Manipulating Features

• Data preparation is important
• What to do if we have too many features? Or feature data not in the form expected?
• Transformations for individual features
• Transformation/selection over the entire feature set

Normalizing Feature Range

• Range of values of a feature can affect performance especially for distance based methods
• e.g. one feature can dominate distance

Normalizing Feature Range

• Linear scaling into [0,1]
  \[ x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \]
• Z-normalization: scale to have mean 0 and std 1
  \[ x' = \frac{x - \mu_x}{\sigma_x} \]
  min, max, mu, sigma calculated from train set only

Discretizing Features

• What if we want to use a discrete-only learning algorithm but have data with real valued features?
• Preprocess to discretize feature
• How?

Discretizing Features

• Unsupervised methods:
  (do not use label to determine partition)
  - Equal bin size (a fixed grid)
    • Predetermined by feature range
    • Pros and cons?
  - Equal frequency
    • Divide to get same number of points in each bin
    • Adapt to data
    • Pros and cons?
Discretizing Features

- Supervised methods (use label):
  - Looks for partitions that help identify label
  - Example: run DT algorithm on single feature and include all node splits from the tree.

Discrete to Numerical Features

- We have discrete features (with unordered or ordinal values) but want to use a learning algorithm that expects numerical features. How?
- Say $x_i$ has possible values A,B,C,D and example has feature value C?

Two dominant approaches:
- As unit vectors: 1000,0100,0010,0001
- As "increasing weight" vectors: 1000,1100,1110,1111

Too many features

- When working on an application it is tempting to identify and record more and more features that may be relevant for classification
- In some applications (e.g., text) there is a huge set of potential features (e.g., in bag of words representation)
- But too many features can be harmful
- Why?

kNN distance function and classification dominated by extra features
- In general with many random features we are likely to get correlations by chance and this leads to overfitting
- Solutions:
  - Instance transformation
  - Feature selection
**Instance Transformation**

- Sometimes the data is given in a high dim space but has some hidden structure
- Unveiling this structure enables data analysis

---

**Manifold Methods**

- Data resides on a "manifold"

![Embedding](image)

- Embed data in low dim space, while preserving local distances
  - [we will not cover technical details]

---

**Relief Algorithm**

- We have already discussed Relief in the context of kNN
- Pick instance + nearest hit + nearest miss
- Update weights based on distances and whether hit is closer/farther than miss
  \[ \|x - y\|^2_w = \sum w_i (x_i - y_i)^2 \]
- Can also be used to select features.

---

**Principal Component Analysis**

- PCA is a standard method for instance transformation
- It is "unsupervised" in that it looks only at example locations and not their labels
- Intuitively: project the data onto k dimensions with highest variation
  - [we will not cover technical details]

---

**Feature Selection**

- **Supervised methods** make use of labels to identify useful features.
  - How?
    - Ad hoc methods
    - Filter methods
    - Wrapper methods

---

**Filter Methods**

- Pick a criterion for evaluating features
  - InfoGain
  - Correlation with label
  \[ \rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \]
  - Mutual information with label
  \[ I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \]
Filter Methods

- Pick a criterion for evaluating features
  - InfoGain
  - Correlation with label
    \[ \rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)\mid X, Y]}{\sigma_X \sigma_Y} \]
  - Mutual information with label
    \[ I(X;Y) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \]

Filter Methods

- What happens if we duplicate the top ranking feature?

Filter Methods

- What happens if we duplicate the top ranking feature?
  - Are single feature tests sufficient?

Filter Methods

- Are single feature tests sufficient?

Wrapper Method

- Adaptive: 2nd feature selected is chosen in context of already using the 1st etc
  - Evaluates feature sets not individual features
  - Selection tailored for the intended learning algorithm
• Eval subset by running and testing alg, e.g., using validation set

• Incremental search over subsets
• Forward, backward, combined, beam, ...

Search Method: sequential forward search

Visualization from Carla Brodley’s slides

• In the context of “machine learning as optimization” the L1 penalty is known to effectively induce sparsity in the results (similar to the weighting in Relief)

• [we will not cover technical details]