-node graph—is to enable graph-signal description of unique team matchups (i, k). A team i may be strong in general but can perform poorly against team k due to matchup problems. Having N subgraphs allows us to express this oddity in one graph-signal of length $N(N-1)$.

C. Edge Weight

The constructed graph $\mathcal{G}$ is effective only if the edge weights are properly computed to reflect the structure of the desired signal. For an intra-subgraph edge s in subgraph $\mathcal{G}_l$, we can write weight $w_{s|l}$ of edge s connecting nodes $e_1(s)$ and $e_2(s)$ using an exponential kernel [5, 6]:

$$w_{s|l} = \exp\left\{-\frac{|\text{dist}_l(e_1(s), e_2(s))|^2}{\sigma_w^2}\right\} \quad (5)$$

where $\text{dist}_l(i, j)$ computes the distance between two samples i and j using *feature functions* associated with the samples, and σ_w is a parameter chosen so that weight $w_{s|l}$ evaluates to be near 1 if distance $\text{dist}_l(i, j)$ is small, and near 0 otherwise. As discussed in [5], feature functions that determine the sample distance metric can be freely chosen by users; in this paper, we use low-level statistics $\mathbf{h}_i$ (a vector of sport-specific quantities as described earlier) and a *pre-filtered* point differential estimate $\hat{\mathbf{x}}$ of the two teams as our feature functions. Using the signal estimate $\hat{\mathbf{x}}$ to determine the weighting parameters (5) used to filter signal $\mathbf{x}$ is similar to the range filter in *bilateral filtering* [19]. We can now write $\text{dist}_l(i, j)$ as:

$$\text{dist}_l(i, j) = \left|\hat{x}_{i|l} - \hat{x}_{j|l}\right|^2 + \gamma(\mathbf{h}_i - \mathbf{h}_j)^T \Psi_l (\mathbf{h}_i - \mathbf{h}_j) \quad (6)$$

where $\hat{x}_{i|l}$ is a point differential estimate of team i against opponent l, and Ψ_l is a positive semi-definite matrix, *i.e.* $\mathbf{h}^T \Psi_l \mathbf{h} \geq 0, \forall\, \mathbf{h}$. For simplicity, we will only consider Ψ to be a diagonal matrix.

For an inter-subgraph edge, because the connecting nodes represent the same team, statistics of the opponents k and l corresponding to the subgraphs $\mathcal{G}_k$ and $\mathcal{G}_l$ are used to compute edge weight (5) instead.

D. Objective Function

1) Fidelity Term: Given the constructed graph $\mathcal{G}$, we can now formalize an objective function as follows. We denote by vector $\mathbf{x}$, $\mathbf{x} \in \mathbb{R}^{N(N-1)}$, the desired graph-signal on graph $\mathcal{G}$, where x_i, $i \in \{1, \ldots, N(N-1)\}$, is the expected point differential of team s against opponent t, *i.e.* $x_{s|t}$, where:

$$t = \left\lfloor \frac{i-1}{N-1} \right\rfloor + 1 \quad (7)$$

$$s = \begin{cases} i - (t-1)(N-1) & \text{if } i - (t-1)(N-1) < t \\ i - (t-1)(N-1) + 1 & \text{o.w.} \end{cases}$$

For example, given there are $N = 4$ teams, x_5 in Fig. 1 is the point differential of team 3 against opponent 2.

We denote by vector $\mathbf{y}$, $\mathbf{y} \in \mathbb{R}^M$, the M available observed game differentials. y_i can be the game outcome of any two opposing teams; the same two teams can also play against each other multiple times. Thus we define a corresponding binary matrix $\mathbf{D}$ of dimension $M \times N(N-1)$ that matches entries in signal $\mathbf{x}$ to observations $\mathbf{y}$; each row in $\mathbf{D}$ contains a single 1 with all other entries being 0. Continuing with our previous example, a k-th row in $\mathbf{D}$ with entry $D_{k,5} = 1$ will map x_5 (point differential of team 3 against opponent 2 according to (7)) to observed y_k. Our fidelity term compares reconstructed signal $\mathbf{x}$ with observed $\mathbf{y}$ via $\mathbf{D}$:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \tag{8}$$

By using a L_2-norm for the fidelity term (8), we are minimizing the mean squared error (MSE) with respect to observations $\mathbf{y}$. From the perspective of a *Maximum A Posteriori* (MAP) formulation, (8) also means that the likelihood $Pr(\mathbf{y}|\mathbf{x})$ of observation $\mathbf{y}$ given graph-signal $\mathbf{x}$ is a Gaussian noise term with mean $\mathbf{D}\mathbf{x}$.

2) Rewriting the Graph-Signal Smoothness Prior: Instead of using $\mathbf{x}^T\mathbf{L}\mathbf{x}$ directly as our graph-signal smoothness prior as discussed in Section III, because we pre-define our graph structure using available domain knowledge, we can first write the graph Laplacian $\mathbf{L}$ as a weighted sum of "mini-Laplacian" $\mathbf{L}_s$:

$$\mathbf{L} = \sum_s w_s \mathbf{L}_s \tag{9}$$

where each mini-Laplacian $\mathbf{L}_s$ accounts only for one edge s in the constructed graph structure:

$$\mathbf{L}_s(i,j) = \begin{cases} -1 & \text{if } i \neq j \text{ and } i,j \in \{e_1(s), e_2(s)\} \\ 1 & \text{if } i = j \text{ and } i \in \{e_1(s), e_2(s)\} \\ 0 & \text{o.w.} \end{cases} \tag{10}$$

As an example, the mini-Laplacian $\mathbf{L}_1$ that accounts for the first edge $e_1 = (1,2)$ that connects nodes 1 and 2 is:

$$\mathbf{L}_1 = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \tag{11}$$

3) MAP Formulation: Similar to other inverse problems [3–7, 10], our objective function is also a sum of fidelity term and a graph-signal smoothness term—accounting for both likelihood and prior in a MAP formulation. However, because we defined our graph structure *a priori*, instead of optimizing directly the graph Laplacian $\mathbf{L}$ that describes the (partial) observations as done in [20], we optimize only matrices Ψ_l that define weights w_s in (5), which in turn compose $\mathbf{L}$ in (9):

$$\min_{\Psi_l, \mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \mu\, \mathbf{x}^T\mathbf{L}\mathbf{x} \tag{12}$$

where μ is a parameter to trade off the fidelity term with the signal prior, as commonly used in conventional inverse problems [3–7, 10].

E. Discussion

Our graph-signal formulation in (12) means that *the graph-signal in question can be interpreted as samples on a smooth continuous manifold [5, 6], and an edge weight between two nodes is proportional to the exponential of the minus distance between the samples*. Thus nodes connected by large weights (similar low-level statistics according to (5)) are samples of close proximity on the same smooth manifold, and therefore should reconstruct to similar sample values (similar point differentials).

However, unlike [5, 6], the optimization variables are the graph-signal $\mathbf{x}$ and the weights that specify the Laplacian $\mathbf{L}$ of our graph structure. The reason is because there is an intimate relationship between the graph-signal $\mathbf{x}$ and the graph $\mathcal{G}$: the graph must properly reflect the structure of the signal for the graph-signal smoothness prior to hold, similarly argued in [20]. From a manifold perspective, it means we are reconstructing samples on the manifold *and* defining the metric used to compute distance between samples, so that the underlying manifold is smooth in the high-dimensional space.

V. Optimization Algorithm

We can minimize (12) by alternately optimizing one set of variables while keeping the other set fixed. When weight parameters Ψ_l are fixed, the optimal $\mathbf{x}^*$ to (12) has a closed-form solution:

$$\mathbf{x}^* = \left(\mathbf{D}^T\mathbf{D} - \mu\mathbf{L}\right)^{-1}\mathbf{y} \tag{13}$$

Note that we use a pre-filtered version $\hat{\mathbf{x}}$ of the desired signal to compute the distance metric (6) used to define edge weights $w_{s|l}$ via (5). For the first iteration, $\hat{\mathbf{x}}$ is a linear-regressed solution; for subsequent iterations, $\hat{\mathbf{x}}$ is the solution computed in the previous iteration.

When $\mathbf{x}$ is fixed, the first term in (12) is not affected by Ψ_l, and we can compute the optimal Ψ_l by minimizing the second term only. We know from (3) that $\mathbf{x}^T\mathbf{L}\mathbf{x}$ can be written as a weighted sum of neighboring node differences. The objective becomes:

$$\min_{\{\Psi_l\}} \sum_l \sum_{s \in \mathcal{G}_l} w_{s|l} \left(x(e_1(s)) - x(e_2(s))\right)^2 \tag{14}$$

We see in (14) that each Ψ_l can be optimized independently. Assuming Ψ_l is a diagonal matrix, and adding the following constraints to avoid trivial solutions,

$$\mathbf{Tr}(\Psi_i) = \Delta, \forall\, i \tag{15}$$

$$\Psi_i \succeq 0, \forall\, i \tag{16}$$

then the optimization for Ψ_i is a convex semidefinite programming problem, which can be solved using numerical optimization tools. To speed up the optimization process, we remove an edge s from $\mathcal{G}_l$ if its initial weight w_s is smaller than a pre-defined threshold, resulting in a sparse subgraph.

VI. Experimental Results

To evaluate the effectiveness of our graph-based data analysis framework, we predict the game outcomes for pairs of teams who have not played against each other yet. The metric we use for evaluation is the squared prediction error between predicted point differential and actual point differential.

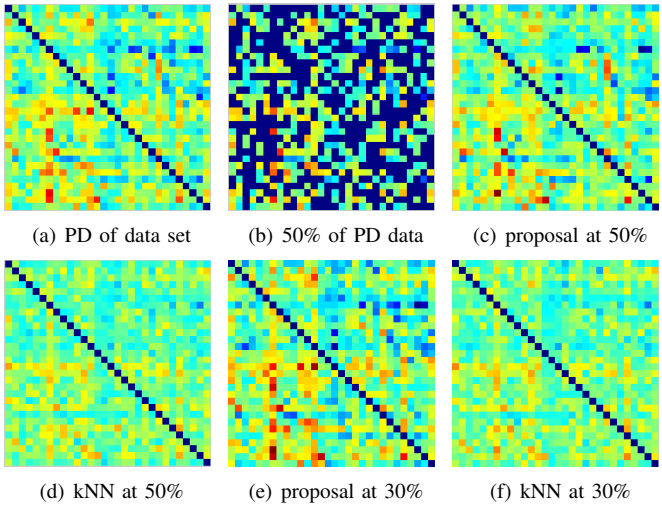
A. Data Collection

We collected publicly available NBA game statistics of the 2013-2014 season from the known website `basketball-reference`[4]. The total number of games played is 1230, and there are 30 teams. For a pair of teams (i, j), low-level statistics and point differentials are computed as averages over games they have played against each other. Offensive statistics include: Field Goals, 3-Point Field Goals, Free Throws, Assists, etc. Defensive statistics include: Steals, Blocks, Turnovers of the opposing player, etc.

B. Prediction of Point Differentials

We randomly mask 50% of the point differentials as missing samples, then apply our model to predict missing differentials given the available ones.

As a competing scheme, a k-nearest neighbors algorithm (kNN) is also used to predict missing point differentials. Similarity between two different entries is defined by the distance of head-to-head statistics. Point differential prediction of missing entry is further calculated by the averaging of point differential of its most similar K entries.



(a) PD of data set (b) 50% of PD data (c) proposal at 50%

(d) kNN at 50% (e) proposal at 30% (f) kNN at 30%

Fig. 2. Missing entries prediction on NBA data set. (a) shows the ground truth game differentials for pairs of teams. (b) shows game differentials with 50% of entries removed. (c) and (d) show the predicted game differentials for our proposal and kNN when 50% of data are missing. (e) and (f) show the predicted game differentials for our proposal and kNN when 30% of data are missing.

We compare the performance of kNN with our approach. The best average squared error of our prediction is achieved when $\mu = 0.2$ and $\Delta = 6$. The best accuracy of kNN is 242.7, achieved when $K = 10$. Plotting indices of teams against indices of teams, Fig. 2(a) shows the point differentials of the ground truth data set (blue / red means negative / positive values, and stronger color means larger magnitude). Fig. 2(b) shows the same ground truth data set with 50% of the entries removed. Estimations of point differentials using our model and kNN are showed in Fig. 2(c) and 2(d) respectively. By comparing the colors between our prediction in Fig. 2(c) and original data in Fig. 2(a), we observe that our model is better at predicting point differentials in the missing data than kNN.

We also randomly mask 30% of the point differentials as missing samples, and then apply our method and kNN for prediction. The best average square error of our prediction is 99.9, while the best average square error of kNN is 136.1. Estimations of point differentials using our model and kNN are showed in Fig. 2(e) and 2(f) respectively. We observe that while both schemes have better predictions when the amount of missing samples is smaller, our proposal recovers distinct patterns in the signal (*e.g.* the red portion in the lower-left quadrant) much better than kNN.

C. Denoising

We next evaluate the effectiveness of our framework on denoising of noise-corrupted data. We corrupt game outcome with additive Gaussian noise and apply our framework on the noisy data. The standard deviation of Gaussian noise is set to half of that of actual game differentials. Here we use identity matrix for $\mathbf{D}$, and regard $\mathbf{x}$ from (12) as the denoised result. Noise-corrupted data, and our denoised data are shown in Fig. 3.

We can observe from Fig. 3 that our denoised result resembles the original data remarkably. The structure of the data (short transitions in color) is recovered to a very large extent, demonstrating the effectiveness of our framework in data denoising. Numerically, our scheme can reduce noise to 0.270 in MSE, while kNN can achieve 0.456.

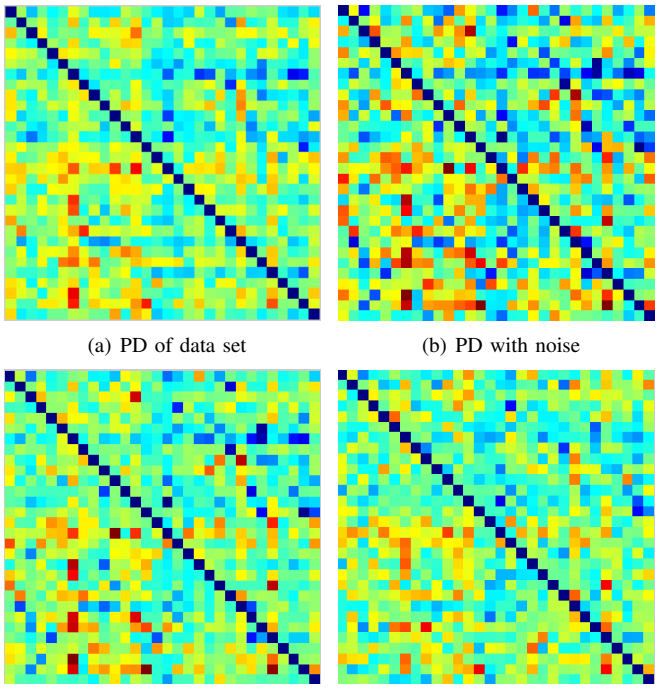
VII. Conclusion

We propose to analyse head-to-head sports statistics from a graph signal processing (GSP) perspective. GSP tools are used for both denoising variability in the observations, and for interpolating missing signal samples—pairs of teams who have not competed together before. In particular, we first construct an appropriate graph for our desired signal (point differentials for every team pair). Assuming a graph-signal smoothness prior, we compute the desired signal on the constructed graph via an alternating convex programming procedure. Experimental results show that our graph-based scheme achieves better predictions than a competing k-nearest neighbor scheme.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in *IEEE Signal Processing Magazine*, May 2013, vol. 30, no.3, pp. 83–98.

[4] http://www.basketball-reference.com



(a) PD of data set (b) PD with noise

(c) our denoise result (d) KNN denoise result

Fig. 3. Denoising on NBA data set. (a) shows the ground truth game differentials for pairs of teams. (b) shows the noise corrupted game differentials. (c) shows the denoised game differentials for our proposal. (d) shows the denoised game differentials predicted for kNN.

[2] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *IEEE International Workshop on Multimedia Signal Processing*, Pula, Italy, October 2013.

[3] W. Hu, G. Cheung, X. Li, and O. Au, "Graph-based joint denoising and super-resolution of generalized piecewise smooth images," in *IEEE International Conference on Image Processing*, Paris, France, October 2014.

[4] P. Wan, G. Cheung, D. Florencio, C. Zhang, and O. Au, "Image bit-depth enhancement via maximum-a-posteriori estimation of graph AC component," in *IEEE International Conference on Image Processing*, Paris, France, October 2014.

[5] J. Pang, G. Cheung, W. Hu, and O. C. Au, "Redefining self-similarity in natural images for denoising using graph signal gradient," in *APSIPA ASC*, Siem Reap, Cambodia, December 2014.

[6] J. Pang, G. Cheung, A. Ortega, and O. C. Au, "Optimal graph laplacian regularization for natural image denoising," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2014.

[7] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Inter-block soft decoding of JPEG images with sparsity and graph-signal smoothness priors," in *IEEE International Conference on Image Processing*, Quebec City, Canada, September 2015.

[8] Y. Mao, G. Cheung, A. Ortega, and Y. Ji, "Expansion hole filling in depth-image-based rendering using graph-based interpolation," in *IEEE International Conference on Acousitics, Speech and Signal Processing*, Vancouver, Canada, May 2013.

[9] Y. Mao, G. Cheung, and Y. Ji, "Graph-based interpolation for DIBR-synthesized images with nonlocal means," in *Symposium on Graph Signal Processing in IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, TX, December 2013.

[10] Y. Mao, G. Cheung, and Y. Ji, "Image interpolation during DIBR view synthesis using graph Fourier transform," in *3DTV-Conference*, Budapest, Hungary, July 2014.

[11] B. Min, J. Kim, C. Choe, H. Eom, and R. McKay, "A compound framework for sports results prediction: A football case study," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 551–562, 2008.

[12] M. Baulch, "Using machine learning to predict the results of sporting matches," *Department of Computer Science and Electrical Engineering, University of Queensland*, 2001.

[13] A. McCabe and J. Trevathan, "Artificial intelligence in sports prediction," in *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*. IEEE, 2008, pp. 1194–1197.

[14] Z. Shi, S. Moorthy, and A. Zimmermann, "Predicting NCAAB match outcomes using ML techniques–some results and lessons learned," in *ECML / PKDD 2013 Workshop on Machine Learning and Data Mining for Sports Analytics*, Prague, Czech Republic, September 2013.

[15] G. Polese, M. Troiano, and G. Tortora, "A data mining based system supporting tactical decisions," in *Proceedings of the 14th international conference on Software engineering and knowledge engineering*. ACM, 2002, pp. 681–684.

[16] L. Smith, B. Lipscomb, and A. Simkins, "Data mining in sports: predicting cy young award winners," *Journal of Computing Sciences in Colleges*, vol. 22, no. 4, pp. 115–121, 2007.

[17] R. Herbrich, T. Minka, and T. Graepel, "TrueSkill: A Bayesian skill rating system," in *Twentieth Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, December 2006.

[18] A. Gadde and A. Ortega, "A probabilistic interpretation of sampling theory of graph signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2015.

[19] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE International Conference on Computer Vision*, Bombay, India, 1998.

[20] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Laplacian matrix learning for smooth graph signal representation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2015.