**FEATURE SIDE- INFORMATION (FSI)**

- Feature side-information: vectorial descriptions of the features, which give more detailed information about feature’s property, typically derived from domain knowledge.

Examples:
- Chemoinformatics → molecule properties
- Genomics/proteomics → gene and protein properties, pathways and more

Text classification → word2vec

**OBJECTIVE**

- Standard supervised learning:
  \[
  X = \begin{bmatrix}
    x_1 & x_2 & \cdots & x_n
  \end{bmatrix},
  \quad \phi = \begin{bmatrix}
    \phi_1 & \phi_2 & \cdots & \phi_m
  \end{bmatrix},
  \quad Y = \begin{bmatrix}
    y_1 & y_2 & \cdots & y_m
  \end{bmatrix},
  \quad \text{where } k \text{ is the number of classes}
  \]

- Supervised learning with FSI:
  \[
  X = \begin{bmatrix}
    x_{11} & x_{12} & \cdots & x_{1n}
    \vdots & \vdots & \ddots & \vdots
    x_{m1} & x_{m2} & \cdots & x_{mn}
  \end{bmatrix},
  \quad \phi = \begin{bmatrix}
    \phi_{11} & \phi_{12} & \cdots & \phi_{1n}
    \vdots & \vdots & \ddots & \vdots
    \phi_{m1} & \phi_{m2} & \cdots & \phi_{mn}
  \end{bmatrix},
  \quad Y = \begin{bmatrix}
    y_1 & y_2 & \cdots & y_m
  \end{bmatrix},
  \quad \text{where } \lambda \text{ is the number of features}
  \]

- Our objective:
  Learn a mapping \( \phi: x \in \mathbb{R}^d \rightarrow y \in \mathbb{R}^m \) using the information provided not only by the matrices \( X, Y \), but also by matrix \( Z \)
  \( X \): instance matrix
  \( Y \): label matrix
  \( Z \): feature side-information matrix

**MAIN IDEA**

- Main idea: the model should treat similar features in a similar manner, i.e., small relative changes in similar features should leave model outputs unaffected.

\[
\phi(x + \lambda_i e_i + \lambda_j e_j) \approx \phi(x + \lambda'_i e_i + \lambda'_j e_j)
\]

where \( \lambda_i + \lambda_j = \lambda'_i + \lambda'_j = c \)

In the limit case, if \( i, j \), features are identical, we have:

**APPROXIMATIONS OF THE REGULARIZER**

- Analytical approximation:
  \[
  \hat{R}(\phi) \approx \sum_{ij} \left( \| \nabla_i \phi(x) - \nabla_j \phi(x) \|^2 \right) S_{ij} P(x) dx
  \]
  \[
  \approx \int \text{Tr}(J(x)LJ^T(x))P(x) dx
  \]

With sample estimation given by:

\[
\hat{R}(\phi) = \sum_{ij} \left( \| \nabla_i \phi(x_k) - \nabla_j \phi(x_k) \|^2 \right) S_{ij}
\]

where \( \nabla_i \phi(x_k) = \frac{\partial \phi}{\partial x_i} \bigg|_{x=x_k} \)

\[
\text{Stochastic approximation of the regularizer is:}
\]

\[
\hat{R}(\phi) = \sum_{ij} \left( \| \phi(x_k + \lambda_i e_i + \lambda_j e_j) - \phi(x_k + \lambda_i' e_i + \lambda_j' e_j) \|^2 \right) S_{ij}
\]

**REGULARIZER DRIVEN BY FSI**

- Pairwise feature regularizer:
  \[
  R_{ij}(\phi) = \int \| \phi(x + \lambda_i e_i + \lambda_j e_j) - \phi(x + \lambda_i' e_i + \lambda_j' e_j) \|^2 S_{ij} I(\lambda) P(x) dx
  \]

where \( I(\lambda) = \begin{cases} 1 & \text{if } \lambda_i + \lambda_j = \lambda'_i + \lambda'_j, \\
0 & \text{otherwise} \end{cases} \)

- Feature side-information: word2vec

**EXPERIMENTAL RESULTS**

- Real world experiments
  - Text classification: word counts
  - Feature side-information: word2vec

**SUMMARY**

- We incorporate feature-side information in the learning of nonlinear models
- Our method provides an indirect way to apply many of the regularizers used in linear models to nonlinear ones.

**REFERENCES**


**FUTURE RESEARCH**

Performance difference is less striking in the real data, we have a number of hypotheses:
- The similarity matrix might not be appropriate for the task at hand.
- Augmentation size can vary depending on where we are in the feature space.
- We are currently exploring strategies to address the above limitations.

**OPEN POSITION**

We have an open post doc position in a project related to this one.

Email: Alexandros.Kalousis@hesge.ch