Feature selection

Lecture 13

IFT 6758, Fall 2020

What is feature selection?

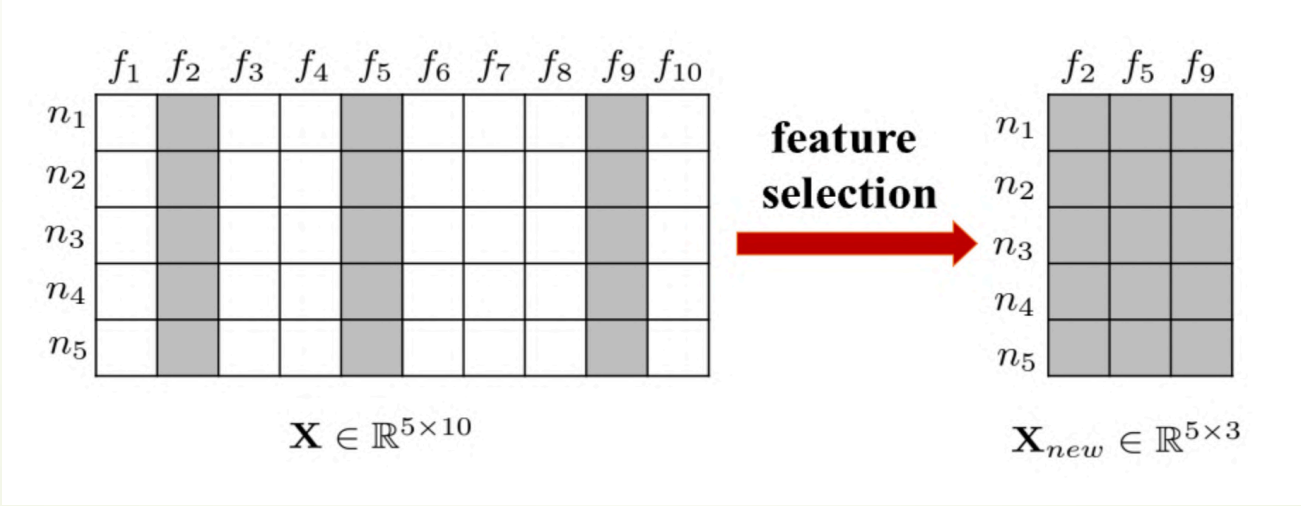
- A procedure in machine learning to find a subset of features that produces 'better' model for given dataset

What is feature selection?

- A procedure in machine learning to find a subset of features that produces 'better' model for given dataset
  - Avoid overfitting and achieve better generalization ability
  - Reduce the storage requirement and training time
  - Interpretability

What is feature selection?

- A procedure in machine learning to find a subset of features that produces 'better' model for given dataset
  - Avoid overfitting and achieve better generalization ability
  - Reduce the storage requirement and training time
  - Interpretability



Relevant vs redundant features

- Feature selection keeps relevant features for learning and removes redundant and irrelevant features

Relevant vs redundant features

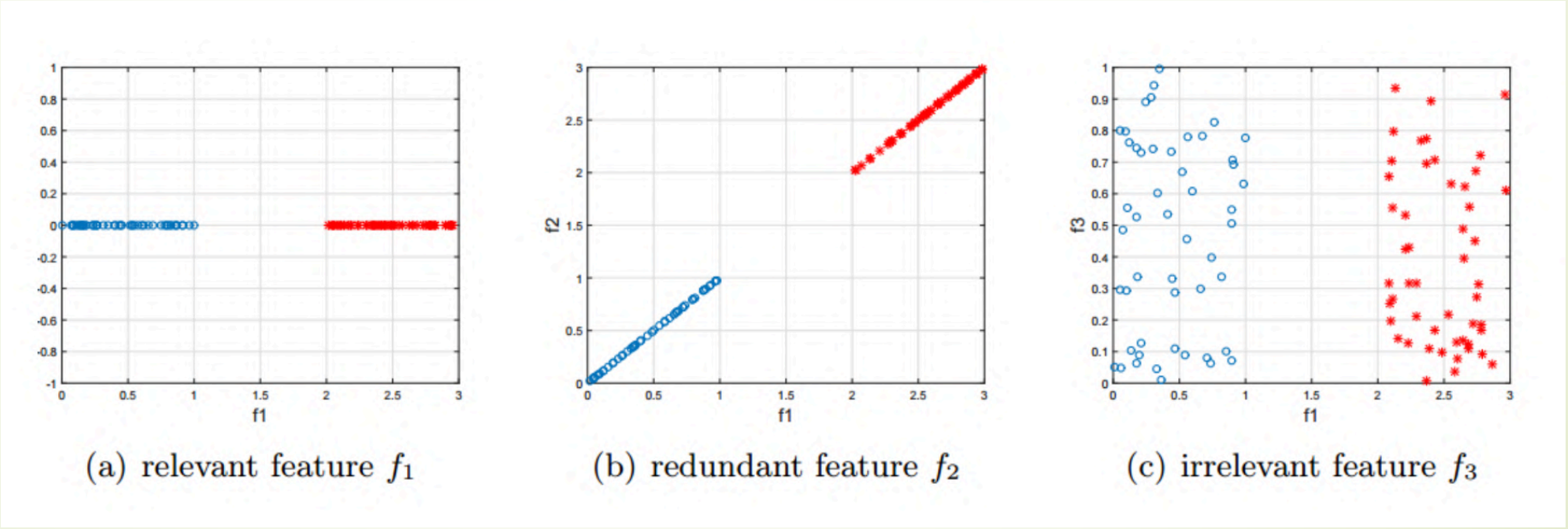
- Feature selection keeps relevant features for learning and removes redundant and irrelevant features
- For example, for a binary classification task (f1 is relevant; f2 is redundant given f1; f3 is irrelevant)

linearly dep

noise

Relevant vs redundant features

- Feature selection keeps relevant features for learning and removes redundant and irrelevant features
- For example, for a binary classification task (f1 is relevant; f2 is redundant given f1; f3 is irrelevant)



(a) relevant feature f_1 (b) redundant feature f_2 (c) irrelevant feature f_3

Feature Extraction vs. Feature Selection

Feature Extraction vs. Feature Selection

- **Commonalities**
  - Speed up the learning process
  - Reduce the storage requirements
  - Improve the learning performance
  - Build more generalized models

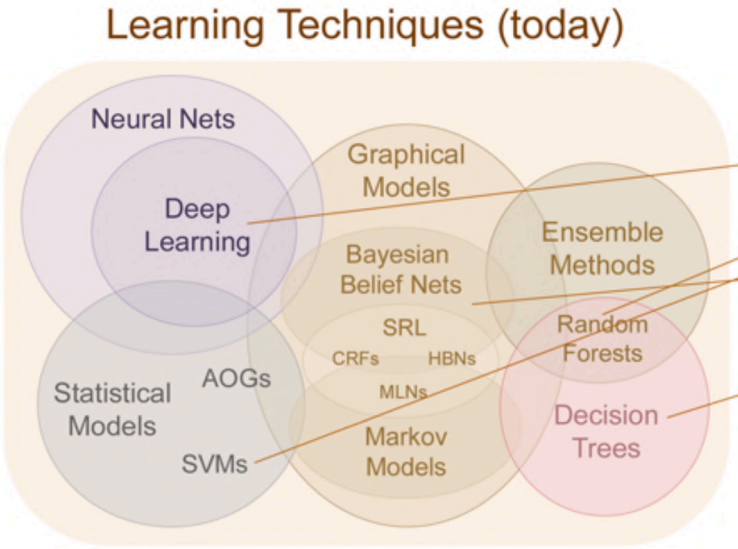
Feature Extraction vs. Feature Selection

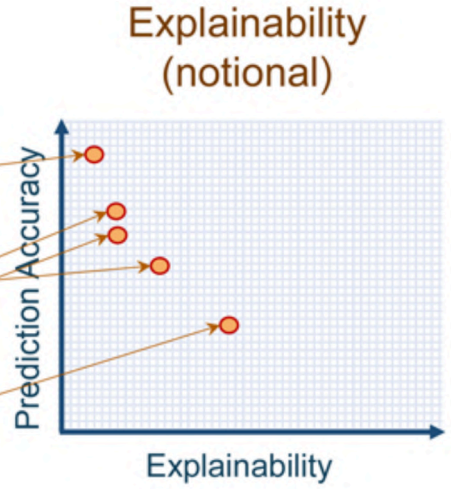
- **Commonalities**
  - Speed up the learning process
  - Reduce the storage requirements
  - Improve the learning performance
  - Build more generalized models

- **Differences**
  - Feature extraction obtains new features while feature selection selects a subset of original ones
  - Feature selection maintains physical meanings and gives models better readability and interpretability

we can plot it in 2 or 3 D, for better understanding/interpretability/explainabilit y

Interpretability of Learning Algorithms





http://nautil.us/issue/40/learning/is-artificial-intelligence-permanently-inscrutable

With feature selection, both the accuracy and interpretability of most learning algorithms can be enhanced !

When feature selection is important?

- Noisy data
- Lots of less frequent features
- Use multi-type features
- Too many features compared to samples reduces overfitting
- Complex model bias vs variance
- Samples in real scenario is inhomogeneous with training & test samples

Feature Selection Algorithms

Feature Selection Algorithms

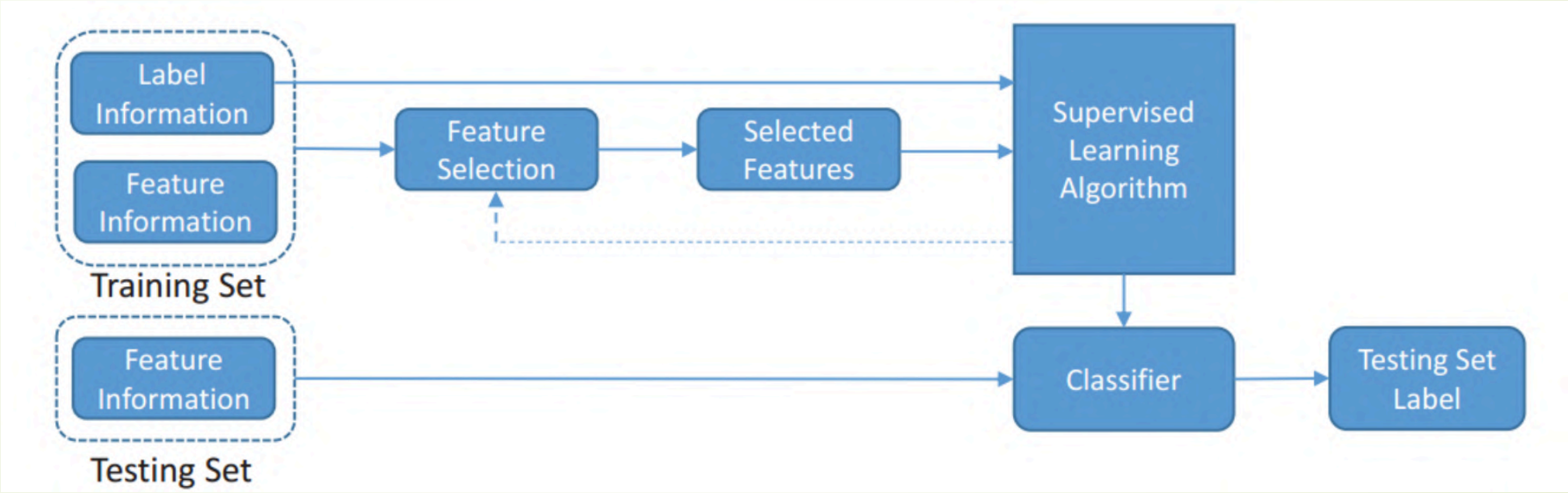
- From the **label perspective** (whether label information is involved during the selection phase):
  - Supervised
  - Unsupervised
  - Semi-Supervised

Feature Selection Algorithms

- From the **label perspective** (whether label information is involved during the selection phase):
  - Supervised
  - Unsupervised
  - Semi-Supervised
- From the **selection strategy perspective** (how the features are selected):
  - Wrapper methods
  - Filter methods
  - Embedded methods

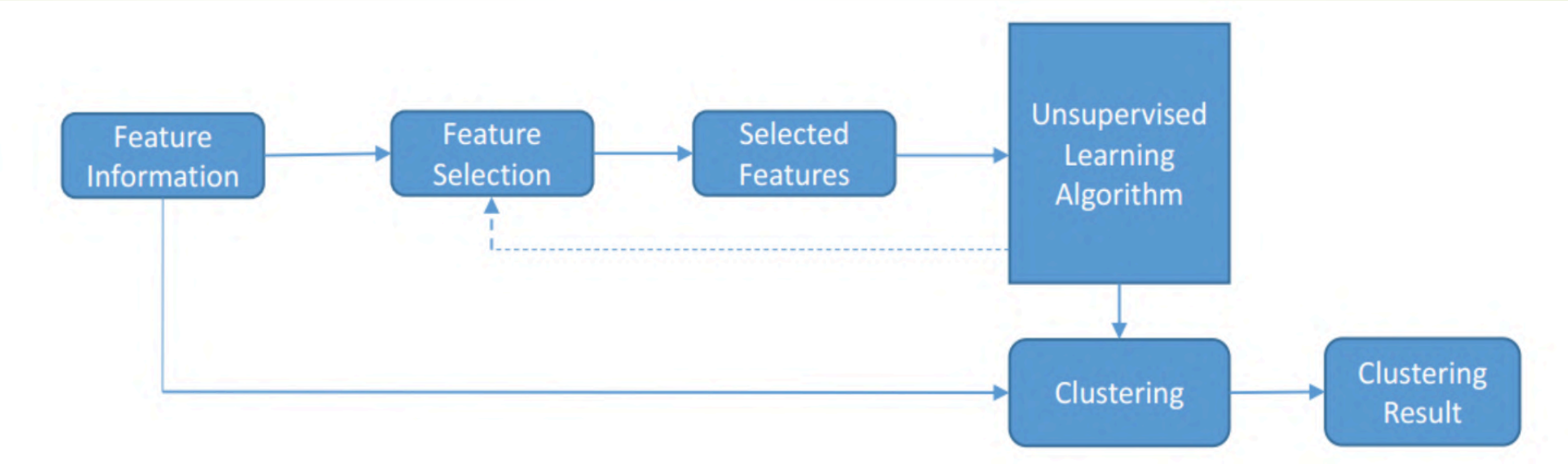
Supervised Feature Selection

- Supervised feature selection is often for classification or regression problems
- Find discriminative features that separate samples from different classes (classification) or approximate target variables (regression)



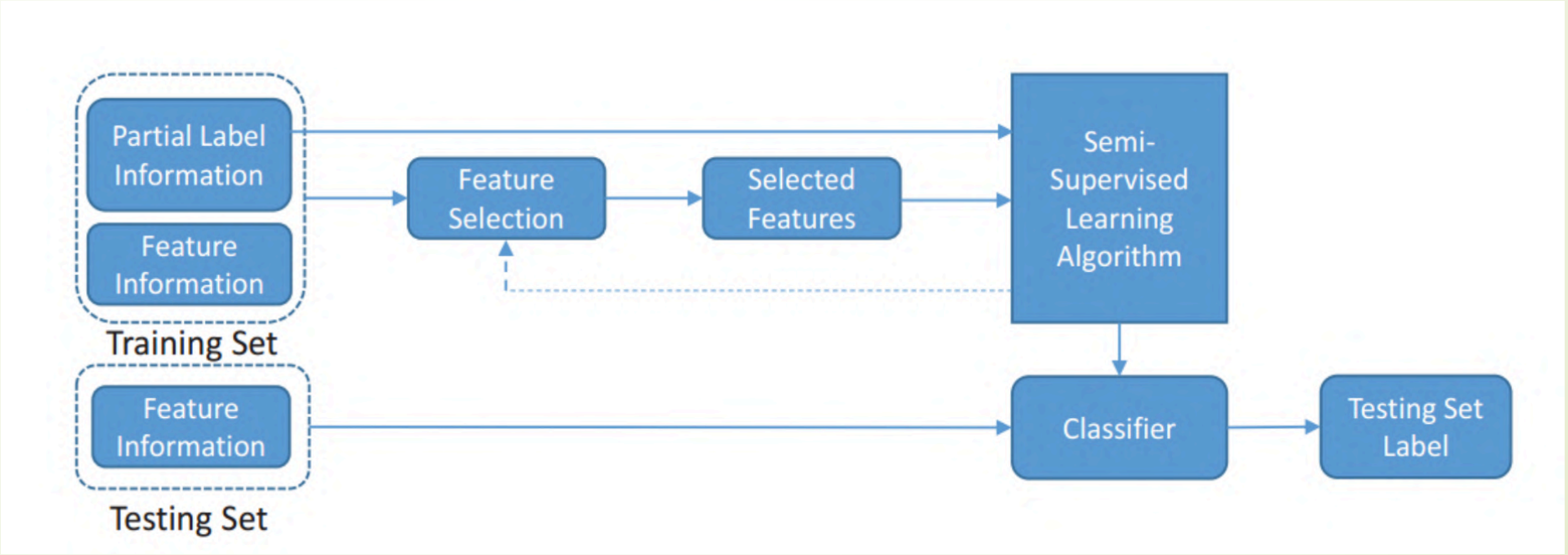
Unsupervised Feature Selection

- It is often for clustering problems
- Label information is expensive to obtain which requires both time and efforts
- Unsupervised methods seek alternative criteria to define feature relevance

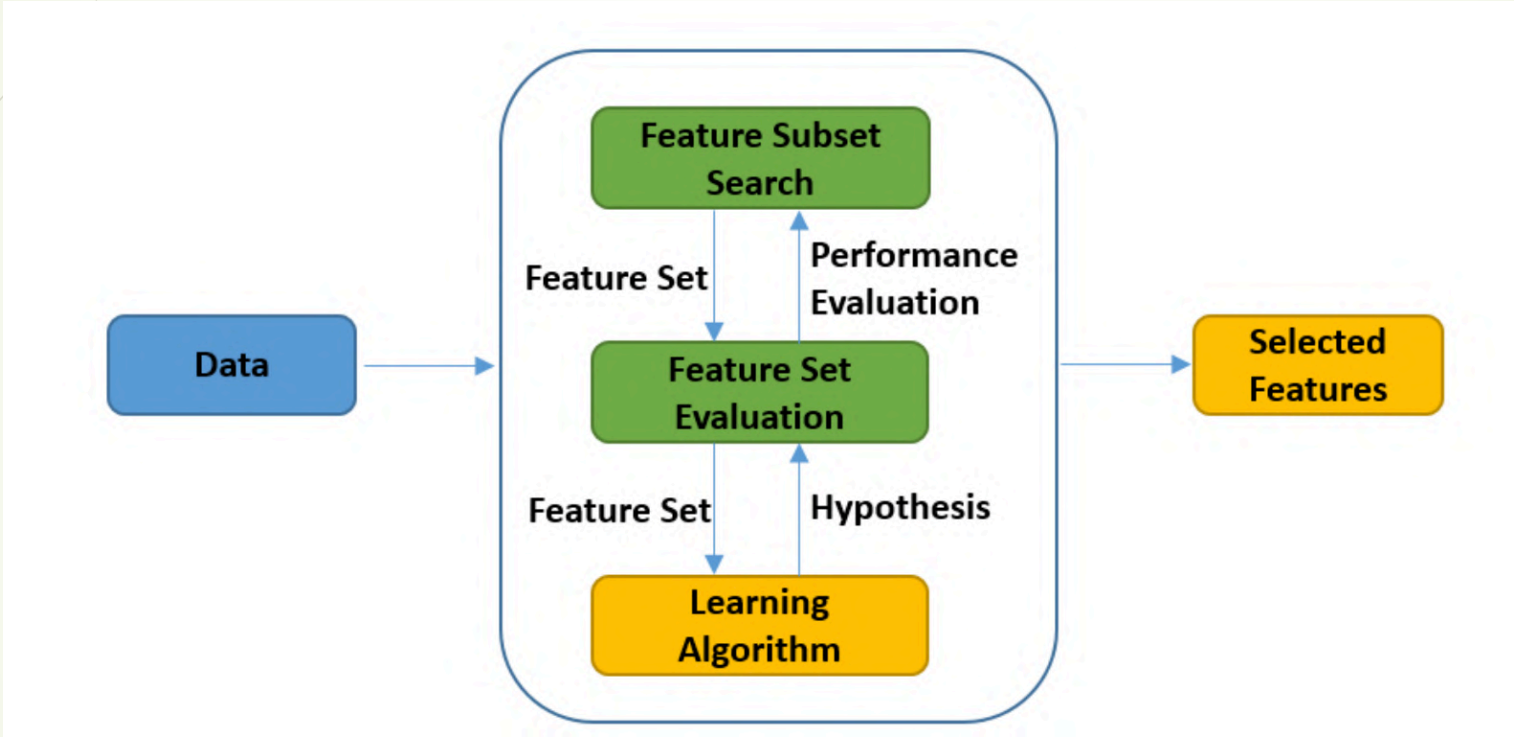


Semi-Supervised Feature Selection

- We often have a small amount of labeled data and a large amount of unlabeled data
- Semi-supervised methods exploit both labeled and unlabeled data to find relevant features



Wrapper Methods



- Step 1: search for a subset of features
- Step 2: evaluate the selected features
- Repeat Step 1 and Step 2 until stopped

Feature Selection Techniques

Subset selection method : Forward Search and Backward Search

Feature Selection Techniques

Subset selection method : Forward Search and Backward Search

- **Forward Search**
  - Start with no features
  - Greedily include the most relevant feature
  - Stop when selected the desired number of features

Feature Selection Techniques

Subset selection method : Forward Search and Backward Search

- **Forward Search**
  - Start with no features
  - Greedily include the most relevant feature
  - Stop when selected the desired number of features

- **Backward Search**
  - Start with all the features
  - Greedily remove the least relevant feature
  - Stop when selected the desired number of features

Feature Selection Techniques

Subset selection method : Forward Search and Backward Search

- **Forward Search**
  - Start with no features
  - Greedily include the most relevant feature
  - Stop when selected the desired number of features

- **Backward Search**
  - Start with all the features
  - Greedily remove the least relevant feature
  - Stop when selected the desired number of features

- Inclusion/Removal criteria uses **cross-validation**

Wrapper Methods

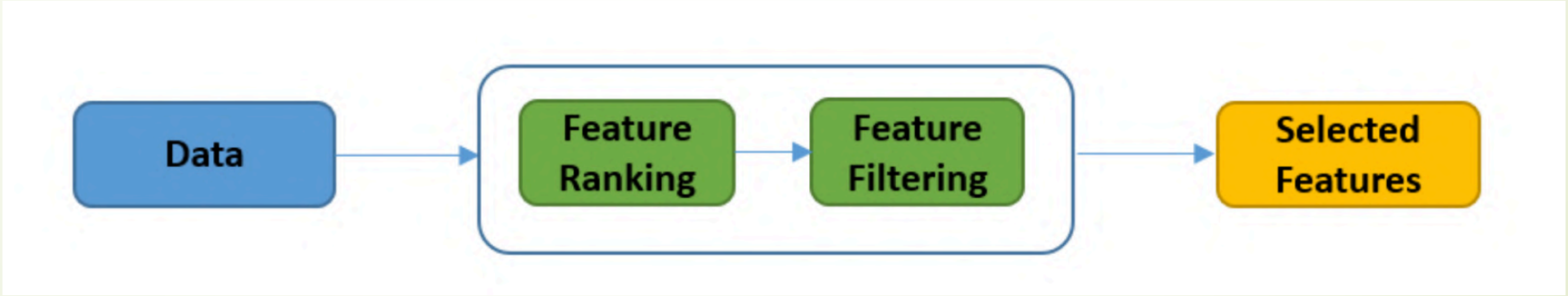
- Can be applied for ANY model
- Rely on the predictive performance of a **predefined learning algorithm** to assess features
- **Shrink / grow feature set** by greedy search
- Repeat until **some stopping criteria** are satisfied
- Achieve **high accuracy** for a particular learning method
- Run **CV / train-val split per feature**

- **Computationally expensive** (worst case search space is 2^d), some typical search strategies are
  - Sequential search
  - Best-first search
  - Branch-and-bound search

exponential complexity, which is bad

Filter Methods

- **Independent** of any learning algorithms
- **Rely on certain characteristics** of data to assess feature importance (e.g., feature correlation, mutual information…)
- More **efficient** than wrapper methods
- The selected features **may not be optimal** for a particular learning algorithm



FILTER METHOD

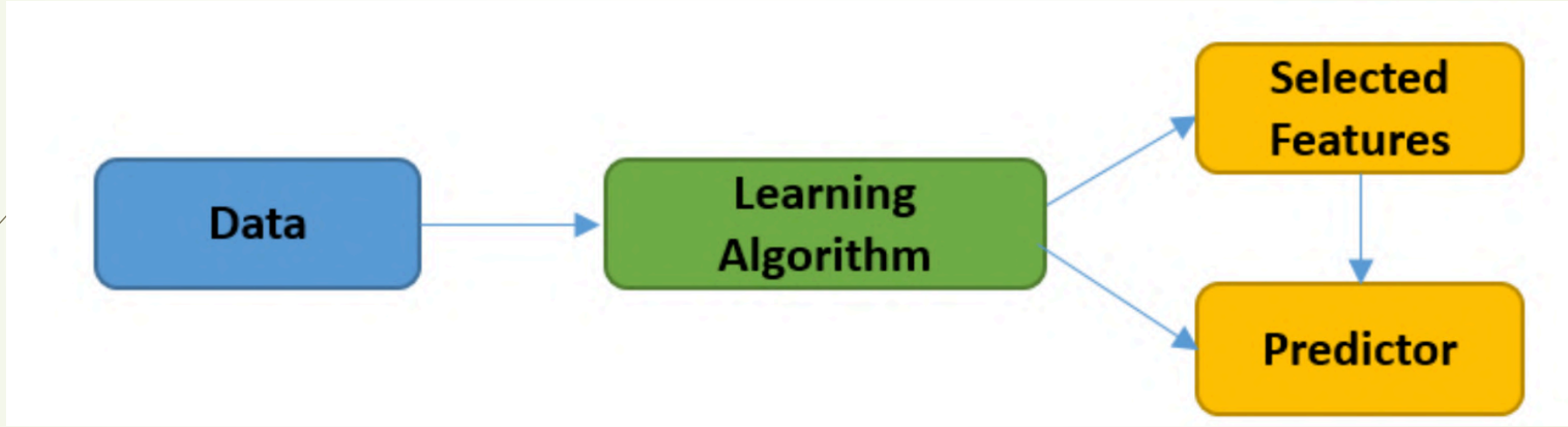
Feature Selection Techniques

Single feature evaluation: Measure quality of features by all kinds of metrics

- Frequency based
- Dependence of feature and label (co-occurrence), e.g., *Mutual information, Chi square statistic*
- Information theory, KL divergence, Information gain
- Gini indexing

Embedded Methods

- A trade-off between wrapper and filter methods by embedding feature selection into the model learning, e.g., ID3



- Inherit the merits of wrapper and filter methods
  - Include the interactions with the learning algorithm
  - More efficient than wrapper methods
- Like wrapper methods, they are biased to the underlying learning algorithms

Selection Criteria

Traditional feature selection

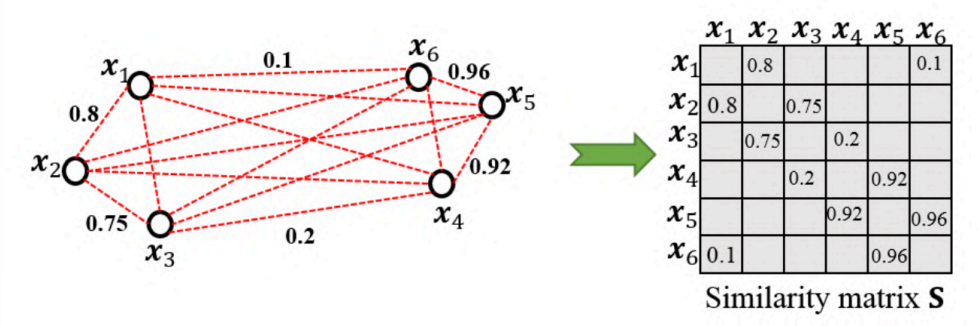
- Similarity based methods
- Information Theory based methods
- Sparse learning based methods
- Statistical methods

Similarity based methods

Similarity technique

Finding similarity index between featurs and choose .

- Pairwise data similarity is often encoded in the data similarity matrix



- E.g., without class label information, it can be defined by the RBF kernel

$$\mathbf{S}_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

- E.g., using the class labels, the similarity can be obtained as

$$\mathbf{S}_{ij} = \begin{cases} \frac{1}{n_l} & \text{if } y_i = y_j = l \\ 0 & \text{otherwise} \end{cases}$$

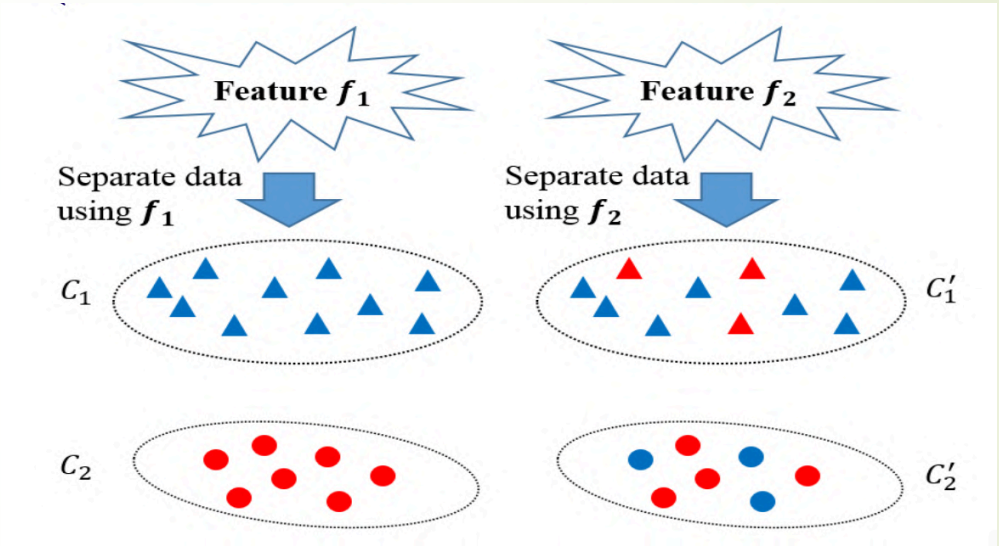
Similarity based Feature Selection

- Similarity based methods assess the importance of features by their ability to preserve data similarity
- A good feature should not randomly assign values to data instances
- A good feature should assign similar values to instances that are close to each other – (the "closeness" is obtained from data similarity matrix)

Similarity based Feature Selection

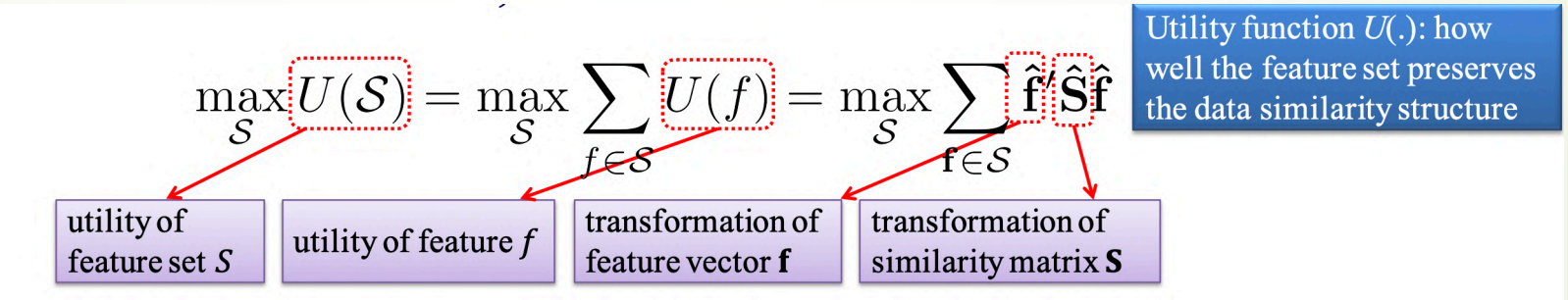
- Similarity based methods assess the importance of features by their ability to preserve data similarity
- A good feature should not randomly assign values to data instances
- A good feature should assign similar values to instances that are close to each other – (the "closeness" is obtained from data similarity matrix)

- Different shapes denote different values assigned by a feature



Similarity based Methods – A General Framework

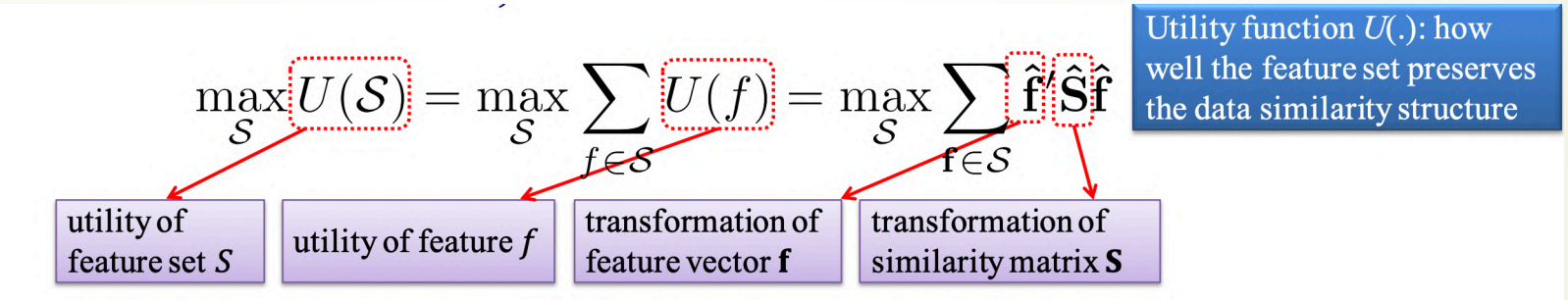
- Suppose data similarity matrix is to find the most relevant features , we need to maximize:



- It is often solved by greedily selecting the top features that maximize their individual utility $U(f)$
- Different methods vary in the way how the vector $\boldsymbol{f}$ and similarity matrix $\boldsymbol{S}$ are transformed to $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{S}}$

Similarity based Methods – A General Framework

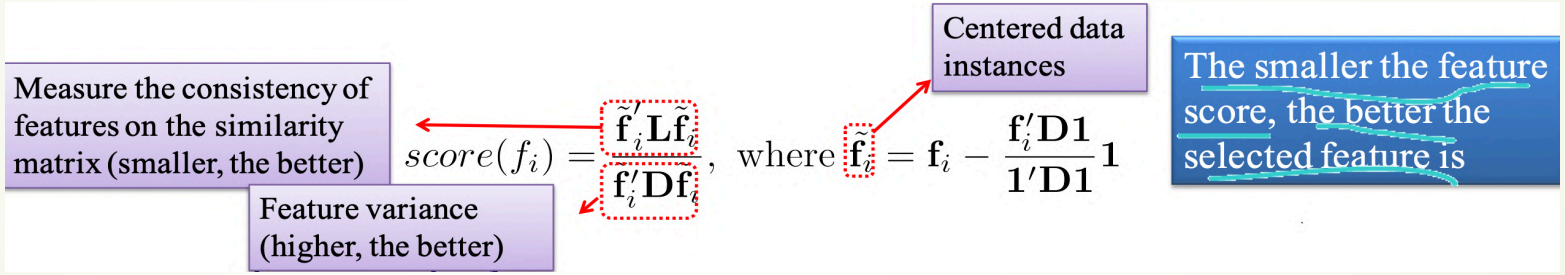
- Suppose data similarity matrix is to find the most relevant features , we need to maximize:



- It is often solved by greedily selecting the top features that maximize their individual utility $U(f)$
- Different methods vary in the way how the vector $\boldsymbol{f}$ and similarity matrix $\boldsymbol{S}$ are transformed to $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{S}}$

Laplacian Score [He et al., 2005]

- First, it builds the data similarity matrix S, diagonal matrix D and Laplacian matrix L without using class labels
- Motivation: a good feature should (1) preserve data similarity structure; and (2) have high feature variance (The features are completely differnet)
- Then the Laplacian Score of feature $\boldsymbol{f_i}$ is:



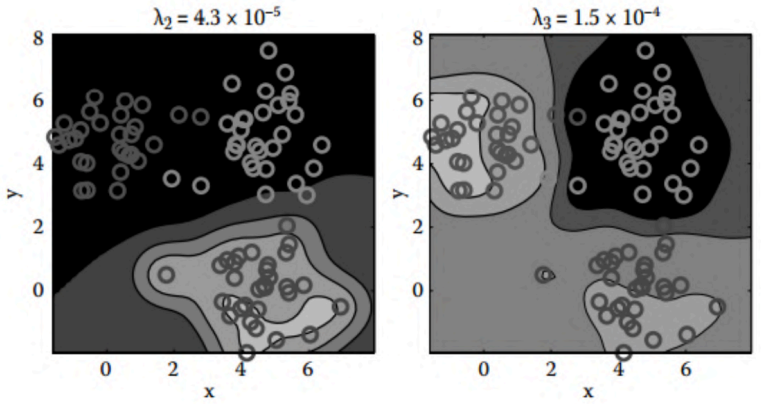
- Laplacian score is also equivalent to:

$$1 - \left(\frac{\tilde{\mathbf{f}}_i}{||\mathbf{D}^{\frac{1}{2}} \tilde{\mathbf{f}}_i||} \right)' \mathbf{S} \left(\frac{\tilde{\mathbf{f}}_i}{||\mathbf{D}^{\frac{1}{2}} \tilde{\mathbf{f}}_i||} \right)$$

Spectral Feature Selection [Zhao and Liu, 2007]

- Eigenvectors of similarity matrix $\mathbf{S}$ carry the data distribution

The 2nd and the 3rd eigenvectors from **S**

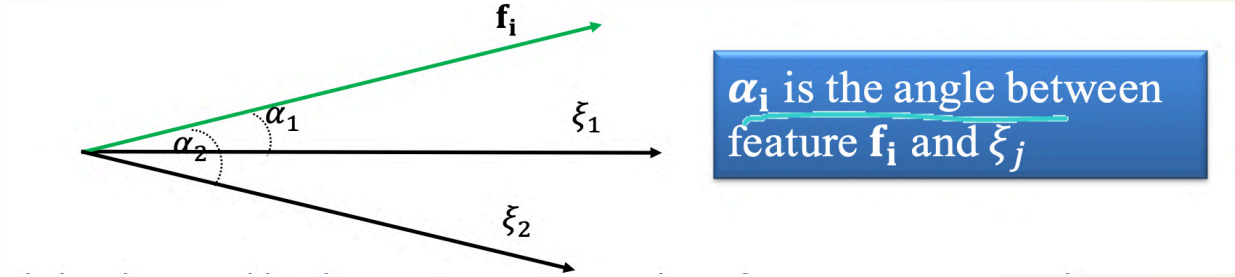


The gray level of the background shows how eigenvectors assign values to the samples

- Observation: eigenvectors assign similar values to the samples that are of the same affiliations

Spectral Feature Selection [Zhao and Liu, 2007]

- Measure feature f_i's consistency by comparing it with the eigenvectors (e.g., ξ_j) using inner product ${\xi_j}^T f_i$



- By considering all eigenvectors, the feature score is:

$$score(f_i) = \sum_{j=1}^{n} \lambda_j (\xi_j' \mathbf{f}_i) = \mathbf{f}_i' \mathbf{S} \mathbf{f}_i$$

Eigenvalues

The higher the feature score, the better the selected feature is

Fisher Score [Duda et al., 2001]

- Given class labels, within class and between class data similarity matrix S^w (local affinity) and S^b (global affinity) are defined as

$$\mathbf{S}_{i,j}^{w} = \begin{cases} 1/n_l & \text{if } y_i = y_j = l \\ 0 & \text{otherwise} \end{cases} \qquad \mathbf{S}_{i,j}^{b} = \begin{cases} 1/n - 1/n_l & \text{if } y_i = y_j = l \\ 1/n & \text{otherwise} \end{cases}$$

- $\boldsymbol{S_{ij}}^{\boldsymbol{w}}$ is larger if x_i and x_j belong to the same class, smaller otherwise
- $\boldsymbol{S_{ij}}^{\boldsymbol{b}}$ is larger if x_i and x_j belong to the different class, smaller otherwise

- A good feature should make instances from different classes far away and make instances from the same class close to each other

Fisher Score [Duda et al., 2001]

- The score of the i-th feature f_i is:

$$score(f_i) = \frac{\mathbf{f}_i' \mathbf{L}^b \mathbf{f}_i}{\mathbf{f}_i' \mathbf{L}^u \mathbf{f}_i}$$

Laplacian matrix obtained from $\mathbf{S}^w$ and $\mathbf{S}^b$

The larger the feature score, the better the selected feature is

- Fisher Score can be calculated from Laplacian Score:

$$fisher\_score(f_i) = 1 - \frac{1}{laplacian\_score(f_i)}$$

Trace Ratio Criteria [Nie et al., 2008]

- Fisher score evaluates the importance of features individually, which may lead to suboptimal solution

  Sometimes the features might be dependent, trace ratio, solves this problem

- Trace Ratio attempts to assess the importance of a subset of features $\mathcal{F}$ simultaneously

A trace ratio form

$$score(\mathcal{F}) = \frac{tr(\mathbf{X}'_{\mathcal{F}}\mathbf{L}^b\mathbf{X}_{\mathcal{F}})}{tr(\mathbf{X}'_F\mathbf{L}^w\mathbf{X}_F)} = \frac{\sum_{s=1}^k \mathbf{f}'_{i_s}\mathbf{S}^w\mathbf{f}_{i_s}}{\sum_{s=1}^k \mathbf{f}'_{i_s}(\mathbf{I}-\mathbf{S}^w)\mathbf{f}_{i_s}}$$

- Maximizing the above score is equivalent to maximize the following, which is a special case of the general framework

$$\frac{\sum_{s=1}^k \mathbf{f}'_{i_s}\mathbf{S}^w\mathbf{f}_{i_s}}{\sum_{s=1}^k \mathbf{f}'_{i_s}\mathbf{f}_{i_s}} = \frac{\mathbf{X}'_{\mathcal{F}}\mathbf{S}^w\mathbf{X}_{\mathcal{F}}}{\mathbf{X}'_{\mathcal{F}}\mathbf{X}_{\mathcal{F}}}$$

Constant number

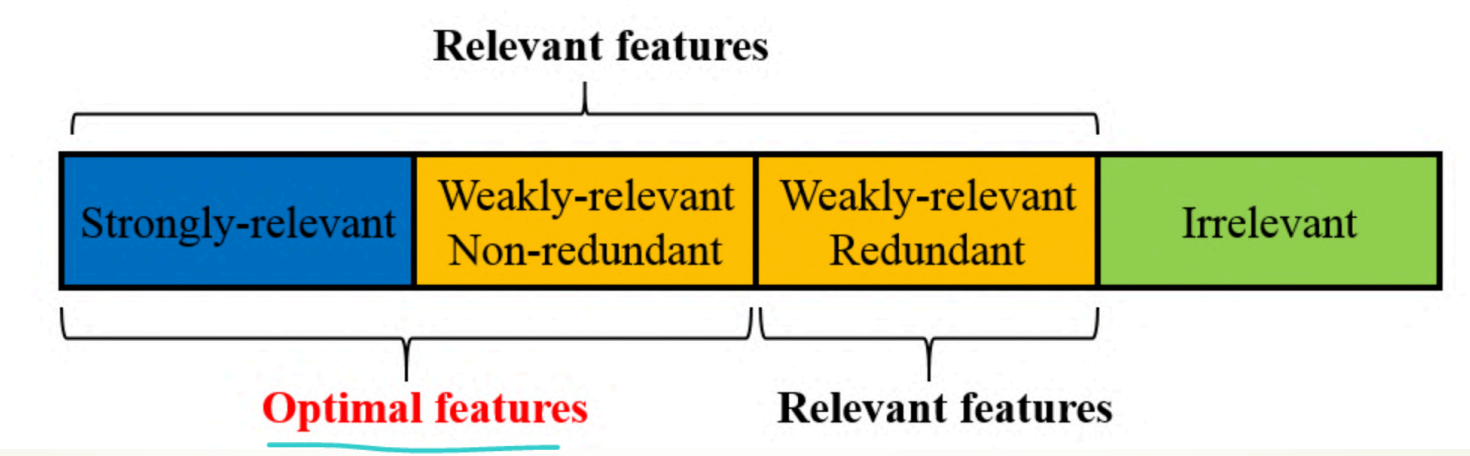
Similarity based Methods Summary

- Many others can also be reduced to the general similarity based feature selection framework
  - Batch-mode Laplacian score [Nie et al. 2008]
  - RelieF [Robnik-Sikonja and Kononenko, 2003]
  - HSIC Criterion [Song et al. 2007] …
- Pros:
  - Simple and easy to calculate the feature scores
  - Selected features can be generalized to subsequent learning tasks
- Cons:
  - Most methods cannot handle feature redundancy

Information Theory based methods

Information Theoretical based Methods

- Exploit different heuristic filter criteria to measure the importance of features



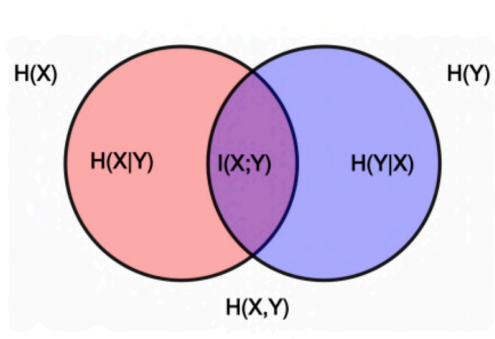
- Our target is to find these "optimal" features

Preliminary - Information Theoretical Measures

- Information gain between X and Y

$$I(X;Y) = H(X) - H(X|Y)$$
$$= \sum_{x_i \in X} \sum_{y_j \in Y} P(x_i, y_j) log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}$$

- Conditional information gain



$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$$
$$= \sum_{z_k \in Z} P(z_k) \sum_{x_i \in X} \sum_{y_j \in Y} P(x_i, y_j|z_k) log \frac{P(x_i, y_j|z_k)}{P(x_i|z_k)P(y_j|z_k)}$$

Information Theoretic based Methods - A General Framework

- Searching for the best feature subset is NP-hard, most methods employ forward/backward sequential search heuristics
- E.g., for forward search, given selected features S, we should do the following for the next selected feature f_k
  - Maximize its correlation with class labels Y: $I(f_k; Y)$
  - Minimize the redundancy w.r.t. selected features in S: $\sum_{f_j \in S} I(f_j; f_k)$
  - Maximize its complementary info w.r.t. selected features in S:

$$\sum_{f_j \in S} I(f_j; f_k | Y)$$

Information Theoretic based Methods - A General Framework

- Given selected features S, the feature score for the next selected feature can be determined by

$$score(f_k) = I(f_k; Y) + \sum_{f_j \in S} g[I(f_j; f_k), I(f_j; f_k|Y)]$$

The higher the feature score, the better the selected feature is

$g(*)$: a function

- If $g(*)$ is a linear function, then it can be represented as

$$score(f_k) = I(f_k; Y) - \beta \sum_{f_j \in S} I(f_j; f_k) + \lambda \sum_{f_j \in S} I(f_j; f_k|Y)$$

Between 0 and 1

- In general, $g(*)$ can be a nonlinear function

Information Gain [Lewis, 1992]

- Information gain only measures the feature importance by its correlation with class labels
- The information gain of a new unselected feature f_k

$$score(f_k) = I(f_k; Y)$$

- Selects features independently
- It is a special case of the linear function by setting $\beta = \lambda = 0$

$$score(f_k) = I(f_k; Y) - \beta \sum_{f_j \in \mathcal{S}} I(f_j; f_k) + \lambda \sum_{f_j \in \mathcal{S}} I(f_j; f_k | Y)$$

Mutual Information Feature Selection [Battiti, 1994]

- Information gain only considers feature relevance
- Features also should not be redundant to each other

- The score of a new unselected feature f_k

$$score(f_k) = I(f_k; Y) - \beta \sum_{f_j \in S} I(f_k; f_j)$$

maximize feature relevance

minimize feature redundancy

- It is also a special case of the linear function by setting $\lambda = 0$

Minimum Redundancy Maximum Relevance [Peng et al., 2005]

- Intuitively, with more selected features, the effect of feature redundancy should gradually decrease
- Meanwhile, pairwise feature independence becomes stronger
- The score of a new unselected feature f_k is

$$score(f_k) = I(f_k; Y) - \frac{1}{|\mathcal{S}|} \sum_{f_j \in \mathcal{S}} I(f_k; f_j)$$

reduced effect of feature redundancy

- MRMR is also a special case of the linear function by setting $\lambda = 0$ and adjusting β adaptively

Conditional Infomax Feature Extraction [Lin and Tang, 2006]

- Correlated feature is useful if the correlation within classes is stronger than the overall correlation
- Correlation does not imply redundancy! [Guyon et al. 2006]

$$score(f_k) = I(f_k;Y) - \sum_{f_j \in \mathcal{S}} I(f_j;f_k) + \sum_{f_j \in \mathcal{S}} I(f_j;f_k|Y)$$

Correlation with selected features

Correlation with selected features within classes

- It is also a special case of the linear function by $\beta = \lambda = 1$

Examples of nonlinear $g(*)$

- Conditional Mutual Information Maximization [Fleuret, 2004]

$$J_{CMIM}(X_k) = I(X_k;Y) - \max_{X_j \in \mathcal{S}} [I(X_j;X_k) - I(X_j;X_k|Y)]$$

- Information Fragments [Vidal-Naquet and Ullman, 2003]

$$J_{IF}(X_k) = \min_{X_j \in \mathcal{S}} [I(X_j X_k;Y) - I(X_j;Y)]$$

Examples of nonlinear $g(*)$

- Interaction Capping [Jakulin, 2005]

$$J_{CMIM}(X_k) = I(X_k;Y) - \sum_{X_j \in \mathcal{S}} \max[0, I(X_j;X_k) - I(X_j;X_k|Y)]$$

- Double Input Sym Relevance [Meyer and Bontempi, 2006]

$$J_{DISR}(X_k) = \sum_{X_j \in \mathcal{S}} \frac{I(X_j X_k;Y)}{H(X_j X_k Y)}$$

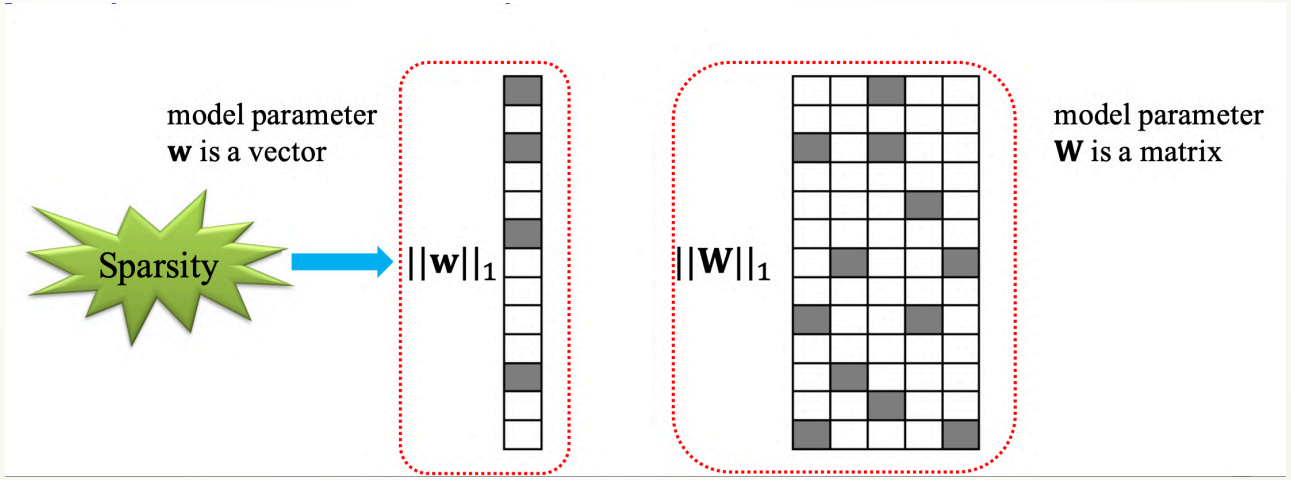
Information Theoretical based Methods - Summary

- Other information theoretical based methods
  - Fast Correlation Based Filter [Yu and Liu, 2004]
  - Interaction Gain Feature Selection [El Akadi et al. 2008]
  - Conditional MIFS [Cheng et al. 2011]…
- Pros:
  - Can handle both feature relevance and redundancy
  - Selected features can be generalized for subsequent learning tasks
- Cons:
  - Most algorithms can only work in a supervised scenario
  - Can only handle discretized data

Sparse learning based methods

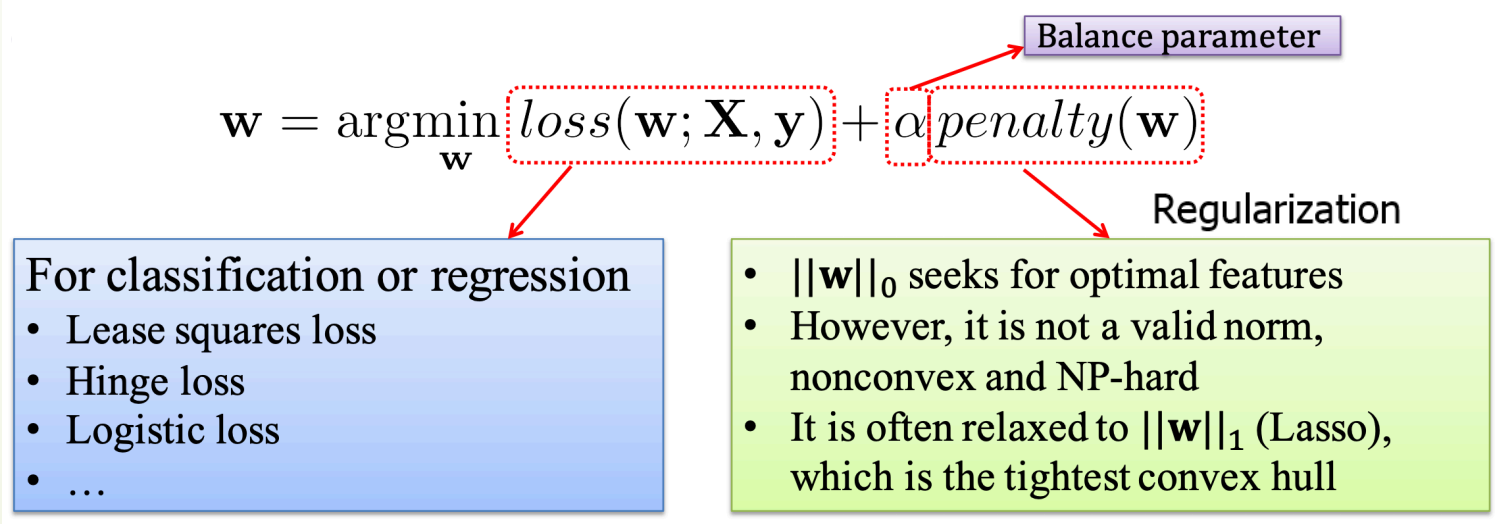
What is Feature Sparsity?

- The model parameters in many data mining tasks can be represented as a vector $\boldsymbol{w}$ or a matrix $\boldsymbol{W}$
- Sparsity indicates that many elements in $\boldsymbol{w}$ and $\boldsymbol{W}$ are small or exactly zero



Sparse Learning Methods - A General Framework

- Let us start from the binary classification or the univariate regression problem
- Let $\boldsymbol{w}$ denote the model parameter (a.k.a. feature coefficient), it can be obtained by solving the following



Lasso [Tibshirani, 1996]

-norm regularization on weight

Lasso [Tibshirani, 1996]

- Based on l-norm regularization on weight

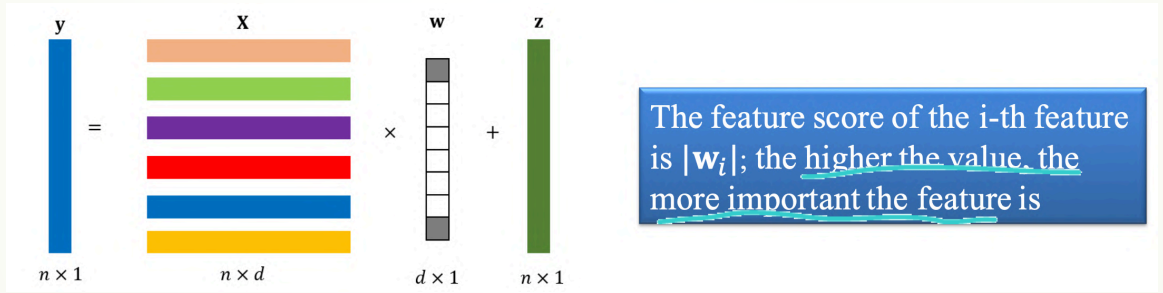
$$\min_{\mathbf{w}} loss(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \|\mathbf{w}\|_1$$

Lasso [Tibshirani, 1996]

- Based on l-norm regularization on weight

$$\min_{\mathbf{w}} loss(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \|\mathbf{w}\|_1$$

- In the case of least square loss with offset value, it looks like this …

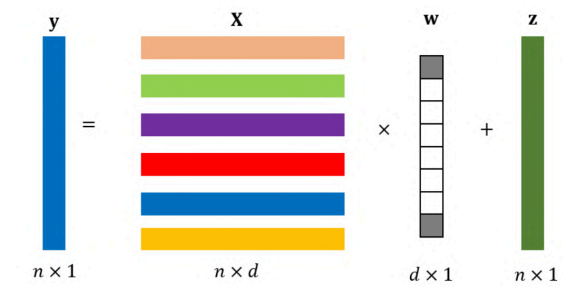


Lasso [Tibshirani, 1996]

- Based on l-norm regularization on weight

$$\min_{\mathbf{w}} loss(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \|\mathbf{w}\|_1$$

- In the case of least square loss with offset value, it looks like this …

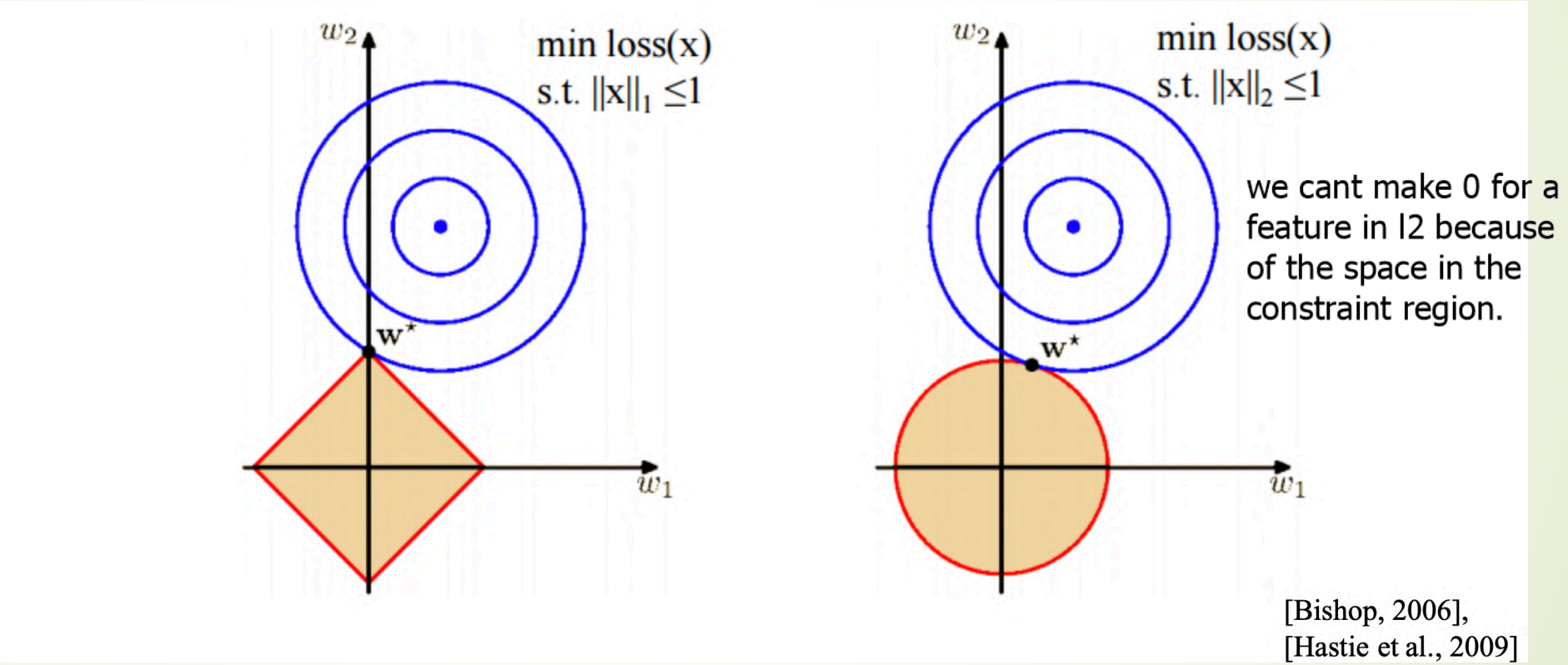


The feature score of the i-th feature is $|\mathbf{w}_i|$; the higher the value, the more important the feature is

- It is also equivalent to the following model

$$\min_{\mathbf{w}} loss(\mathbf{w}; \mathbf{X}, \mathbf{y}) \text{ s.t. } \|\mathbf{w}\| \leq t$$

Why l-norm Induces Sparsity?



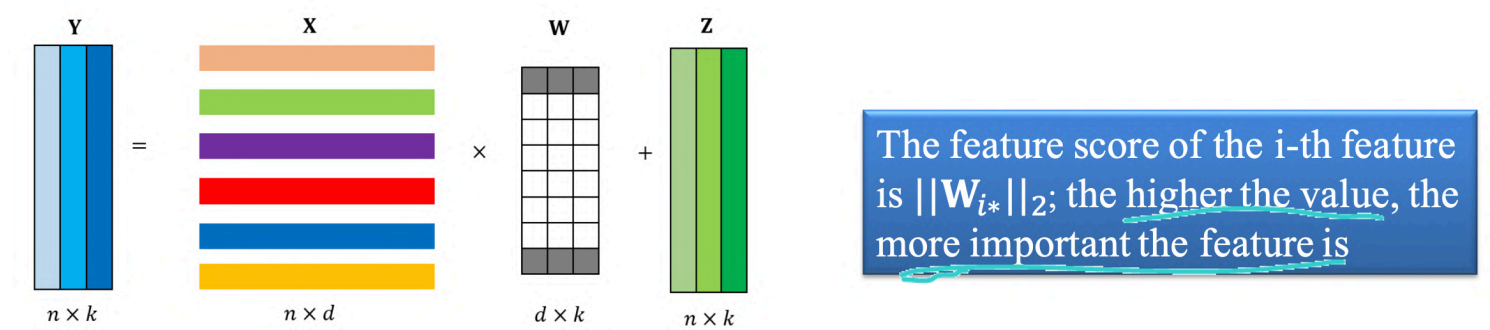
[Bishop, 2006],
[Hastie et al., 2009]

Extension to Multi-Class or Multi-Variate Problems

- Require feature selection results to be consistent across multiple targets in multi-class classification or multi-variate regression

$$\min_{\mathbf{W}} loss(\mathbf{W}; \mathbf{X}, \mathbf{Y}) + \alpha \|\mathbf{W}\|_{2,1}$$

- $\|W\|$ achieves joint feature sparsity across multiple targets
- In the case of least square loss with offset, it looks like this

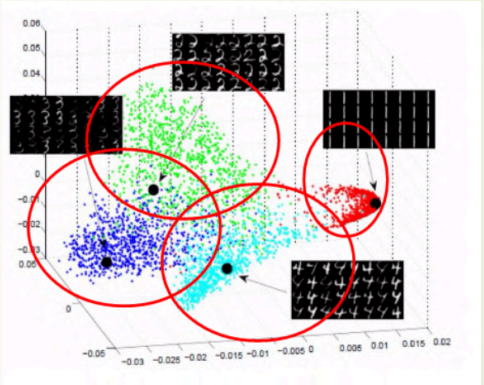


The feature score of the i-th feature is $||\mathbf{W}_{i*}||_2$; the higher the value, the more important the feature is

Unsupervised Sparse Learning based Feature Selection

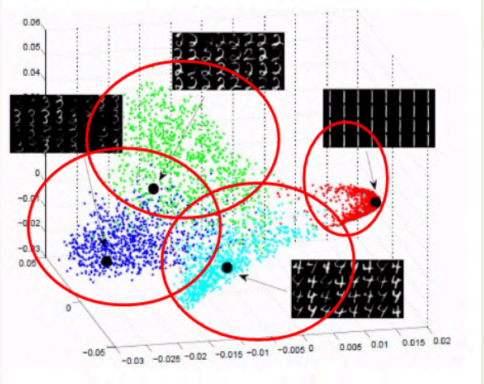
Unsupervised Sparse Learning based Feature Selection

- Without class labels, we attempt to find discriminative features that can preserve data clustering structure



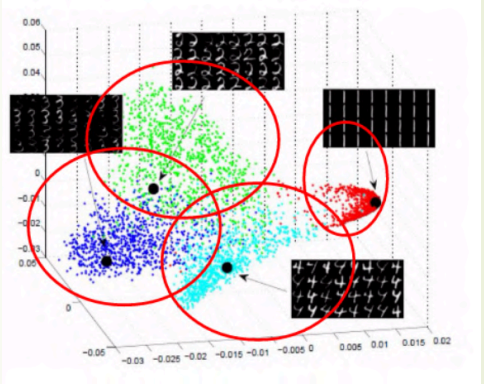
Unsupervised Sparse Learning based Feature Selection

- Without class labels, we attempt to find discriminative features that can preserve data clustering structure
- There are two options
  - Obtain clusters and then perform FS (e.g., MCFS)
  - Embed FS into clustering (e.g., NDFS)



Unsupervised Sparse Learning based Feature Selection

- Without class labels, we attempt to find discriminative features that can preserve data clustering structure
- There are two options
  - Obtain clusters and then perform FS (e.g., MCFS)
  - Embed FS into clustering (e.g., NDFS)



- The second option is preferred as not all features are useful to find clustering structure

go back and forth in option 2

Type 1	Data → Clustering Structure → Learning Model	Typical methods: MCFS, MRFS, SPFS, FSSL…
Type 2	Data → Clustering Structure → Learning Model	Typical methods: NDFS, JELSR, RUFS, EUFS…

Multi-Cluster Feature Selection (MCFS) [Cai et al., 2011]

Multi-Cluster Feature Selection (MCFS) [Cai et al., 2011]

- Basic idea: the selected features should preserve cluster structure

Multi-Cluster Feature Selection (MCFS) [Cai et al., 2011]

- Basic idea: the selected features should preserve cluster structure
- Step 1: spectral clustering to find intrinsic cluster structure

$$\mathbf{S}_{ij} = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}} \Rightarrow \mathbf{Le} = \lambda \mathbf{De}$$

intrinsic cluster indicator vector

Multi-Cluster Feature Selection (MCFS) [Cai et al., 2011]

- Basic idea: the selected features should preserve cluster structure
- Step 1: spectral clustering to find intrinsic cluster structure

$$\mathbf{S}_{ij} = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}} \Rightarrow \mathbf{Le} = \lambda \mathbf{De}$$

$\mathbf{e}$: intrinsic cluster indicator vector

- Step 2: perform Lasso on each cluster:

$$\min_{w_i} ||\mathbf{X}\mathbf{w}_i - \mathbf{e}_i||_2^2 + \alpha||\mathbf{w}_i||_1$$

Multi-Cluster Feature Selection (MCFS) [Cai et al., 2011]

- Basic idea: the selected features should preserve cluster structure
- Step 1: spectral clustering to find intrinsic cluster structure

$$\mathbf{S}_{ij} = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}} \Rightarrow \mathbf{Le} = \lambda \mathbf{De}$$

intrinsic cluster indicator vector

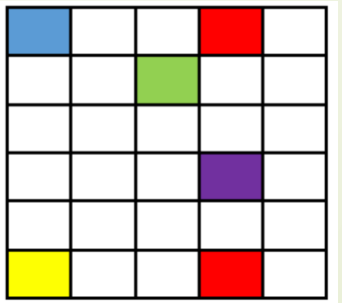
- Step 2: perform Lasso on each cluster:

$$\min_{w_i} ||\mathbf{X}\mathbf{w}_i - \mathbf{e}_i||_2^2 + \alpha||\mathbf{w}_i||_1$$

- Step 3: combine multiple feature coefficient together and get feature score

$$MCFS(j) = \max_i |\mathbf{W}_{ji}|$$

The higher the feature score, the more important the feature is



Nonnegative Unsupervised Feature Selection (NDFS) [Li et al., 2012]

- Perform spectral clustering and feature selection jointly
- The weighted cluster indicator matrix G can be obtained by using nonnegative spectral analysis:

$$\min_{\mathbf{G}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{S}(i,j)||\frac{\mathbf{G}_{i*}}{\sqrt{\mathbf{D}(i,i)}}-\frac{\mathbf{G}_{j*}}{\sqrt{\mathbf{D}(j,j)}}||_2^2 = tr(\mathbf{GLG}')$$

$$\mathbf{G}'\mathbf{G}=\mathbf{I}, \mathbf{G}\geq 0$$

Diagonal matrix obtained from RBF kernel similarity matrix **S**

- Embed cluster matrix into feature selection

$$\min_{\mathbf{G},\mathbf{W}} tr(\mathbf{GLG}')+\beta(||\mathbf{XW}-\mathbf{G}||_F^2+\alpha||\mathbf{W}||_{2,1})$$

$$\text{s.t.} \quad \mathbf{G}'\mathbf{G}=\mathbf{I}, \mathbf{G}\geq 0$$

- Feature score obtained from $\boldsymbol{W}$ (higher the value, the better)

Sparse Learning based Methods - Summary

- Other sparse learning based methods
  - Multi-label informed feature selection [Jian et al. 2016]
  - Embedded unsupervised feature selection [Wang et al. 2015]
  - Adaptive structure learning feature selection [Du et al. 2015]

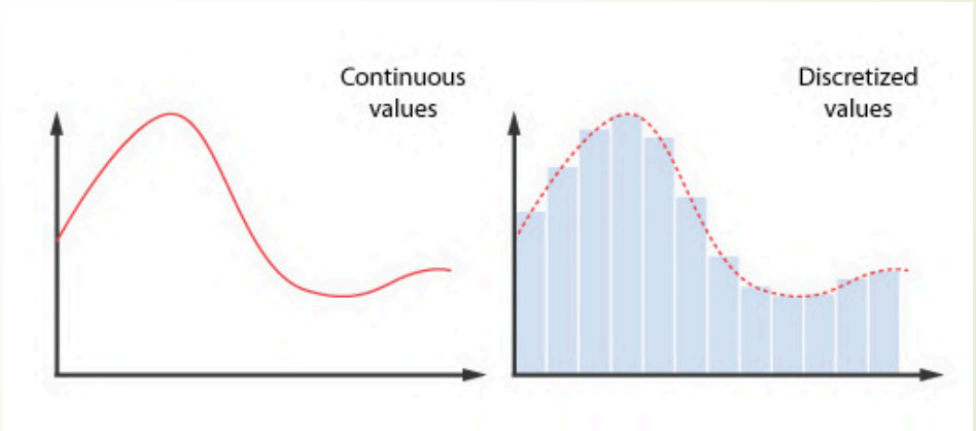
- Pros:
  - Obtain good performance for the underlying learning method
  - Good model interpretability

- Cons:
  - The selected features may not be suitable for other tasks
  - Require solving non-smooth optimization problems, which is computationally expensive

Statistical methods

Statistical based Methods

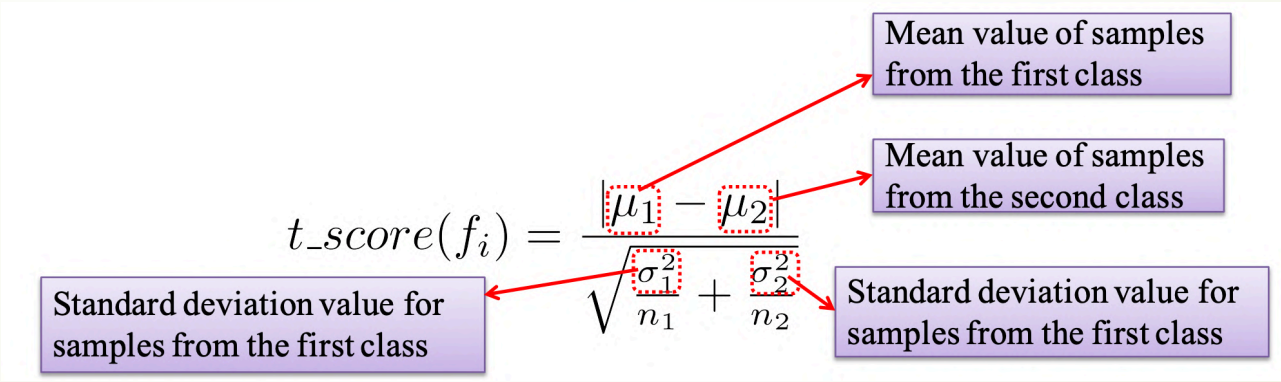
- This family of algorithms are based on different statistical methods to measure feature importance
- Most of them are filter feature selection methods
- Most algorithms evaluate features individually, so the feature redundancy is inevitably ignored
- Most algorithms can only handle discrete data, the numerical features have to be discretized first



T-Score [Davis and Sampson, 1986]

- It is used for binary classification problems
- Assess whether the feature makes the means of samples from two classes statistically significant

- The t-score of each feature f_i is



- The higher the t-score, the more important the feature is

Chi-Square Score [Liu and Setiono, 1995]

- Utilize independence test to assess whether the feature is independent of class label
- Given a feature with r values, its feature score is

$$Chi\_square\_score(f_i) = \sum_{j=1}^{r}\sum_{s=1}^{c} \frac{(n_{js} - \mu_{js})^2}{\mu_{js}}$$

instances with the j-th feature value and in class s

$$\mu_{js} = \frac{n_{*s} n_{j*}}{n}$$

instances with the j-th feature value

instances in class s

- Higher chi-square indicates that the feature is more important

Statistical based Methods - Summary

- Other statistical based methods
  - Low variance – CFS [Hall and Smith, 1999]
  - Kruskal Wallis [McKnight, 2010]…

- Pros:
  - Computational efficient
  - The selected features can be generalized to subsequent learning tasks

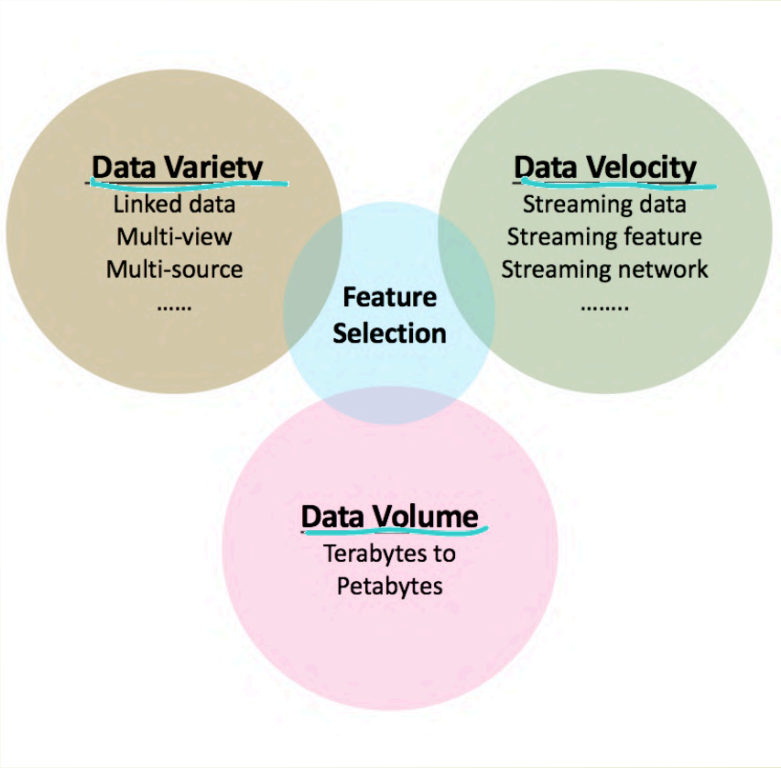
- Cons:
  - Cannot handle feature redundancy
  - Require data discretization techniques
  - Many statistical measures are not that effective in high-dim space

Other Types of Methods

- Reconstruction based Feature Selection
  - Minimize reconstruction error of data with selected features
  - Reconstruction function can be both linear and nonlinear
- Hybrid Feature Selection
  - Construct a set of different feature selection results
  - Aggregate different outputs into a consensus result

Feature Selection Issues

- Recent popularity of big data presents challenges to conventional FS
  - Streaming data and features
  - Heterogeneous data
  - Structures between features
  - Volume of collected data



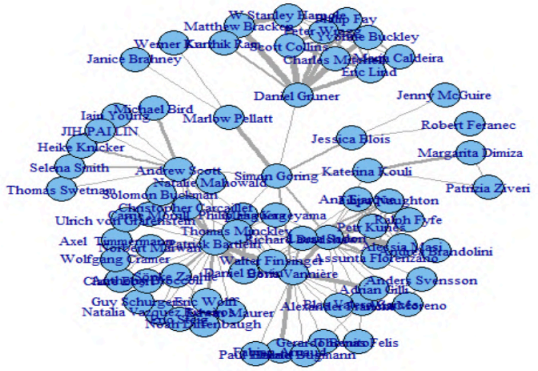
Some other methods of FS

- **Feature Selection with Structured Features**
- **Feature Selection with Heterogeneous Data**
- **Multi-Source Feature Selection**

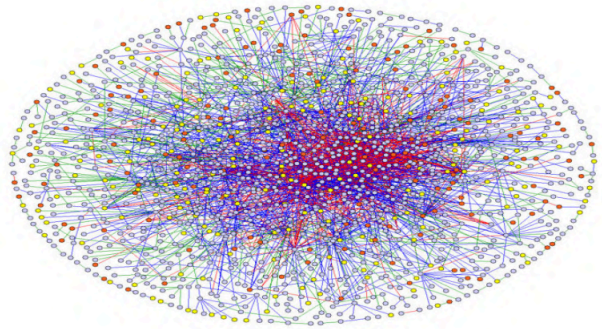
Feature Selection with Heterogeneous Data

Feature Selection with Heterogeneous Data

Social network



Coauthor network



Gene network

Transportation network

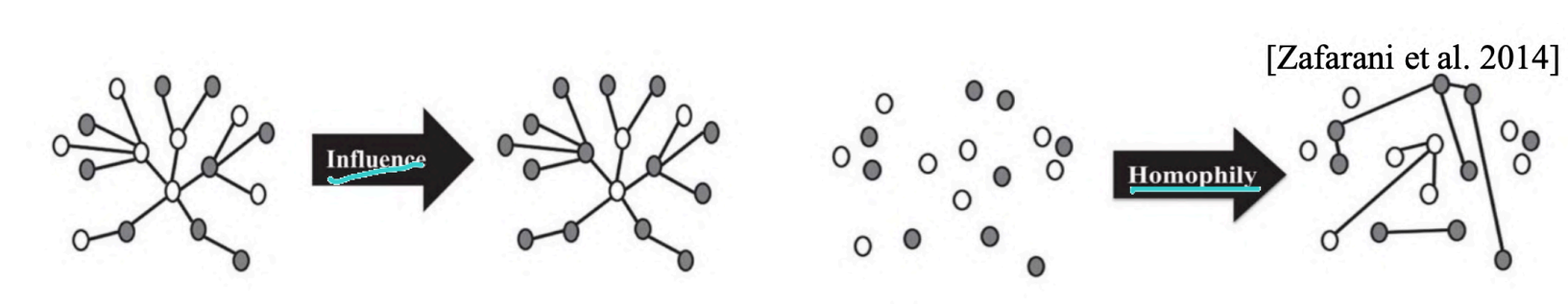
Feature Selection with Heterogeneous Data

- Traditional feature selection algorithms are for a single source and are heavily based on the data i.i.d. assumption
- Heterogeneous data is prevalent and is often not i.i.d.
  - Networked data
  - Data from multiple sources
- It is necessary to leverage feature selection to fuse multiple data sources synergistically

Why Performing Feature Selection with Networks?

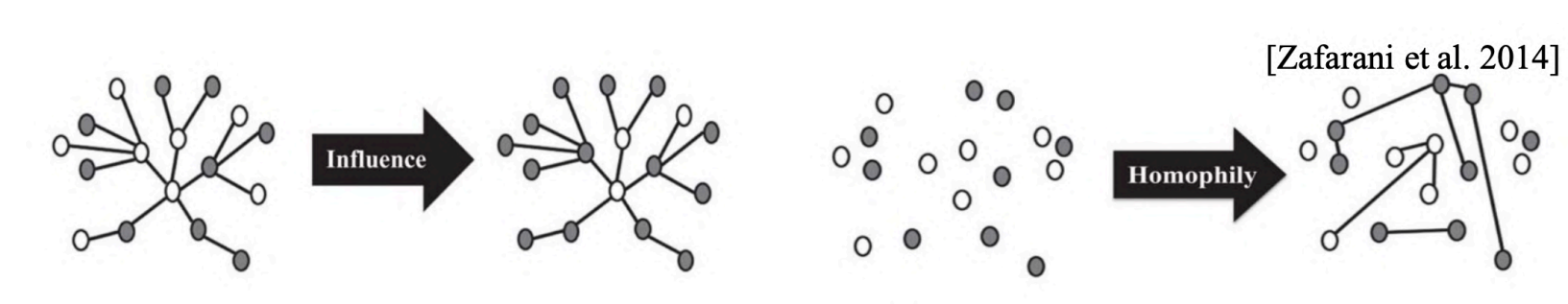
Why Performing Feature Selection with Networks?

- Social Influence & Homophily: node features and network are inherently correlated



Why Performing Feature Selection with Networks?

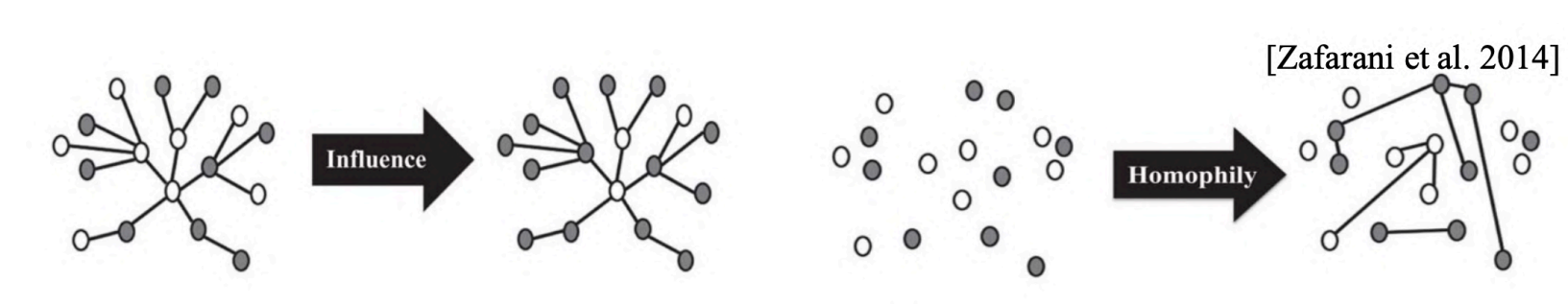
- Social Influence & Homophily: node features and network are inherently correlated



- Many learning tasks are enhanced by modeling the correlation

Why Performing Feature Selection with Networks?

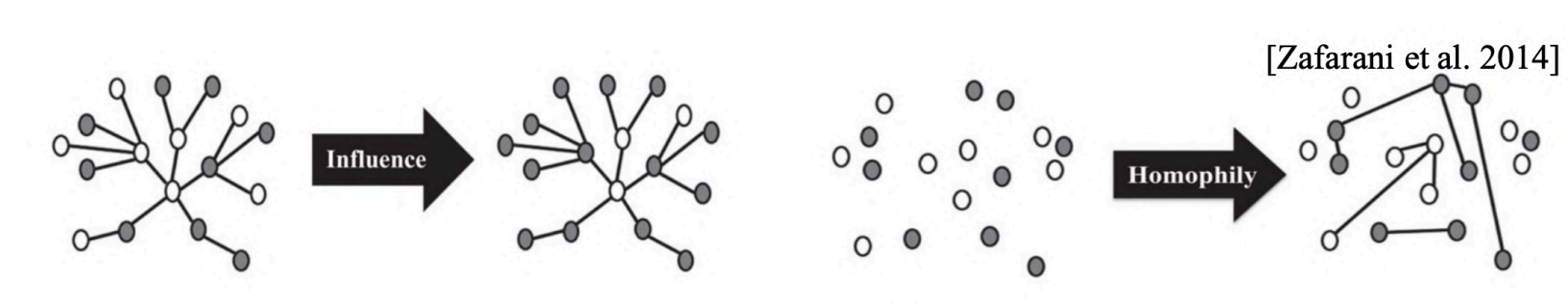
- Social Influence & Homophily: node features and network are inherently correlated



- Many learning tasks are enhanced by modeling the correlation
  - Community detection
  - Anomaly detection
  - Collective classification

Why Performing Feature Selection with Networks?

- Social Influence & Homophily: node features and network are inherently correlated



- Many learning tasks are enhanced by modeling the correlation
  - Community detection
  - Anomaly detection
  - Collective classification

- But not all features are hinged with the network structure

Challenges of Feature Selection with Networked Data

Challenges of Feature Selection with Networked Data

- Feature selection on networked data faces unique challenges
  - How to model link information **there is link in network**
  - How to fuse heterogeneous information sources
  - Label information is costly to obtain

Challenges of Feature Selection with Networked Data

- Feature selection on networked data faces unique challenges
  - How to model link information
  - How to fuse heterogeneous information sources
  - Label information is costly to obtain
- Unique properties from network and features of instances bring more challenges

Feature Selection on Networks (FSNet) [Gu and Han 2011]

- Use a linear classifier to capture the relationship between content information X and class labels Y

$$\min_{\mathbf{W}} \|\mathbf{XW}-\mathbf{Y}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{W}\|_F^2$$

Joint feature sparsity

Avoid overfitting

- *F:* Frobenius norm
- *2,1:* sum of the Euclidean norms of the columns of the matrix

- Employ graph regularization to model links

$$tr(\mathbf{W}'\mathbf{X}'\mathbf{L}\mathbf{X}\mathbf{W})$$

undirected network: $\mathbf{L}=\mathbf{D}-\mathbf{A}$ — Adjacency matrix

directed network: $\mathbf{L}=\mathbf{\Pi}-\frac{1}{2}(\mathbf{\Pi P}+\mathbf{P}'\mathbf{\Pi})$

P: transition matrix of random walk
Π: stationary distribution

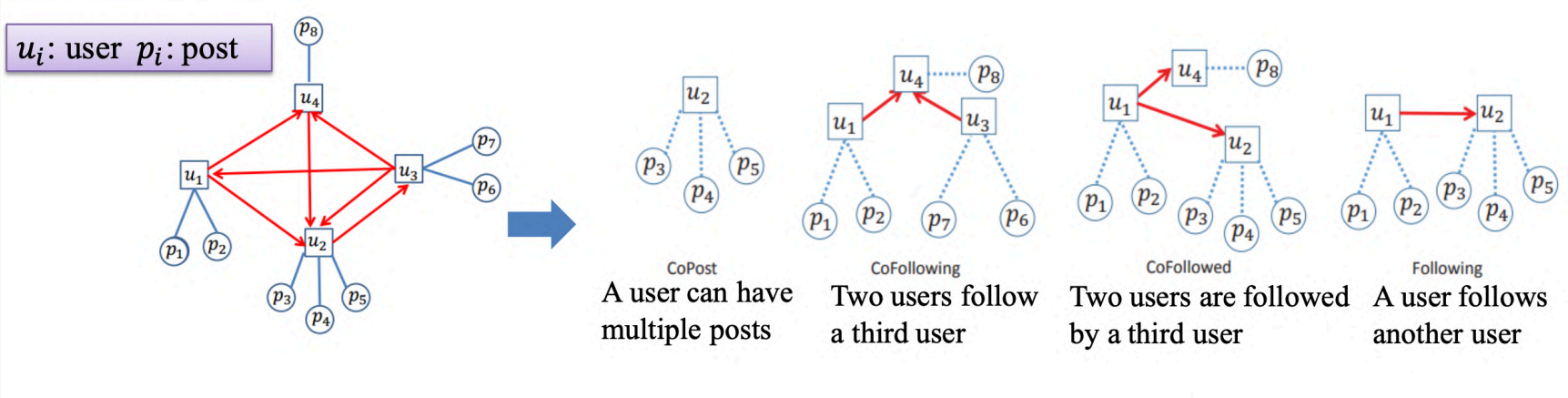
- Objective function of FSNet

$$\min_{\mathbf{W}} \|\mathbf{XW}-\mathbf{Y}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{W}\|_F^2 + \gamma tr(\mathbf{W}'\mathbf{X}'\mathbf{L}\mathbf{X}\mathbf{W})$$

Feature scores are obtained from matrix **W**

Linked Feature Selection (LinkedFS) [Tang and Liu 2012]

- Investigate feature selection on social media data with various types of social relations: four basic types



- These social relations are supported by social theories (Homophily and Social Influence)

Linked Feature Selection (LinkedFS) [Tang and Liu, 2012]

- For CoPost hypothesis
  - Posts by the same user are likely to be of similar topics

- Feature selection with the CoPost hypothesis

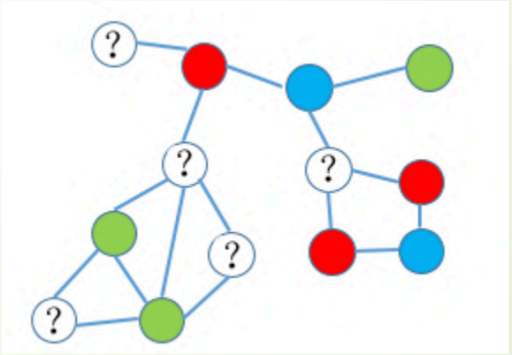
$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{Y}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta \sum_{u \in \mathbf{u}} \sum_{\{p_i, p_j\} \in \mathbf{p}_u} \|\mathbf{X}(i,:)\mathbf{W} - \mathbf{X}(j,:)\mathbf{W}\|_2^2$$

CoPost hypothesis

CoPost relations

Personalized Feature Selection [Li et al 2017]

- Content information of nodes are highly idiosyncratic
  - E.g., blogs, posts and images of different users could be diverse and with different social foci
  - E.g., the same content could convey different meanings: "The house prices are getting higher and higher!": Seller – *positive*, Buyer – *negative* [Wang et al 2016]
- But nodes share some commonality to some extent
- How to tackle the idiosyncrasy and commonality of node features for learning such as node classification?



Personalized Feature Selection [Li et al 2017]

- To find personalized features, we attempt to achieve feature sparsity within each local feature weight

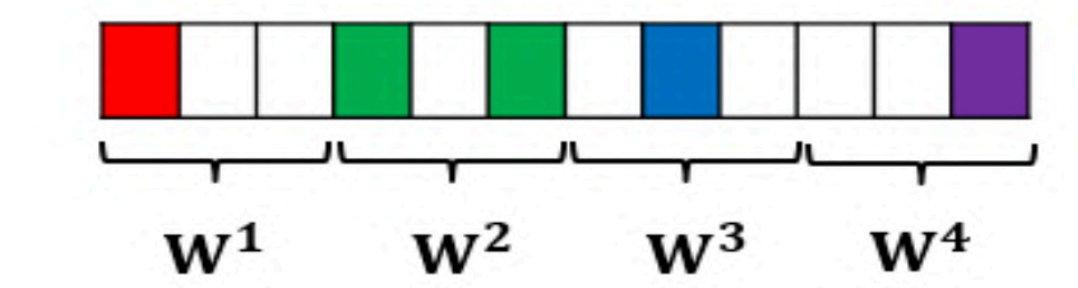
$$\min_{\tilde{\mathbf{W}},\mathbf{W}^i} \sum_{i=1}^{n} \|\mathbf{x}_i(\tilde{\mathbf{W}} + \mathbf{W}^i) - \mathbf{y}_i\|_2^2 + \beta \sum_{i=1}^{n} \|\mathbf{W}^i\|_{2,1}^2 + \gamma \|\tilde{\mathbf{W}}\|_{2,1}$$

global feature weight for all nodes

local feature weight for the i-th node

exclusive group lasso

- The exclusive group lasso encourages intra-group competition but discourages inter-group competition



Personalized Feature Selection [Li et al 2017]

- We cluster local weights into groups to reduce overfitting

$$\min_{\mathbf{W}} \sum_{i,j=1}^{n} \mathbf{A}_{i,j} \|\mathbf{W}^i - \mathbf{W}^j\|_F$$

Make connected nodes borrow strength from each other

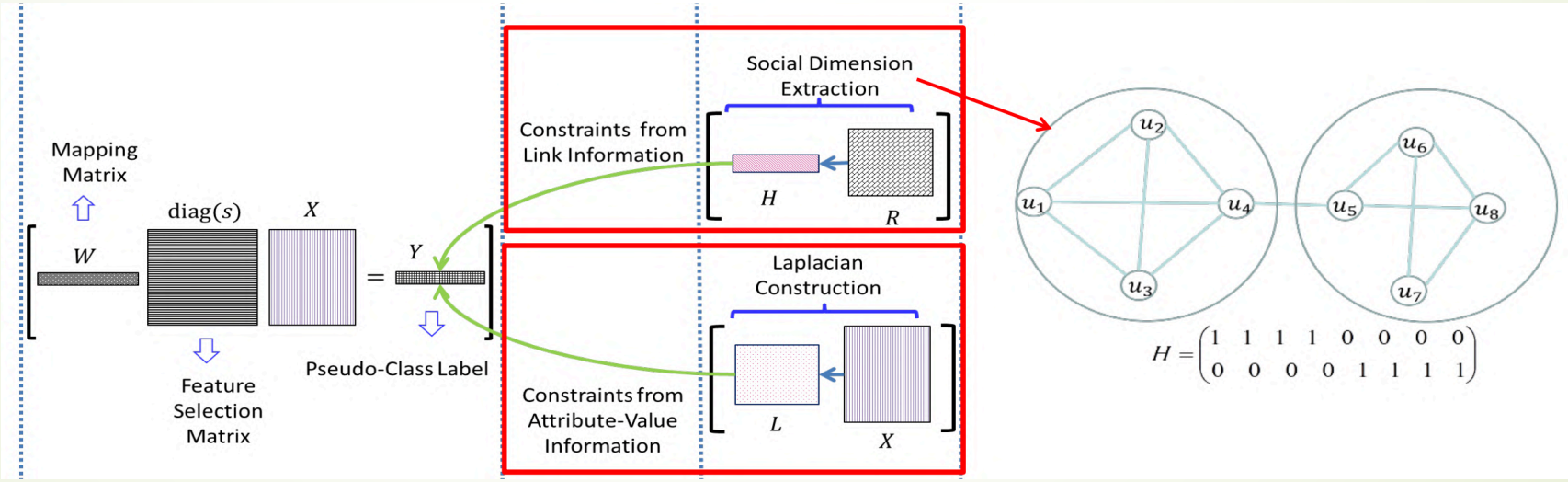
- The objective function

$$\min_{\tilde{\mathbf{W}},\mathbf{W}^i} J(\tilde{\mathbf{W}}, \mathbf{W}^i) = \sum_{i=1}^{n} \|\mathbf{x}_i(\tilde{\mathbf{W}} + \mathbf{W}^i) - \mathbf{y}_i\|_2^2$$

$$+\alpha \sum_{i,j=1}^{n} \mathbf{A}(i,j)\|\mathbf{W}^i - \mathbf{W}^j\|_F + \beta \sum_{i=1}^{n} \|\mathbf{W}^i\|_{2,1}^2 + \gamma \|\tilde{\mathbf{W}}\|_{2,1}$$

Linked Unsupervised Feature Selection (LUFS) [Tang and Liu 2012]

- Data is often unlabeled in networked data
- No explicit definition of feature relevance
- Fortunately, links provide additional constraints



Linked Unsupervised Feature Selection (LUFS) [Tang and Liu 2012]

- Obtain within, between and total social dimension scatter matrices S_w, S_b, S_t

$$\mathbf{S}_w = \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{F}\mathbf{F}'\mathbf{Y}, \mathbf{S}_b = \mathbf{Y}'\mathbf{F}\mathbf{F}'\mathbf{Y}, \mathbf{S}_t = \mathbf{Y}'\mathbf{Y}$$

Weighted social dimension matrix

$$\mathbf{F} = \mathbf{H}(\mathbf{H}'\mathbf{H})^{-\frac{1}{2}}$$

- Instances are similar within social dimensions while dissimilar between social dimensions

$$\max_{\mathbf{W}} tr((\mathbf{S}_t)^{-1}\mathbf{S}_b)$$

- Similar instances in terms of their contents are more likely to share similar topics

$$\min tr(\mathbf{Y}'\mathbf{L}\mathbf{Y})$$

Obtained from content similarity matrix using RBF

Linked Unsupervised Feature Selection (LUFS) [Tang and Liu 2012]

- Optimization framework of LUFS

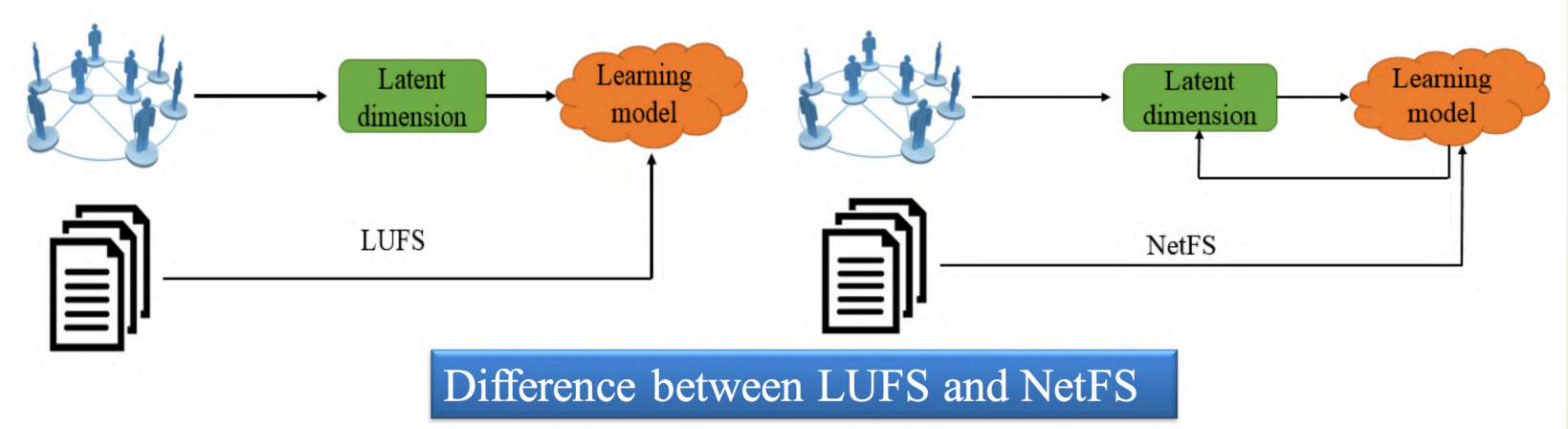
$$\min_{\mathbf{W},\mathbf{s}} \; tr(\mathbf{YLY}') - \alpha tr((\mathbf{S}_t)^{-1}\mathbf{S}_b)$$
$$\text{s.t.}\, \mathbf{s} \in \{0,1\}^d,\, \mathbf{s}'\mathbf{1} = k, \quad \leftarrow \quad \mathbf{Y} = \mathbf{W}'\text{dig}(\mathbf{s})\mathbf{X}$$
$$\|\mathbf{Y}(:,i)\|_0 = 1,\, 1 \le i \le n.$$

- Spectral relaxation on and impose $l_{2,1}$-norm regularization

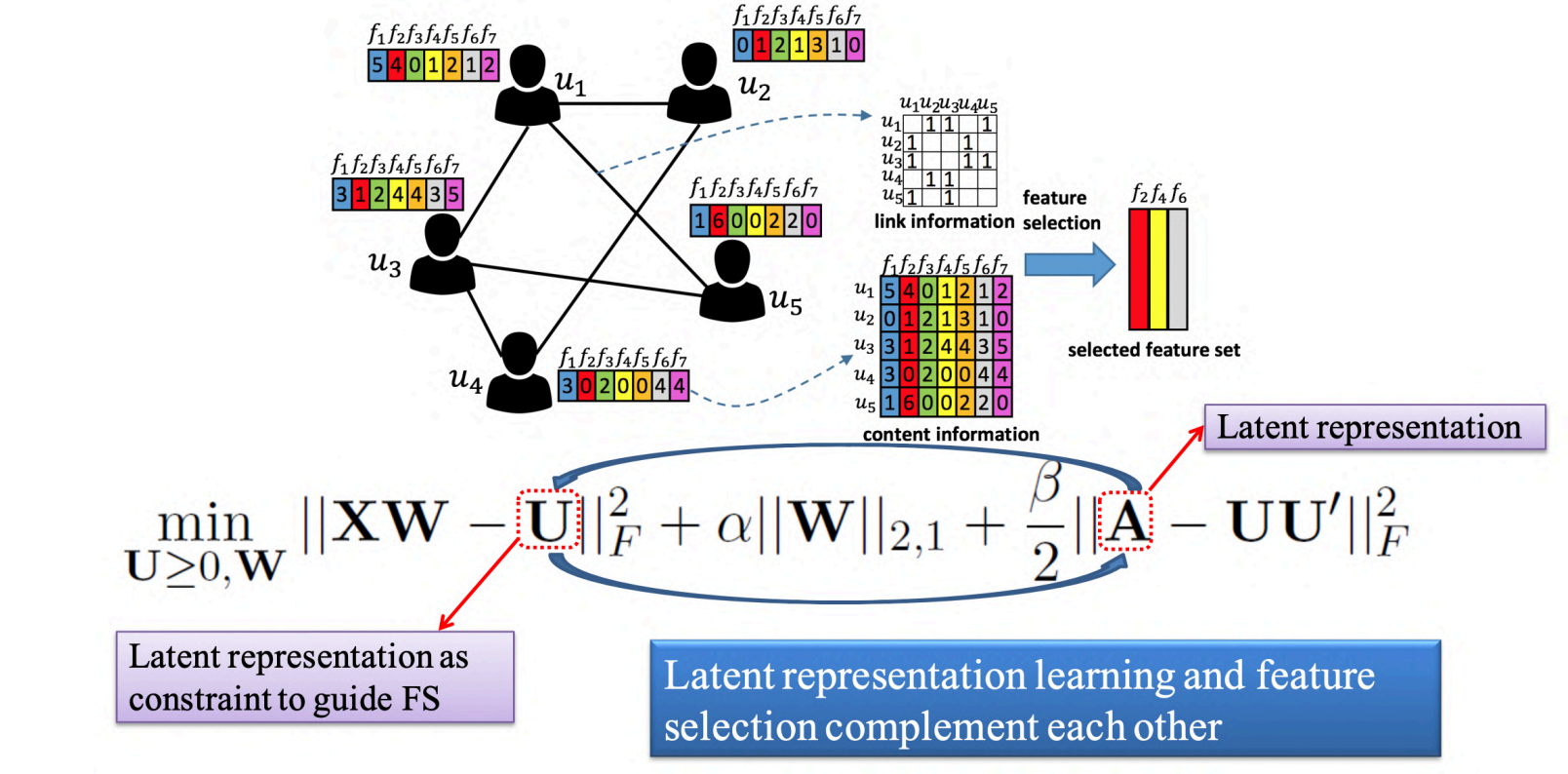
$$\min_{\mathbf{W}} tr(\mathbf{W}'(\mathbf{X}'\mathbf{LX} + \alpha\mathbf{X}'(\mathbf{I}_n - \mathbf{FF}'))\mathbf{W}) + \beta\|\mathbf{W}\|_{2,1}$$
$$\text{s.t.} \quad \mathbf{W}'(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_d)\mathbf{W} = \mathbf{I}_c$$

Robust Unsupervised FS on Networks (NetFS) [Li et al 2016]

- LUFS performs network structure modeling and feature selection separately
- NetFS embeds latent representation modeling into feature selection and is more robust to noise links



Robust Unsupervised FS on Networks (NetFS) [Li et al 2016]



Multi-Source Feature Selection

Multi-Source Feature Selection [Zhao and Liu 2008]

- Given multiple local geometric patterns in similarity matrix S_i, the global $S = \sum_{i=1}^{m} S_i$
- Geometry-dependent sample covariance matrix for the target source X_i

$$\mathbf{C} = \frac{1}{n-1}\mathbf{\Pi}\mathbf{X}_i'(\mathbf{S} - \frac{\mathbf{S}\mathbf{1}\mathbf{1}'\mathbf{S}}{\mathbf{1}'\mathbf{S}\mathbf{1}})\mathbf{X}_i\mathbf{\Pi}$$

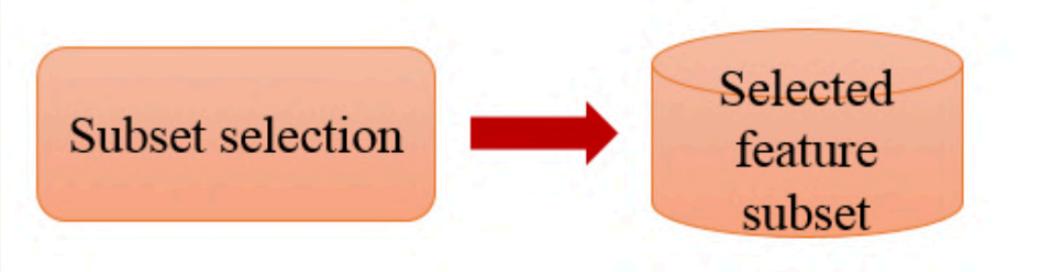
$$\mathbf{D}_{kk} = \sum_j \mathbf{S}_{kj}$$
$$\mathbf{\Pi}_{jj} = \|\mathbf{D}^{0.5}\mathbf{X}_i(:,j)\|^{-1}$$

- Two ways to obtain relevant features from
  - Sort the diagonal of C and return the features with the highest variances (consistent with global pattern)
  - Apply sparse PCA to select features that are able to retain the total variance maximally

Evaluation of Feature Selection

- Feature weighting: given a desired feature number k, rank features according to the feature scores, and then return the top-k

- Feature subset selection: directly return the obtained feature subset (cannot specify beforehand)



Evaluation of Feature Selection - Supervised

- **Supervised feature selection**

1. Divide data into training and testing set
2. Perform feature selection to obtain selected features
3. Obtain the training and testing data on the selected features
4. Feed into a classifier (e.g., SVM)
5. Obtain the classification performance on (e.g., F1, AUC)

- **The higher the classification performance, the better the selected features are**

Evaluation of Feature Selection - Unsupervised

- **Unsupervised feature selection**

1. Perform feature selection on data to obtain selected features
2. Obtain new data on the selected features
3. Perform clustering (given #m clusters)
4. Compare the obtained clustering with the ground truth
5. Obtain clustering evaluation results (e.g., Normalized Mutual Information)

- **The higher the clustering performance, the better the selected features are**

Challenges of Feature Selection

- Scalability
- Stability

Scalability Challenge: Data size

- With the growth of data size, the scalability of most feature selection algorithms is jeopardized
- Data of TB scale cannot be easily loaded into memory and limits the usage of FS algorithms
- In many cases, one pass of data is desired, the second or more pass can be impractical

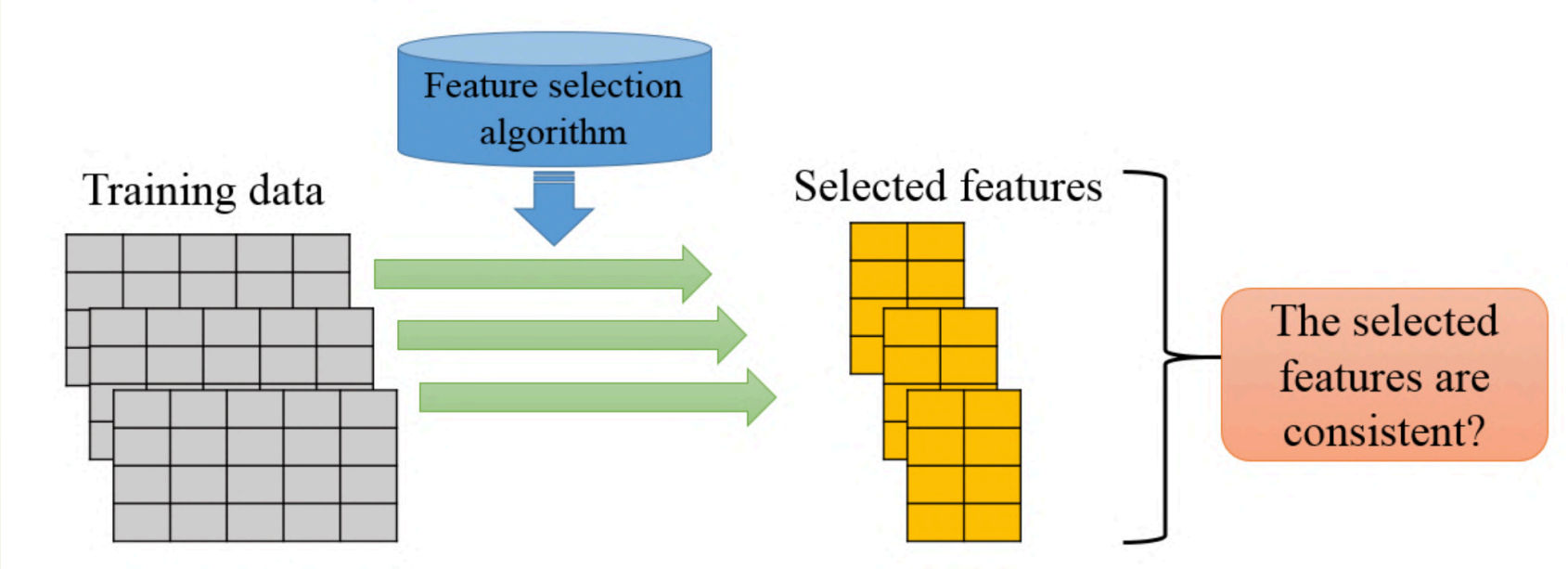
- **Potential Solution: use distributed programming framework to perform parallel feature selection**

Scalability Challenge: Feature size

- Most existing feature selection algorithms have a time complexity proportional to $O(d^2)$ or even $O(d^3)$
- Data of ultra high-dimensionality emerges
  - Text mining
  - Information retrieval
  - Brain image
- For many feature selection algorithms, efficiency deteriorates quickly with larger feature dimension
- Well-designed feature selection algorithms work in linear or sub-linear time are preferred

Stability Challange

- Stability of FS algorithms is also an important measure
- Definition: the sensitivity of a feature selection algorithm to the perturbation of training data



Achieving Stability

- Perturbation of training data in various formats
  - Addition/deletion of training samples
  - Inclusion of noisy/outlier samples
- Stability of feature selection helps domain experts be more confident with the selected features
  - Biologists would like to see the same set of genes selected each time when they obtain new data; otherwise they will not trust the algorithm
- Many feature selection algorithms suffer from low stability with small perturbation!

Model Selection: Which Set of Features to Use?

Model Selection: Which Set of Features to Use?

- We usually need to specify the number of selected features in feature weighting methods
- Finding the "optimal" number is difficult
  - A large number will increase the risk in including irrelevant and redundant features, jeopardizing learning performance
  - A small number will miss some relevant features

Model Selection: Which Set of Features to Use?

- We usually need to specify the number of selected features in feature weighting methods
- Finding the "optimal" number is difficult
  - A large number will increase the risk in including irrelevant and redundant features, jeopardizing learning performance
  - A small number will miss some relevant features
- Solution: apply heuristics such as "grid search" strategy, but performing "grid search" is very time-consuming

Model Selection: Which Set of Features to Use?

- We usually need to specify the number of selected features in feature weighting methods
- Finding the "optimal" number is difficult
  - A large number will increase the risk in including irrelevant and redundant features, jeopardizing learning performance
  - A small number will miss some relevant features
- Solution: apply heuristics such as "grid search" strategy, but performing "grid search" is very time-consuming
- Choosing the # of selected features is still an open problem

Model Selection for Unsupervised Learning

Model Selection for Unsupervised Learning

- In unsupervised feature selection, we often need to specify the number of cluster or pseudo class labels

Model Selection for Unsupervised Learning

- In unsupervised feature selection, we often need to specify the number of cluster or pseudo class labels
- However, we often have limited knowledge about the intrinsic cluster structure of data

Model Selection for Unsupervised Learning

- In unsupervised feature selection, we often need to specify the number of cluster or pseudo class labels
- However, we often have limited knowledge about the intrinsic cluster structure of data
- Different cluster number may lead to different cluster structures
  - May merge smaller clusters into a big cluster
  - May split one big cluster into multiple small clusters

Model Selection for Unsupervised Learning

- In unsupervised feature selection, we often need to specify the number of cluster or pseudo class labels
- However, we often have limited knowledge about the intrinsic cluster structure of data
- Different cluster number may lead to different cluster structures
  - May merge smaller clusters into a big cluster
  - May split one big cluster into multiple small clusters
- Lead to different feature selection results

Model Selection for Unsupervised Learning

- In unsupervised feature selection, we often need to specify the number of cluster or pseudo class labels
- However, we often have limited knowledge about the intrinsic cluster structure of data
- Different cluster number may lead to different cluster structures
  - May merge smaller clusters into a big cluster
  - May split one big cluster into multiple small clusters
- Lead to different feature selection results
- Without label information, we cannot perform cross validation

Privacy and Security Issues in Feature Selection

- Many collected data for learning are highly sensitive, e.g., medical details, census records
- Most feature selection algorithms cannot address the privacy issues
  - require privacy-preserving FS
- Feature privacy
  - Find optimal feature subset with the total privacy degree less than a given threshold
- Sample privacy (differential privacy)
  - Know all but one entry of the data, and cannot gain additional info about the entry with the output of the algorithm

Summary

Summary

- Feature selection is effective to tackle the curse of dimensionality and is essential to many data mining and machine learning problems

Summary

- Feature selection is effective to tackle the curse of dimensionality and is essential to many data mining and machine learning problems
- The objectives of feature selection include
  - Building simpler and more comprehensive models
  - Improving learning performance
  - Preparing clean and understandable data

Summary

- Feature selection is effective to tackle the curse of dimensionality and is essential to many data mining and machine learning problems
- The objectives of feature selection include
  - Building simpler and more comprehensive models
  - Improving learning performance
  - Preparing clean and understandable data
- Feature selection is equally important in the age of deep learning and big data

Summary

- Feature selection is effective to tackle the curse of dimensionality and is essential to many data mining and machine learning problems
- The objectives of feature selection include
  - Building simpler and more comprehensive models
  - Improving learning performance
  - Preparing clean and understandable data
- Feature selection is equally important in the age of deep learning and big data
- We provide a structured overview of feature selection from a data perspective
  - Feature selection for conventional data (four main categories)
  - Feature selection with structured features
  - Feature selection with heterogeneous data
  - Feature selection with multisource data

Resources

- Blog on feature selection
- More on group Lasso