Handling Missing Data with Graph Representation Learning

You et.al. NeurIPS 2020

Presented by: Rucha Bhalchandra Joshi

February 8, 2021

The Missing Data Problem

Issues with Learning

In computational biology, clinical studies, survey research, finance, and economics.

Two approaches

Feature Imputation

Missing feature values are estimated based on observed values

Label Prediction

Downstream labels are learned directly from incomplete data

Existing Works for Missing Data Problem

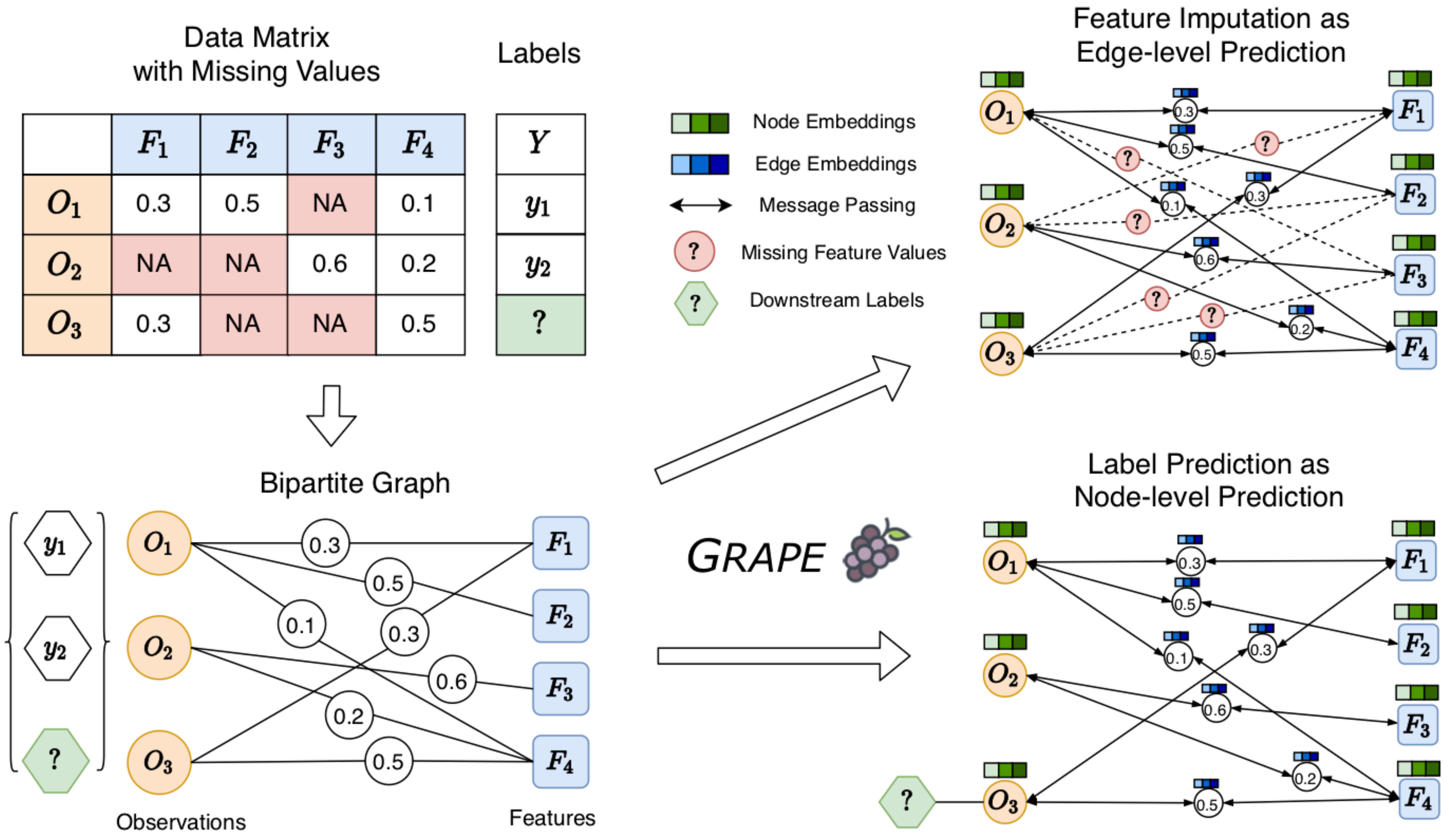
Statistical Methods

- Strong assumptions about data distributions
- Lack flexibility for handling mixed datatypes
- Fail to generalize to unseen data without retraining

Deep Networks

- Fail to make full use of feature values
- Biased assumptions about data by special value initialization

The GRAPE Framework



Overview of GRAPE framework

The GRAPE Framework: GNN

Algorithm 1 GRAPE forward computation

Input: Graph $\mathcal{G} = (\mathcal{V}; \mathcal{E})$; Number of layers L; Edge dropout rate r_{drop}; Weight matrices $\mathbf{P}^{(l)}$ for *message passing*, $\mathbf{Q}^{(l)}$ for *node updating*, and $\mathbf{W}^{(l)}$ for *edge updating*; non-linearity σ; aggregation functions AGG_l; neighborhood function $\mathcal{N} : v \times \mathcal{E} \rightarrow 2^{\mathcal{V}}$

Output: Node embeddings $\mathbf{h}_v$ corresponding to each $v \in \mathcal{V}$

$\mathbf{h}_v^{(0)} \leftarrow \boxed{\text{INIT}(v), \forall v \in \mathcal{V}}$ Augmented node features → better representation power

$\mathbf{e}_{uv}^{(0)} \leftarrow \mathbf{e}_{uv}, \forall \mathbf{e}_{uv} \in \mathcal{E}$

$\mathcal{E}_{drop} \leftarrow \boxed{\text{DROPEDGE}(\mathcal{E}, r_{drop})}$ Edge dropout → improved model generalization

for $l \in \{1, \ldots, L\}$

  for $v \in \mathcal{V}$

    $\mathbf{n}_v^{(l)} = \text{AGG}_l\Big(\sigma(\mathbf{P}^{(l)} \cdot \text{CONCAT}(\mathbf{h}_v^{(l-1)}, \mathbf{e}_{uv}^{(l-1)}) \mid \forall u \in \mathcal{N}(v, \mathcal{E}_{drop}))\Big)$

    $\mathbf{h}_v^{(l)} = \sigma(\mathbf{Q}^{(l)} \cdot \text{CONCAT}(\mathbf{h}_v^{(l-1)}, \mathbf{n}_v^{(l)}))$

  for $(u, v) \in \mathcal{E}_{drop}$

    $\boxed{\mathbf{e}_{uv}^{(l)} = \sigma(\mathbf{W}^{(l)} \cdot \text{CONCAT}(\mathbf{e}_{uv}^{(l-1)}, \mathbf{h}_u^{(l)}, \mathbf{h}_v^{(l)}))}$ Edge embedding → utilize edge features

  $z_v \leftarrow h_v^L$

The GRAPE Framework: GNN

Edge-level predictions at L-th layer

$$\hat{\mathbf{D}}_{uv} = \mathbf{O}_{edge}\left(\mathrm{CONCAT}(\mathbf{h}_u^{(L)}, \mathbf{h}_v^{(L)})\right)$$

Node-level predictions at L-th layer, using imputed dataset

$$\hat{\mathbf{Y}}_u = \mathbf{O}_{node}\left(\hat{\mathbf{D}}_u\right)$$

$\mathbf{O}_{edge}$ and $\mathbf{O}_{node}$ are feedforward neural networks

The GRAPE Framework

Important features of this framework:

1. Connections between different features and between different observations

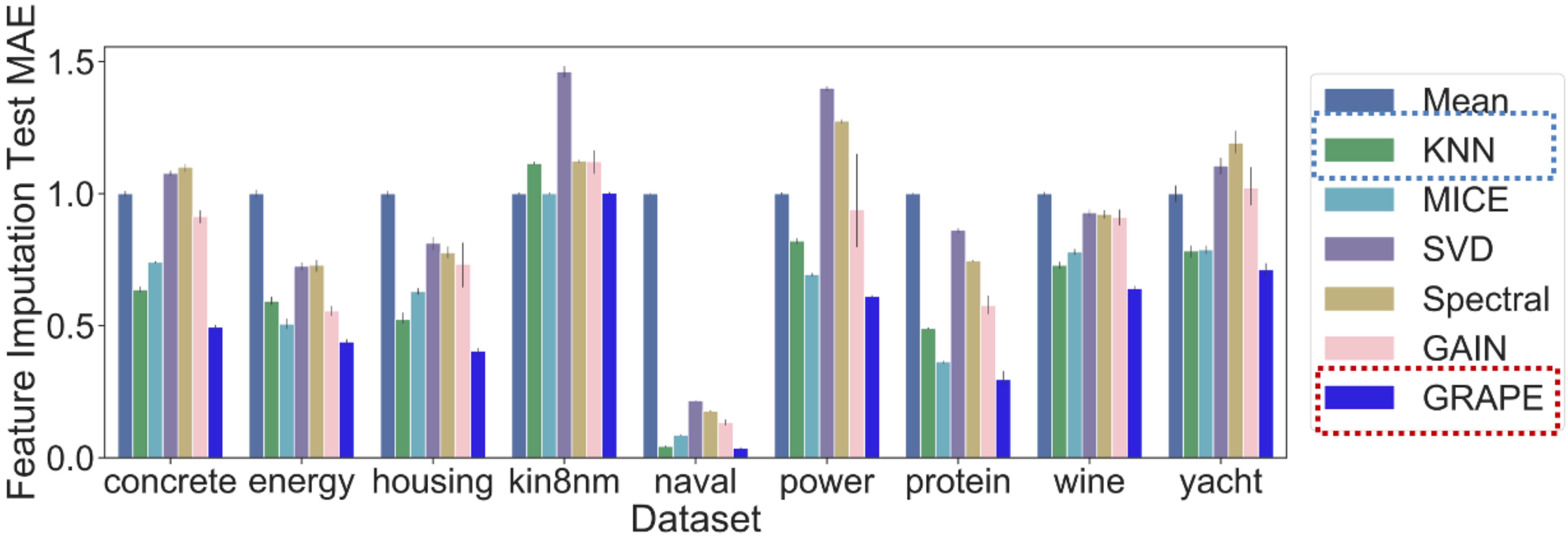
2. Propagate and borrow information from other features/observations in graph localized way

3. End-to-end feature imputation and label prediction, leads to strong performance improvement

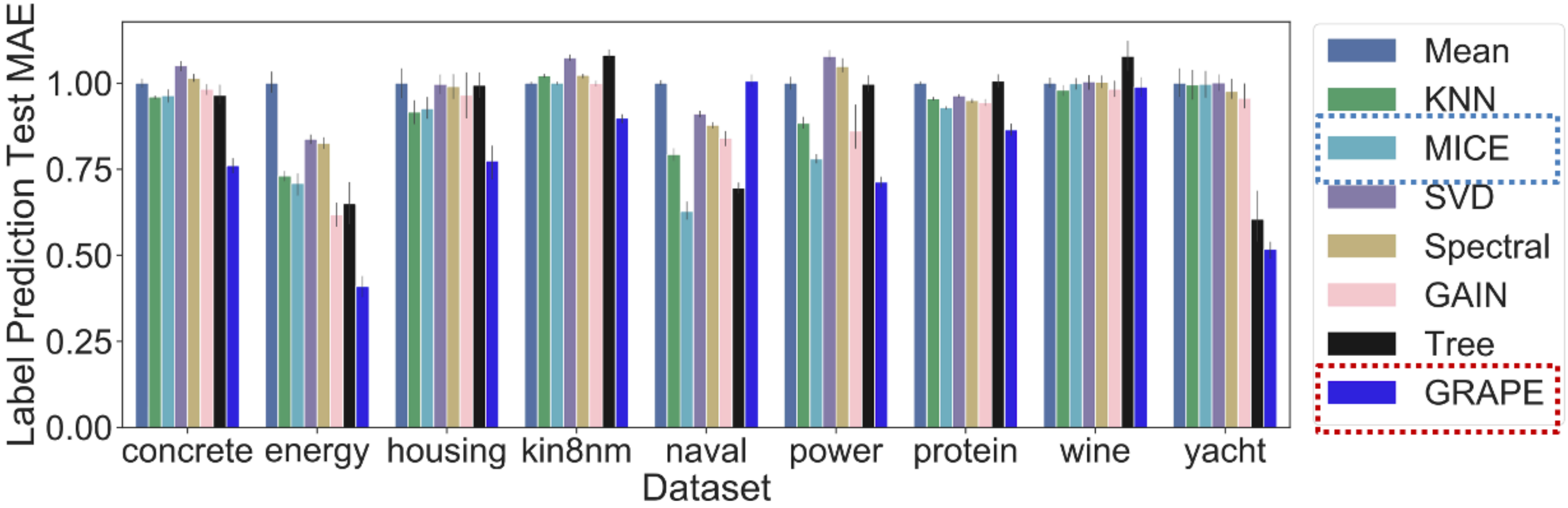
Datasets

- **Mean imputation (Mean):** Imputes the missing Dij with the mean of all the samples with observed values in dimension j.
- **K-nearest neighbors (KNN):** Imputes the missing value Dij using the KNNs that have observed values in dimension j with weights based on the Euclidean distance to sample i.
- **Multivariate imputation by chained equations (MICE):** Runs multiple regression where each missing value is modeled conditioned on the observed non-missing values
- **Iterative SVD (SVD):** Imputes missing values based on matrix completion with iterative low-rank SVD decomposition
- **Spectral regularization algorithm (Spectral):** Matrix completion model, imputes missing values using SVD
- **Generative Adversarial Imputations Nets (GAIN):** state-of-the-art deep imputation model with generative adversarial training
- **Decision tree (Tree):** statistical method that can handle missing values for label prediction, baseline only for the label prediction task

Results: Overall Comparisons



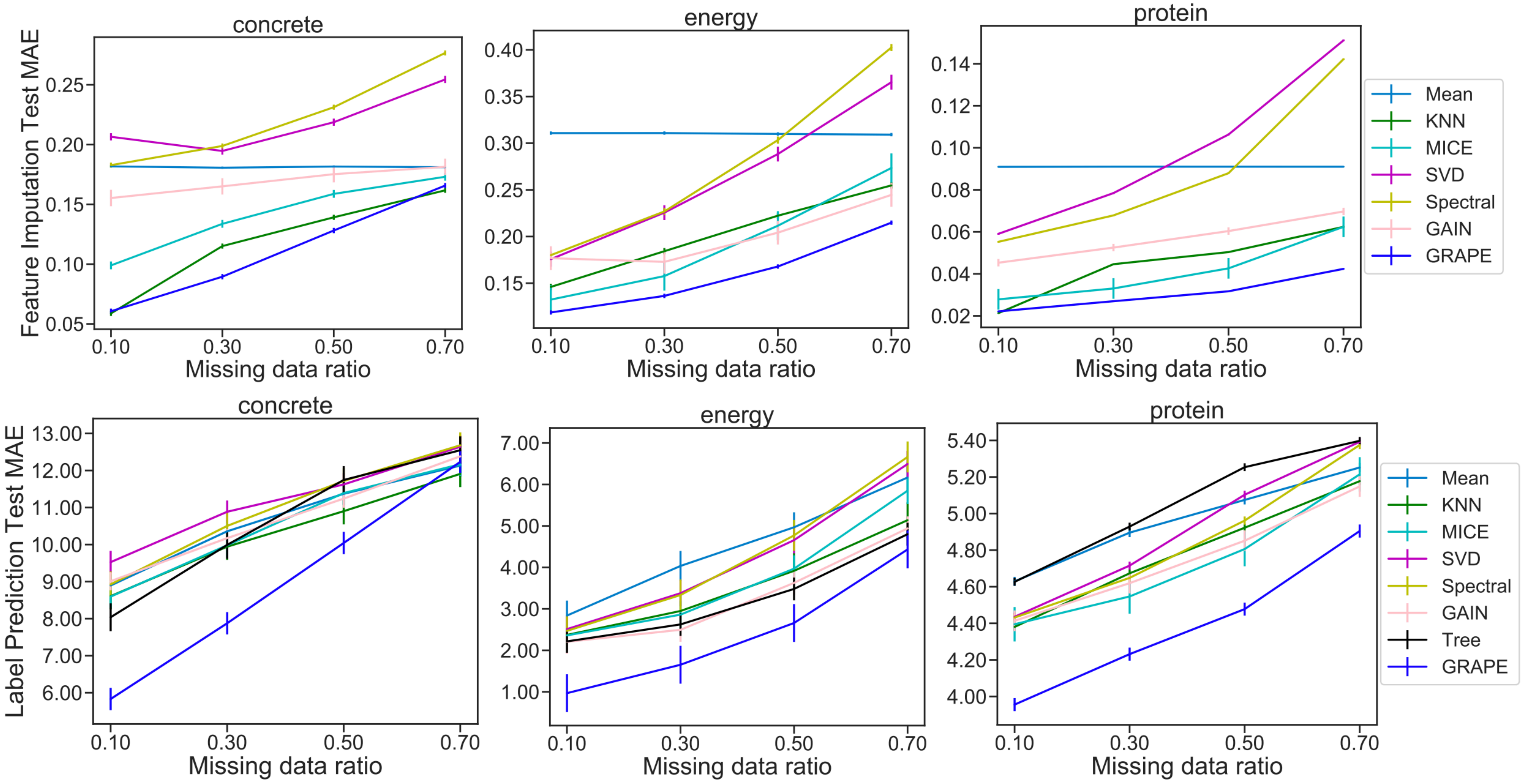
Feature imputation: 20% lower MAE than best baseline (KNN)



Label prediction: 10% lower MAE than best baseline (MICE)

Averaged MAE of *feature imputation*(upper) and *label prediction*(lower) on UCI datasets over 5 trials at data missing level of 0.3.

Results: Varying Missing Data Ratio



Averaged MAE of *feature imputation*(upper) and *label prediction*(lower) with different missing ratios over 5 trials. GRAPE yields 12% lower MAE on imputation and 2% lower MAE on prediction tasks across different missing data ratios

Results: Ablation Study

	concrete	energy	housing	kin8nm	naval	power	protein	wine	yacht
Without edge dropout	0.171	0.148	0.104	0.262	0.021	0.192	0.047	0.094	0.204
With edge dropout	**0.090**	**0.136**	**0.075**	**0.249**	**0.008**	**0.102**	**0.027**	**0.063**	**0.151**
SUM(·)	0.094	0.143	0.078	0.277	0.024	0.134	0.040	0.069	0.154
MAX(·)	**0.088**	0.142	**0.074**	0.252	**0.006**	**0.102**	**0.024**	**0.063**	0.153
MEAN(·)	0.090	**0.136**	0.075	**0.249**	0.008	**0.102**	0.027	**0.063**	**0.151**
Impute then predict	9.36	2.59	3.80	0.181	**0.004**	4.80	4.48	**0.524**	9.02
End-to-End	**7.88**	**1.65**	**3.39**	**0.163**	0.007	**4.61**	**4.23**	0.535	**4.72**

Averaged MAE of GRAPE on UCI datasets over 5 trials
Edge dropout (upper) reduces the average MAE by 33% on feature imputation tasks.
MEAN(·) is adopted in implementation given by authors.
End-to-End training (lower) reduces the average MAE by 19% on prediction tasks (excluding two outliers).

Conclusion

Feature Imputation - edge level task

Label Prediction - node level task

Learning in end-to-end fashion

Extension of GNNs to include edge values

Significant improvement over state-of-the-art approaches