Computer Science Department - Rutgers University **Fall 2020**

CS 536 : Final Project - Data Completion and Interpolation 16:198:536

For your final project, you will take a data set of the form $\{\underline{x}^1, \underline{x}^2, \ldots, \underline{x}^m\}$, and construct a system for predicting or interpolating missing features (frequently given as empty or NA) from the present features in new records. Notice that in this case, instead of singling out a single feature or factor for prediction/regression/classification, *any* feature of $\underline{X}$ might be missing and need to be predicted or reconstructed from the features that are present - and the features that are present may vary from instance to instance.

> As an example of such a system, we might do the following as a **basic completion agent**:
>
> - For a data point $\underline{x}$ missing feature i,
>   - If feature i is discrete, fill x_i with the **mode** of feature i from the data set.
>   - If feature i is real, fill x_i with the **mean** of feature i from the data set.
> - Iterate over each feature missing i, completing $\underline{x}$.
>
> The completed $\underline{x}$ will then have the 'typical' properties of the data set.

Notice here though, that one thing the above completion agent does not do is take into account any of the features of $\underline{x}$ that are actually present.

Before getting into the specifics of the data, here are some problems that you will have to address when coming up with a solution:

- How to represent or process the data. Data features may contain a number of diverse data types (real values, integer values, categorial or binary values, ordered categorial values, open/natural language responses). How can you represent these for processing and prediction?
- How to model the problem of interpolation. What are the inputs, what are the outputs? An important if subtle question to consider here - what does it mean to predictor or interpolate a missing feature?
- Model selection. What kind of model or models do you want to consider?
- Quantifying loss or error. How can you quantify how good a model is, how to measure its loss/error? This is important not only in terms of evaluating your model, but in terms of training as well - how can you refine and improve your model without a way of comparing them?
- Training. What kind of training algorithm can you apply to your model(s)? What design choices do you have to make here?
- Feature selection. It is frequently useful in learning problems to focus on specific features and exclude others, to try to eliminate spurious features and focus on what matters. How can that be applied here?
- Validation. How can you prevent or avoid over-fitting? Can you apply the usual training/testing/validation paradigm to this problem? How do you choose the training or testing data? Note that a record won't need to be complete to still be useful, potentially, in interpolation. Can cross-validation be applied here? This can be especially important when the data set is not overwhelmingly large and data must be used carefully.
- Evaluation. How good is your final model? How can you evaluate this? What are the limits and strengths of your model - how many features does a new record actually need to be able to interpolate well?

These are important questions to answer when building and evaluating machine learning systems, and I expect thorough treatments of each in your final writeup.

1 Requirements

You must submit any code written to complete the assignment (and any supplemental model files, etc). The code should be well-commented and clear. Additionally, the code should be your own - any learning algorithms you implement should be your own work, and not rely on other people's algorithms or external libraries. You may use existing libraries or frameworks to help *represent* your model and handle it in code, but you are not allowed to use any built in training or optimization algorithms, or any pre-coded machine learning algorithms. Your system must learn based on your own algorithms.

Additionally, you must submit a writeup that includes and addresses the following points, quantifying relationships and plotting data where appropriate:

- **Describe your Model:** What approach did you take? What design choices did you make, and why? How did you represent the data? How can you evaluate your model for goodness of fit? Did you make an effort to identify and exclude irrelevant variables? How did you handle missing data?
- **Describe your Training Algorithm:** Given your model, how did you fit it to the data? What design choices did you make, and why? Were you able to train over all features? What kinds of computational tradeoffs did you face, and how did you settle them?
- **Describe your Model Validation:** How did you try to avoid overfitting the data? How did you handle the modest (in ML terms) size of the data set?
- **Evaluate your Model:** Where is your model particularly successful, where does it lack? Does it need a certain amount of features in order to interpolate well? Are there some features it is really good at predicting and some it is really poor at predicting? Why do you think that is? How does your model stack up against the basic completion agent?
- **Analyze the Data:** What features were particularly valuable in predicting/interpolating? What features weren't particularly useful at all? Were there any features about the researcher/experimental environment that were particularly relevant to predicting answers - and if so, what conclusions can you draw about the replicability of those effects?
- **Generate Data:** Use your system to try to generate realistic data, and compare your generated data to the real data. How good does it look? What does it mean for it to 'look good'?

Bonus: A bonus is available for a 5 minute video summarizing your project and results. Graphs are a must.