Feature Selection

IFT6758 - Data Science

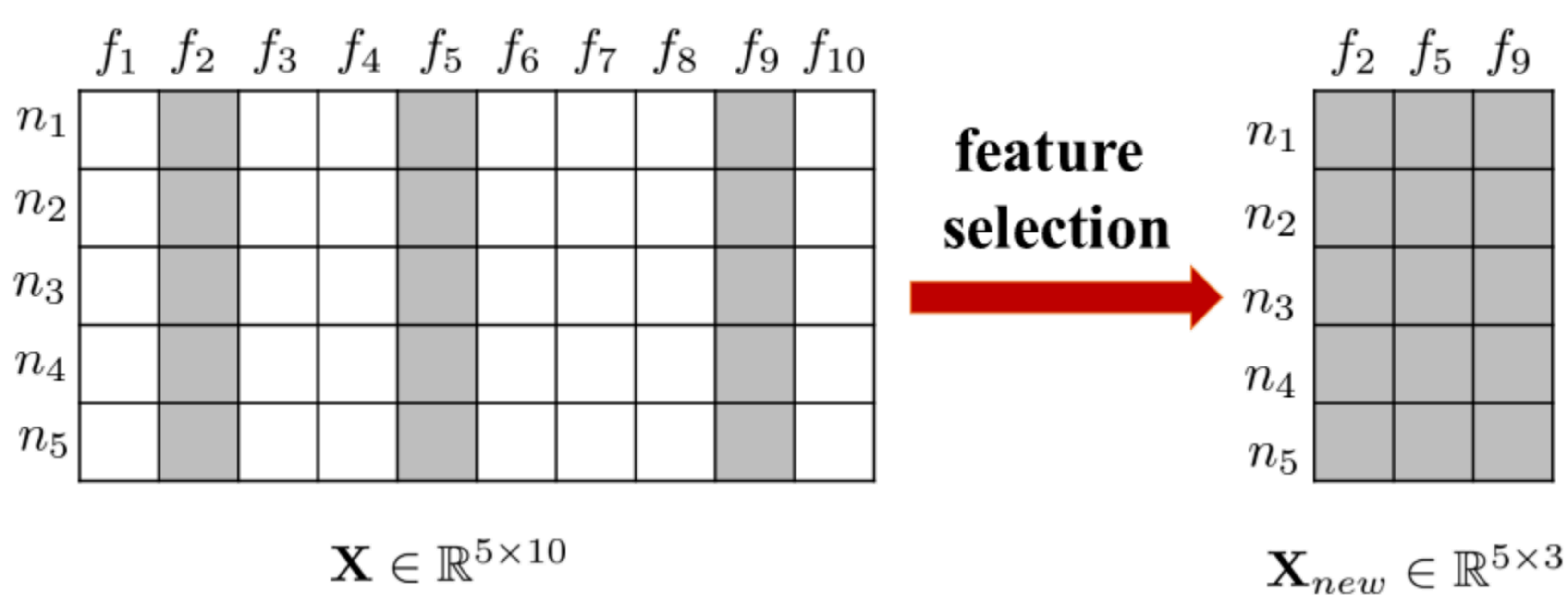
Resources:

http://www.public.asu.edu/~jundongl/tutorial/KDD17/KDD17.pdf

http://dongguo.me

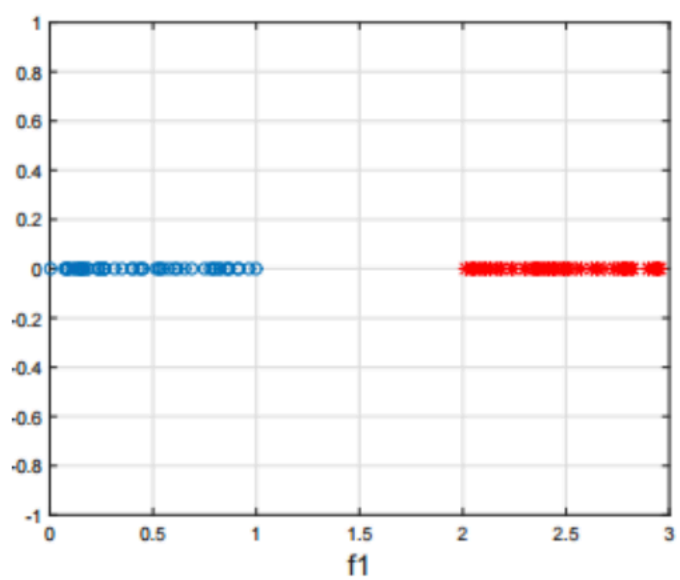
What is feature selection?

- 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

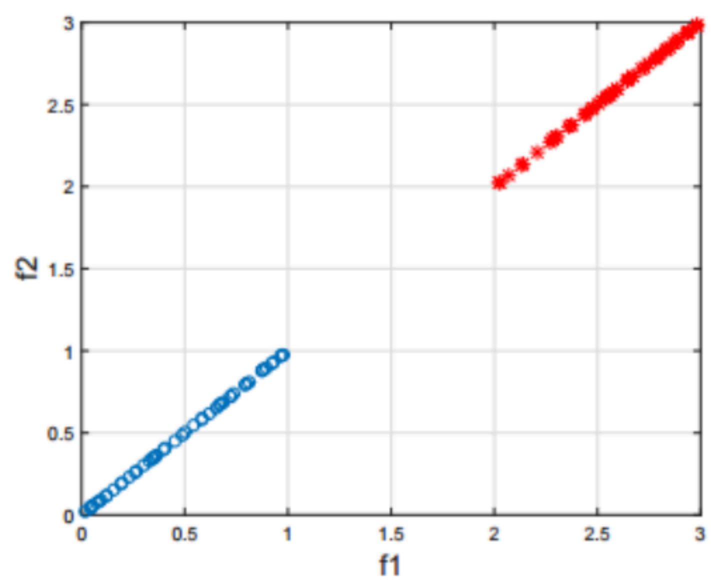


Relevant vs. Redundant features

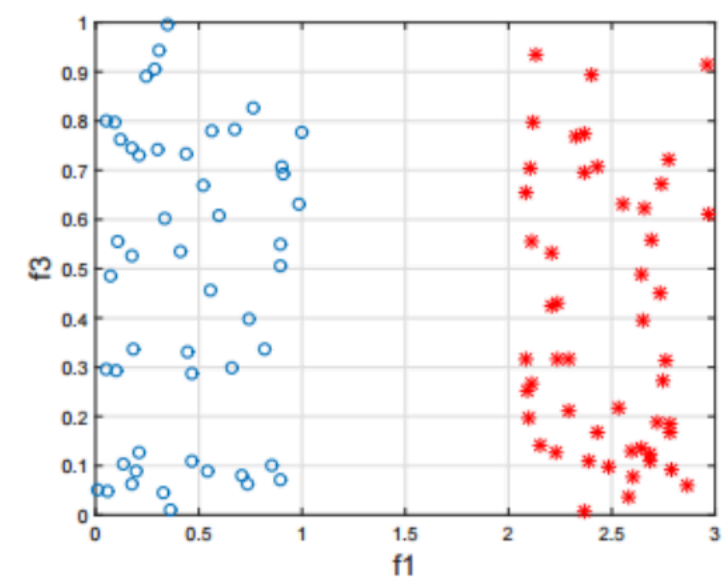
- Feature selection keeps relevant features for learning and removes redundant and irrelevant features
- For example, for a binary classification task (f1 is relevant; f2 is redundant given f1; f3 is irrelevant)



(a) relevant feature f_1



(b) redundant feature f_2



(c) irrelevant feature f_3

Feature Extraction vs. Feature Selection

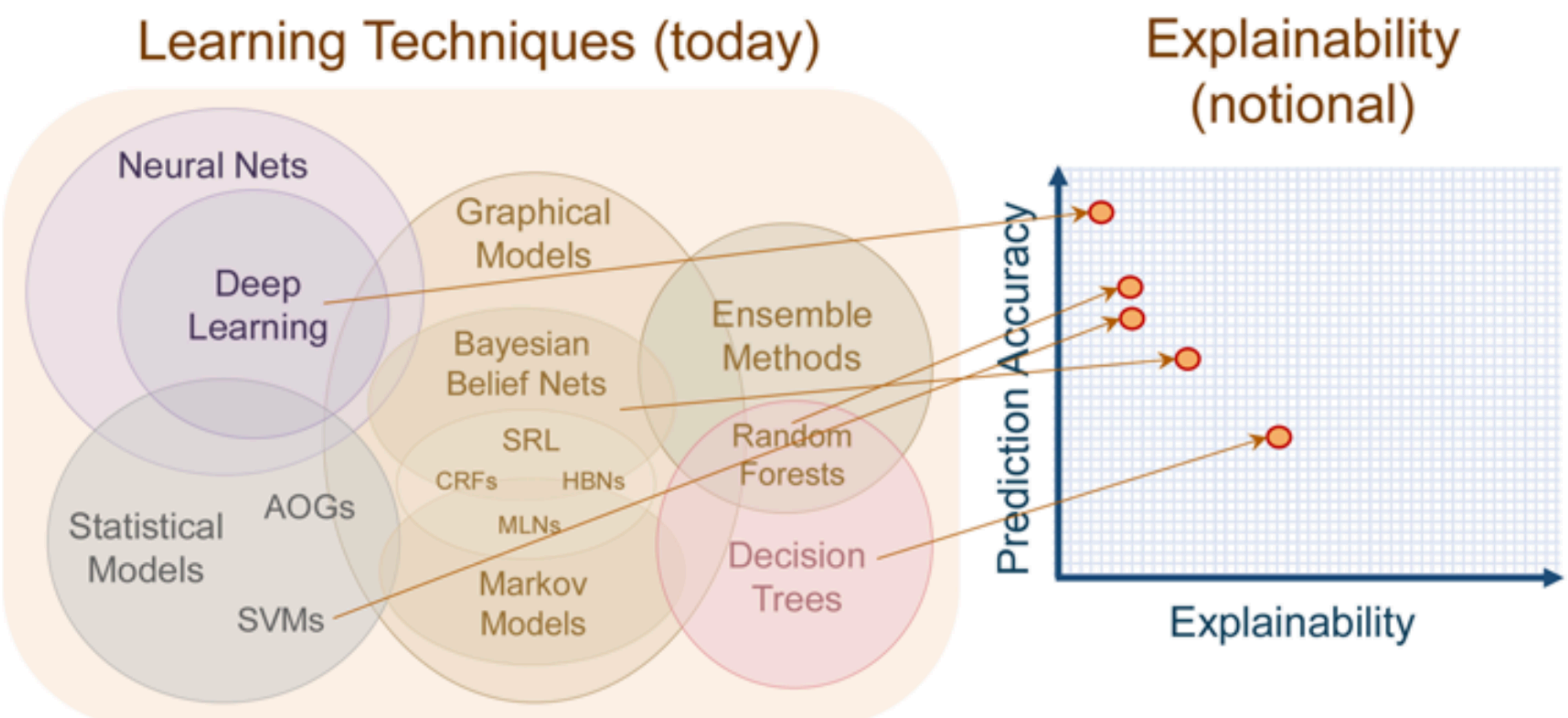
- Commonalities

  – 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**

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 !

More about this topic (Week 14)

When feature selection is important?

- Noisy data
- Lots of low frequent features
- Use multi-type features
- Too many features comparing to samples
- Complex model
- Samples in real scenario is inhomogeneous with training & test samples

Feature Selection Algorithms

- From the label perspective (whether label information is involved during the selection phase):

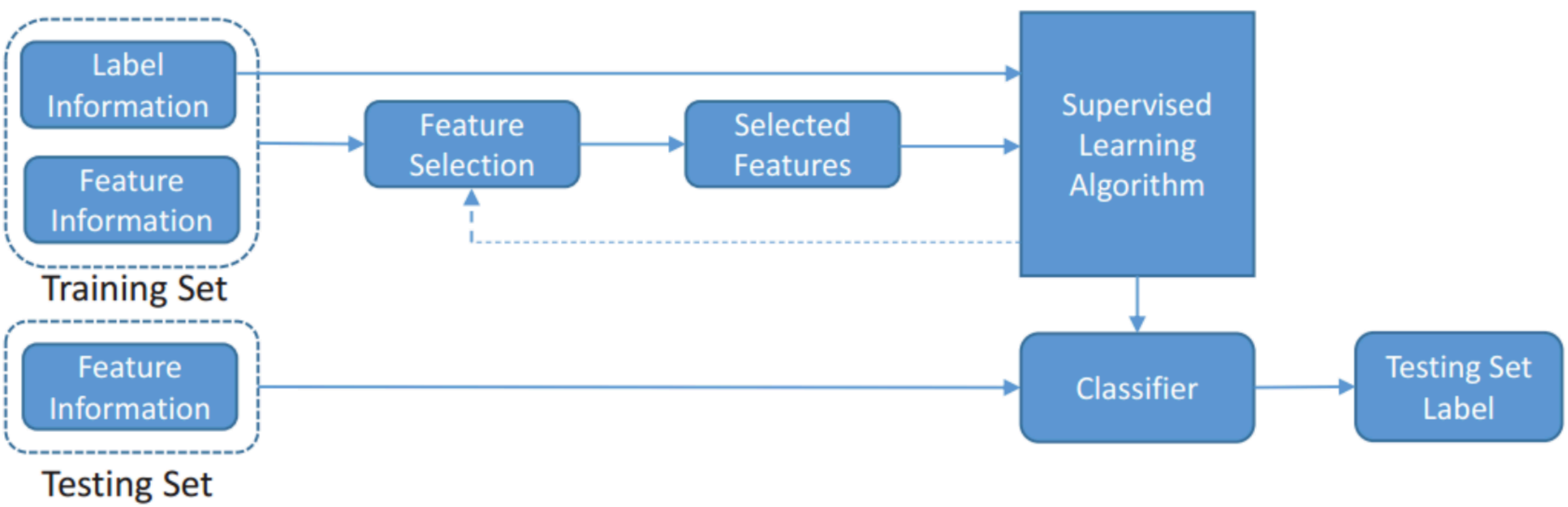
  – Supervised
  – Unsupervised
  – Semi-Supervised

- 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)

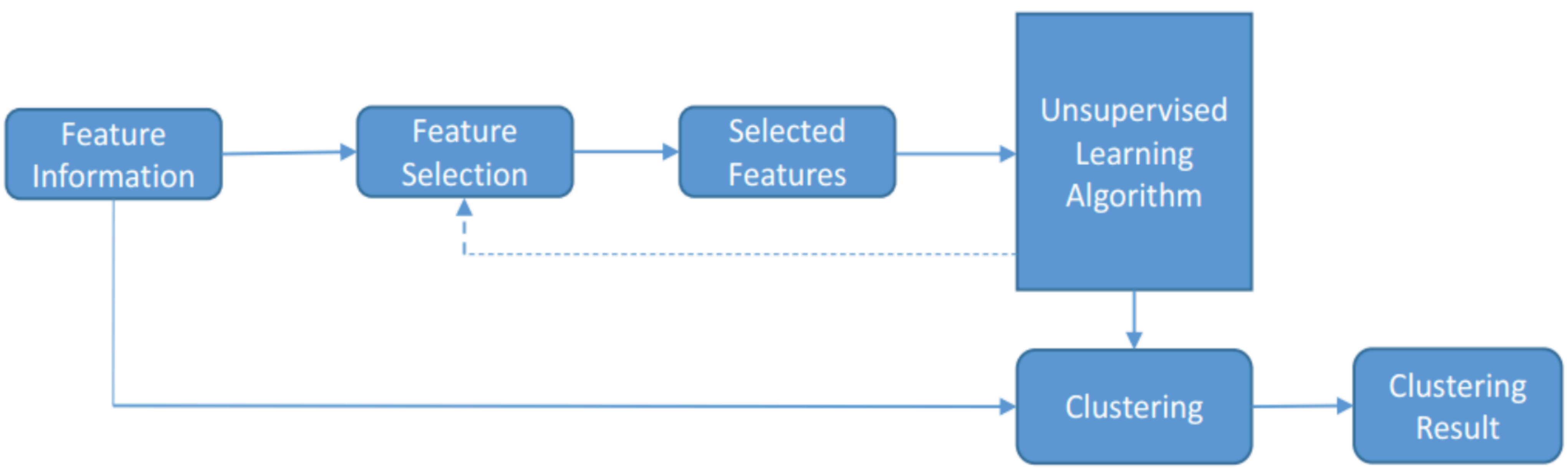


Mila

Université de Montréal

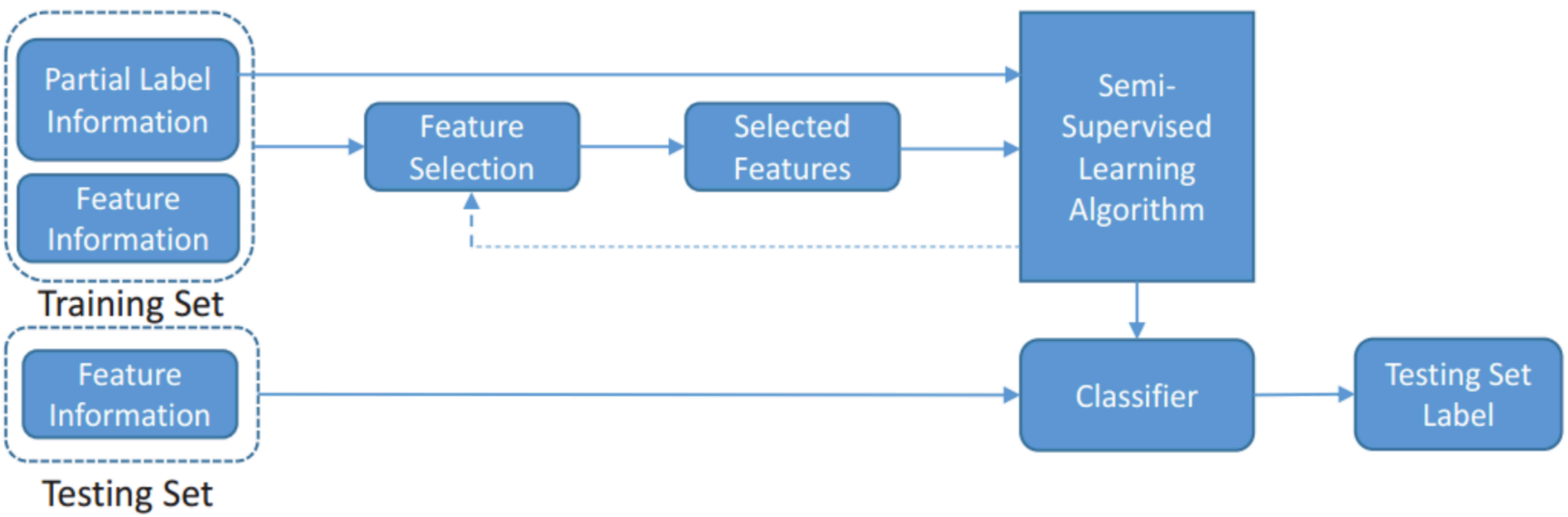
Unsupervised Feature Selection

- 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

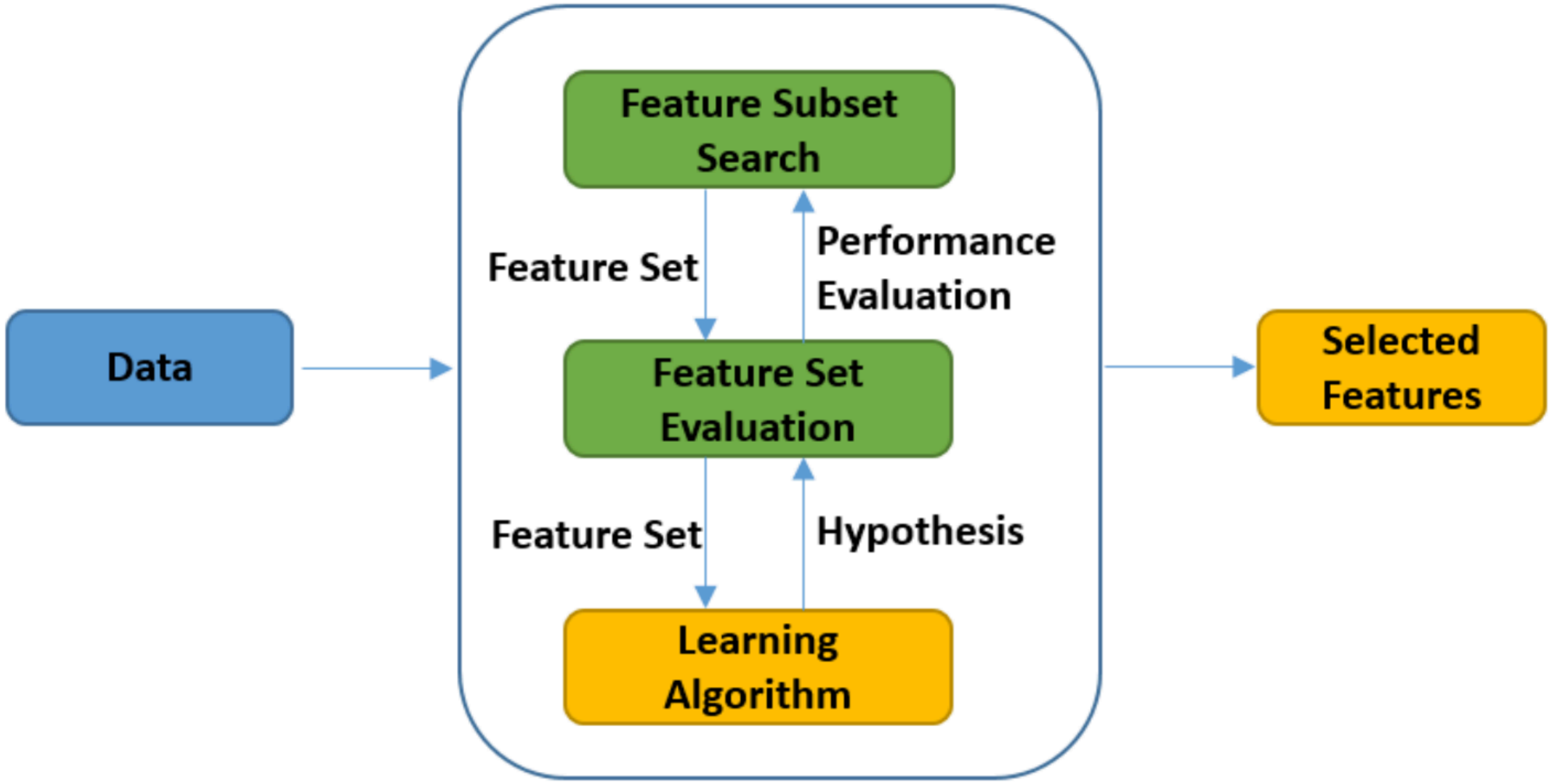


Semi-Supervised Feature Selection

- 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 : Two types: 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

Mila

Université de Montréal

Wrapper Methods

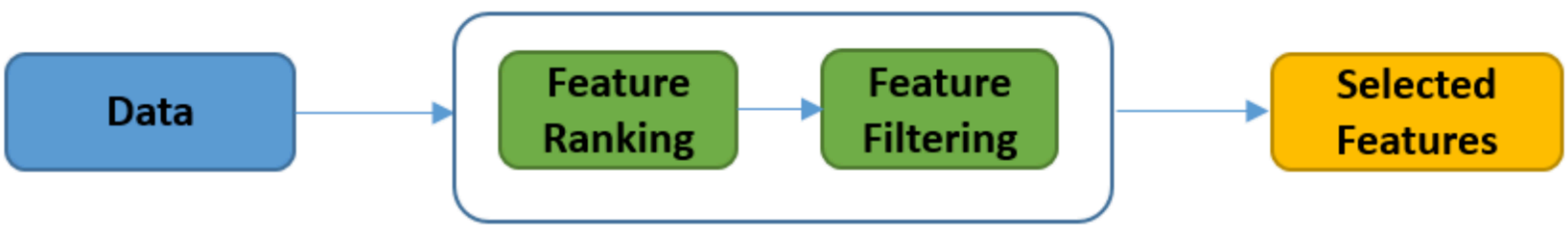
- Can be applied for ANY model!
- Rely on the predictive performance of a predefined learning algorithm to assess features
- Shrink / grow feature set by greedy search
- Repeat until some stopping criteria are satisfied
- Achieve high accuracy for a particular learning method
- Run CV / train-val split per feature
- Computational expensive (worst case search space is 2^{d}) , some typical search strategies are
  – Sequential search
  – Best-first search
  – Branch-and-bound search

Mila

Université de Montréal

Filter Methods

- Independent of any learning algorithms
- Relying 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

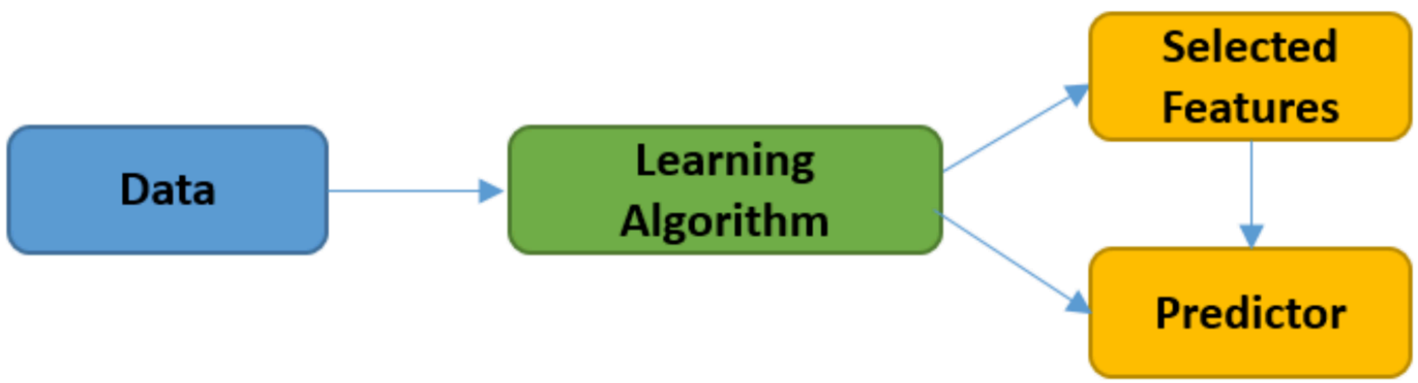


Feature Selection Techniques

- **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

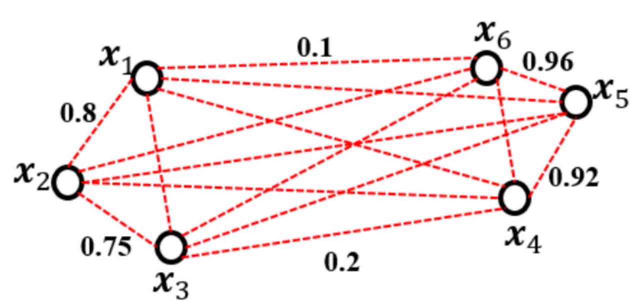
Selection Criteria

Traditional Feature Selection

Similarity based methods	Information Theoretical based methods	Sparse Learning based methods	Statistical based methods

Similarity Technique

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



	x_1	x_2	x_3	x_4	x_5	x_6
x_1		0.8				0.1
x_2	0.8		0.75			
x_3		0.75		0.2		
x_4			0.2		0.92	
x_5				0.92		0.96
x_6	0.1				0.96	

Similarity matrix **S**

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

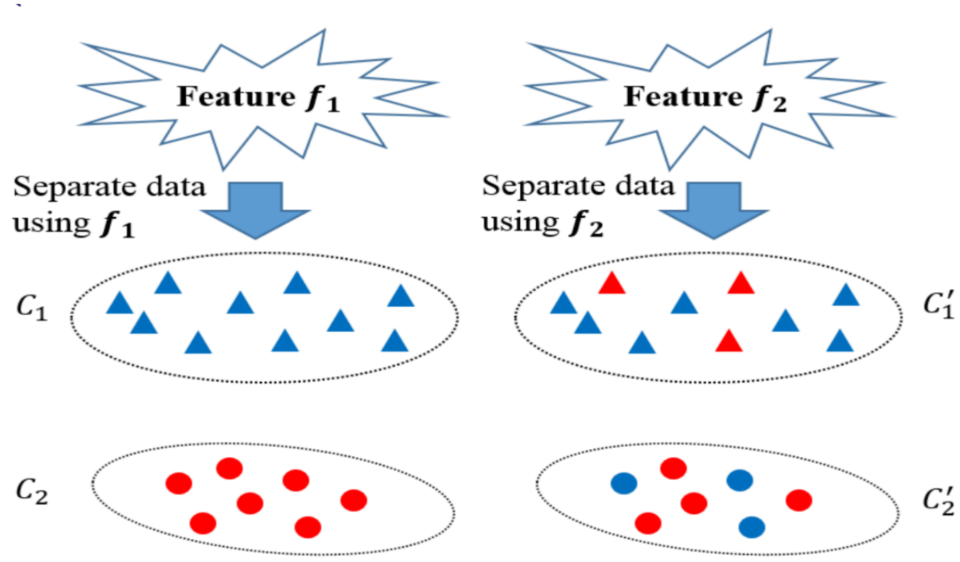
$$\mathbf{S}_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

- 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)



Different shapes denote different values assigned by a feature

Similarity based Methods – A General Framework

- Suppose data similarity matrix is $\mathbf{S} \in \mathbb{R}^{n \times n}$ to find the most relevant features , we need to maximize:

Utility function $U(.)$: how well the feature set preserves the data similarity structure

$$\max_{\mathcal{S}} \boxed{U(\mathcal{S})} = \max_{\mathcal{S}} \sum_{f \in \mathcal{S}} \boxed{U(f)} = \max_{\mathcal{S}} \sum_{\mathbf{f} \in \mathcal{S}} \boxed{\hat{\mathbf{f}}'} \boxed{\hat{\mathbf{S}}} \hat{\mathbf{f}}$$

utility of feature set S

utility of feature f

transformation of feature vector **f**

transformation of similarity matrix **S**

- 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 **f** and similarity matrix **S** are transformed to $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$

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

- Then the Laplacian Score of feature f_i is :

Centered data instances

Measure the consistency of features on the similarity matrix (smaller, the better)

$$score(f_i) = \frac{\boxed{\tilde{\mathbf{f}}_i'\mathbf{L}\tilde{\mathbf{f}}_i}}{\boxed{\tilde{\mathbf{f}}_i'\mathbf{D}\tilde{\mathbf{f}}_i}}, \text{ where } \boxed{\tilde{\mathbf{f}}_i} = \mathbf{f}_i - \frac{\mathbf{f}_i'\mathbf{D}\mathbf{1}}{\mathbf{1}'\mathbf{D}\mathbf{1}}\mathbf{1}$$

Feature variance (higher, the better)

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

- Laplacian score is also equivalent to: $1 - \left(\frac{\tilde{\mathbf{f}}_i}{||\mathbf{D}^{\frac{1}{2}}\tilde{\mathbf{f}}_i||}\right)' \mathbf{S} \left(\frac{\tilde{\mathbf{f}}_i}{||\mathbf{D}^{\frac{1}{2}}\tilde{\mathbf{f}}_i||}\right)$

- A special case of the similarity-based FS framework

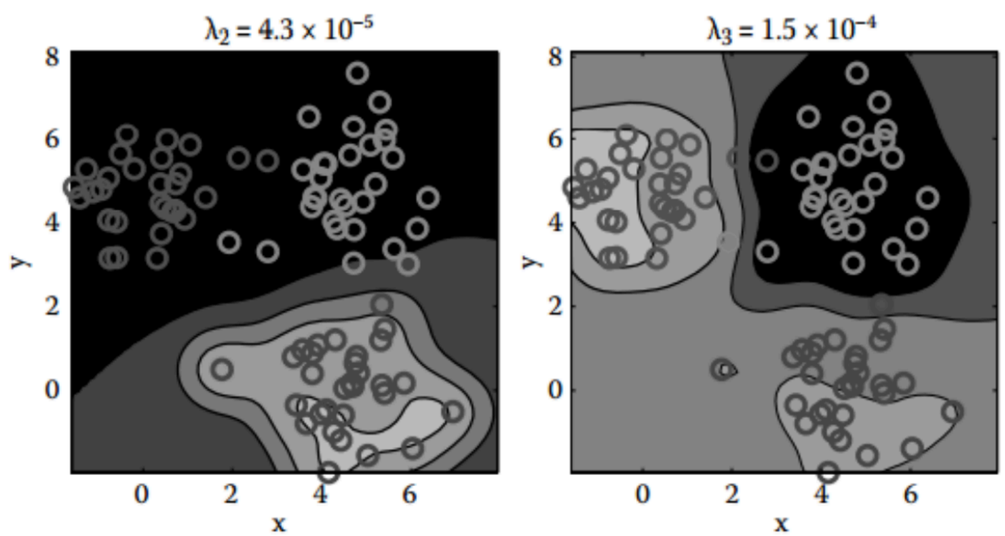
Mila

Université de Montréal

Spectral Feature Selection [Zhao and Liu, 2007]

- Eigenvectors of similarity matrix **S** carry the data distribution

The 2^{nd} and the 3^{rd} eigenvectors from **S**

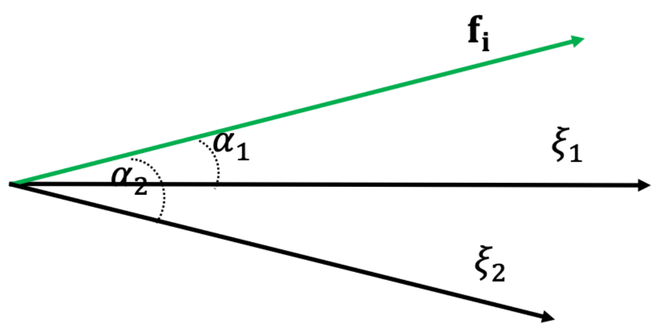


The gray level of the background shows how eigenvectors assign values to the samples

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

Spectral Feature Selection [Zhao and Liu, 2007]

- Measure features' consistency by comparing it with the eigenvectors (e.g. ξ_j) using inner product $\xi_j{}'\mathbf{f_i}$



$\boldsymbol{\alpha}_\mathbf{i}$ is the angle between feature $\mathbf{f_i}$ and ξ_j

- By considering all eigenvectors, the feature score is:

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

Eigenvalues

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

- A special case of the similarity-based FS framework

Fisher Score [Duda et al., 2001]

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

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

- $\boldsymbol{S}^w_{ij}$ is larger if x_i and x_j belong to the same class, smaller otherwise
- $\boldsymbol{S}^b_{ij}$ is larger if x_i and x_j belong to the different classes, smaller otherwise
- A good feature should make instances from different classes **far away** and make instances from the same class **close to each other**

Mila

Université de Montréal

Fisher Score [Duda et al., 2001]

- The score of the i-th feature f_i is

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

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

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

- A special case of the similarity-based FS framework

Trace Ratio Criteria [Nie et al., 2008]

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

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

$\mathcal{F}$ simultaneously

A trace ratio form

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

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

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

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

Mila

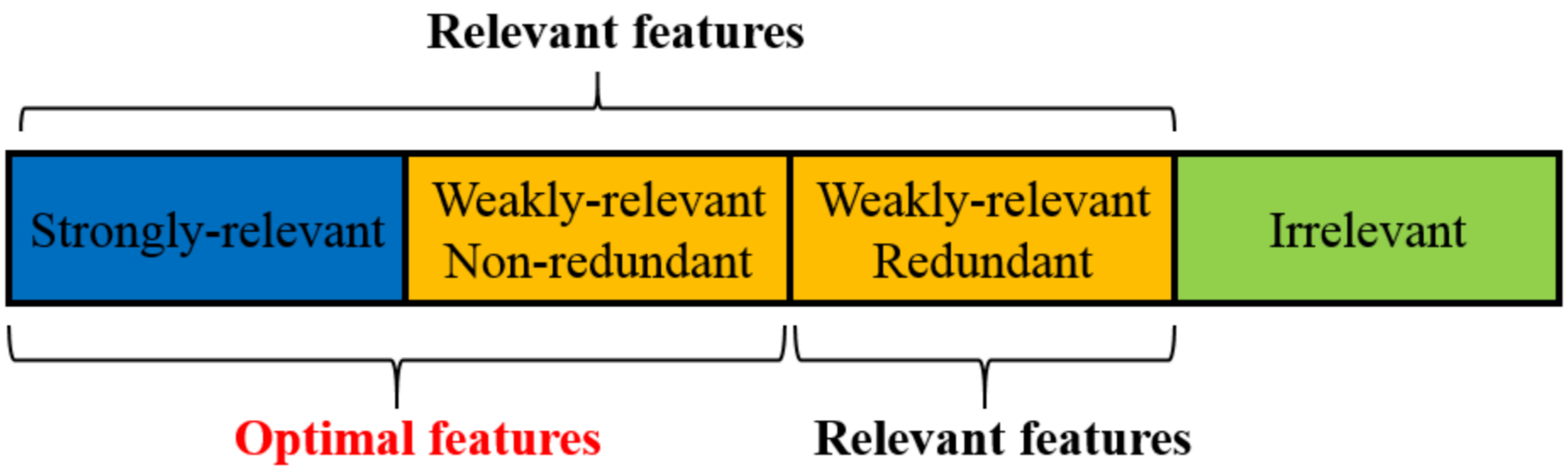
Université de Montréal

Traditional Feature Selection

Similarity based methods	Information Theoretical based methods	Sparse Learning based methods	Statistical 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

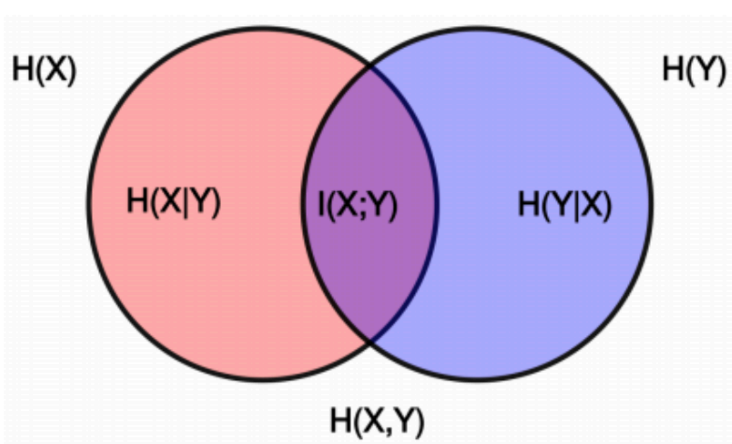
Preliminary - Information Theoretical Measures

- Entropy of a discrete variable X

$$H(X) = -\sum_{x_i \in X} P(x_i) log(P(x_i))$$

- Conditional entropy of X given Y

$$H(X|Y) = -\sum_{y_i \in Y} P(y_j) \sum_{x_i \in X} P(x_i|y_j) log(P(x_i|y_j))$$



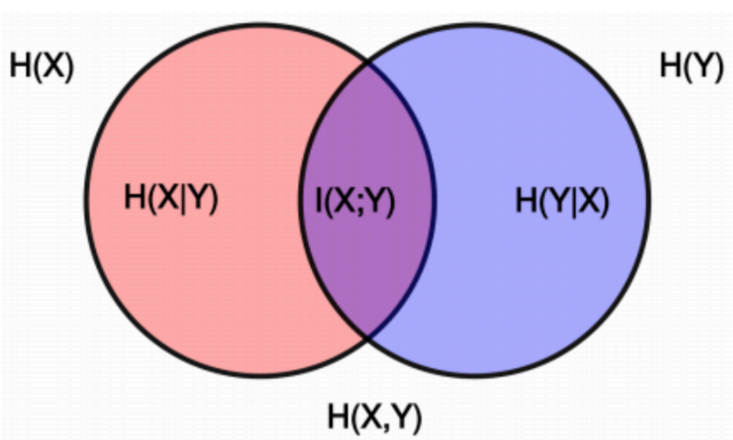
Preliminary - Information Theoretical Measures

- 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_i

– Maximize its correlation with class labels Y:

$$I(f_i; Y)$$

– Minimize the redundancy w.r.t. selected features in S:

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

– Maximize its complementary info w.r.t. selected features in S:

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

Information Theoretic based Methods - A General Framework

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

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

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

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

Between 0 and 1

- But also, g(*) can be a nonlinear function

Mila

Université de Montréal

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

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

- Selecting features independently

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

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

Mila

Université de Montréal

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

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

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 x is f_k

$$score(f_k) = I(f_k;Y) - \underbrace{\boxed{\frac{1}{|\mathcal{S}|}\sum_{f_j \in \mathcal{S}} I(f_k;f_j)}}_{\text{reduced effect of feature redundancy}}$$

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

$$score(f_k) = I(f_k;Y) - \underbrace{\boxed{\sum_{f_j \in \mathcal{S}} I(f_j;f_k)}}_{\text{Correlation with selected features}} + \underbrace{\boxed{\sum_{f_j \in \mathcal{S}} I(f_j;f_k|Y)}}_{\text{Correlation with selected features within classes}}$$

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

Function $g(*)$ Can Also Be Nonlinear

- 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]

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

Mila Université de Montréal

Function g($*$) Can Also Be Nonlinear

- 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]

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

Mila

Université de Montréal

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

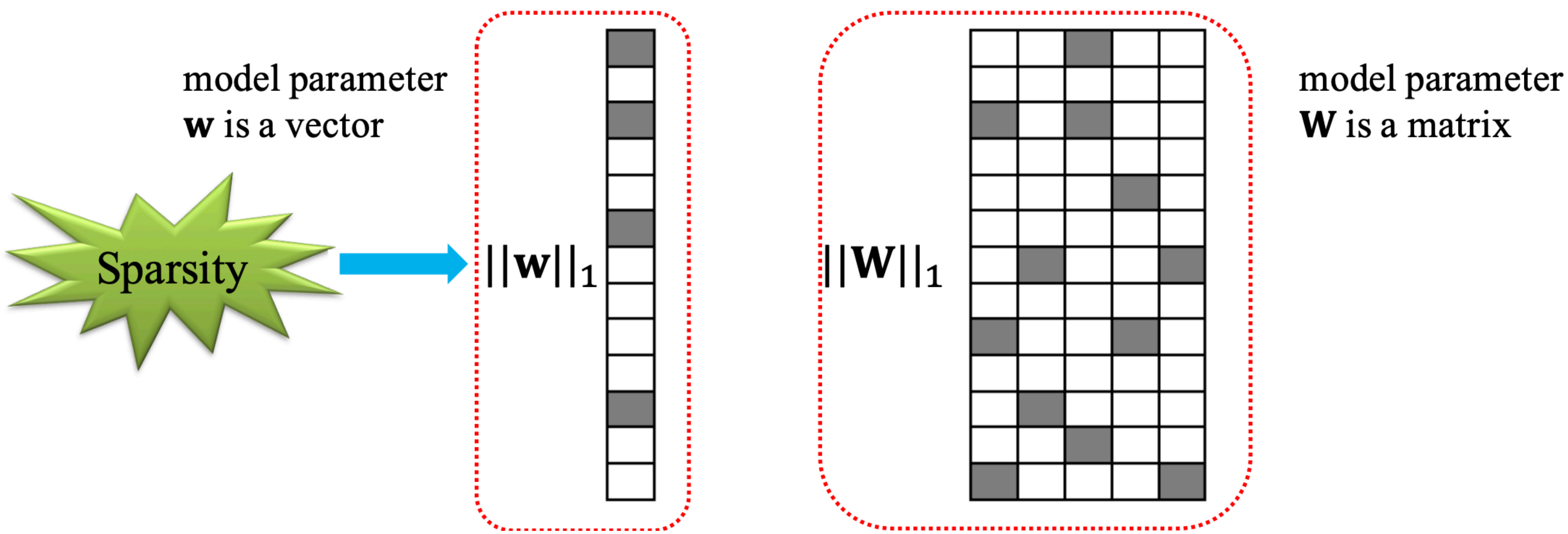
- Cons
  – Most algorithms can only work in a supervised scenario
  – Can only handle discretized data

Traditional Feature Selection

Similarity based methods	Information Theoretical based methods	Sparse Learning based methods	Statistical based methods

What is Feature Sparsity?

- The model parameters in many data mining tasks can be represented as a vector **w** or a matrix **W**
- Sparsity indicates that many elements in **w** and **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 **w** denote the model parameter (a.k.a. feature coefficient), it can be obtained by solving

Balance parameter

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}}\ \boxed{loss(\mathbf{w}; \mathbf{X}, \mathbf{y})} + \boxed{\alpha}\,\boxed{penalty(\mathbf{w})}$$

For classification or regression
- Lease squares loss
- Hinge loss
- Logistic loss
- …

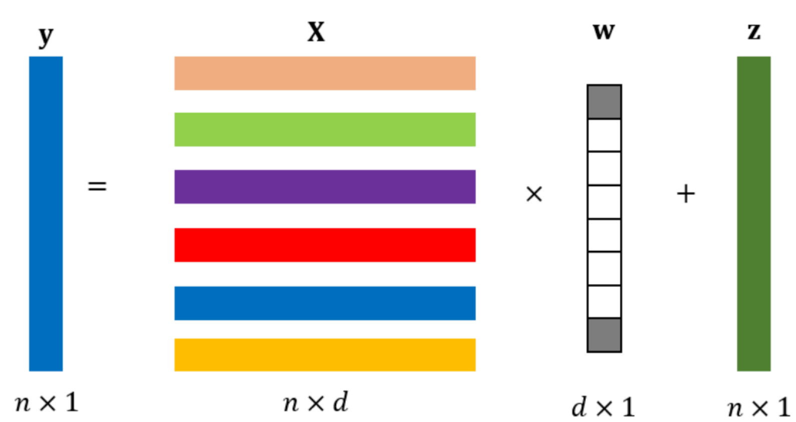
- $||\mathbf{w}||_0$ seeks for optimal features
- However, it is not a valid norm, nonconvex and NP-hard
- It is often relaxed to $||\mathbf{w}||_1$ (Lasso), which is the tightest convex hull

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

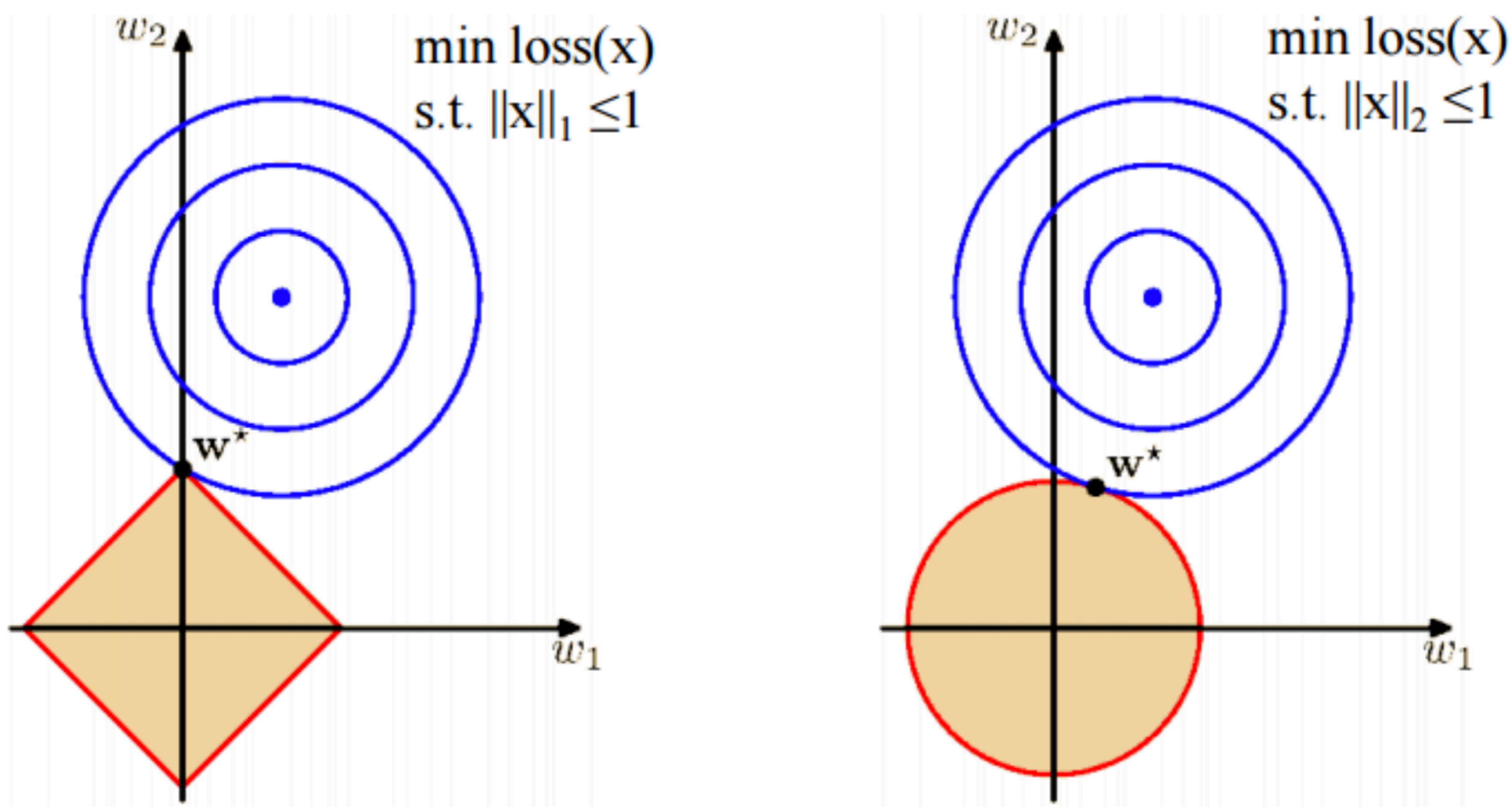
- It is also equivalent to the following model

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

Mila

Université de Montréal

Why ℓ=-norm Induces Sparsity?



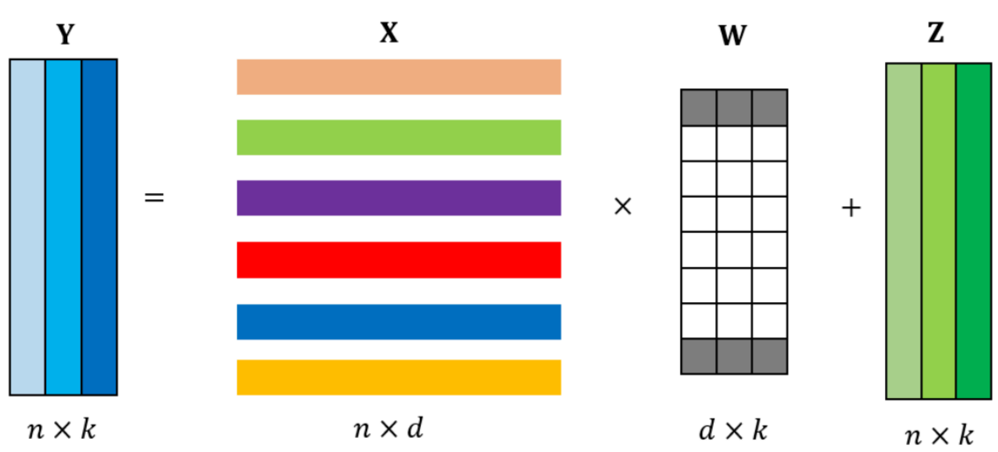
[Bishop, 2006],
[Hastie et al., 2009]

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

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

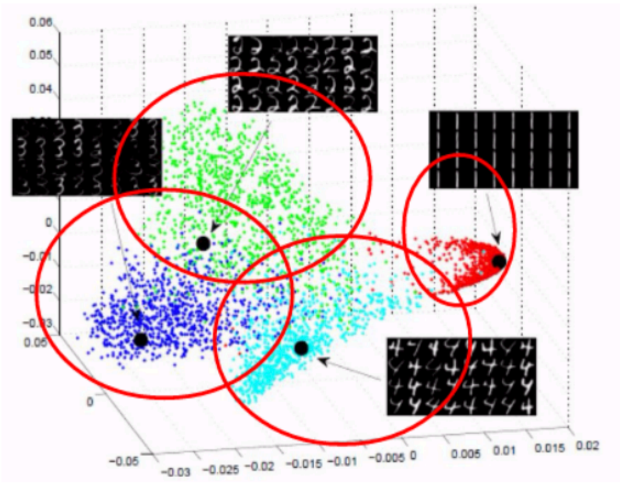
- ||**W**||2 achieves joint feature sparsity across multiple targets • In the case of least square loss with offset, it looks like this



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

- 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 2nd option is preferred as not all features are useful to find clustering structure

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]

- Basic idea: the selected features should preserve cluster structure

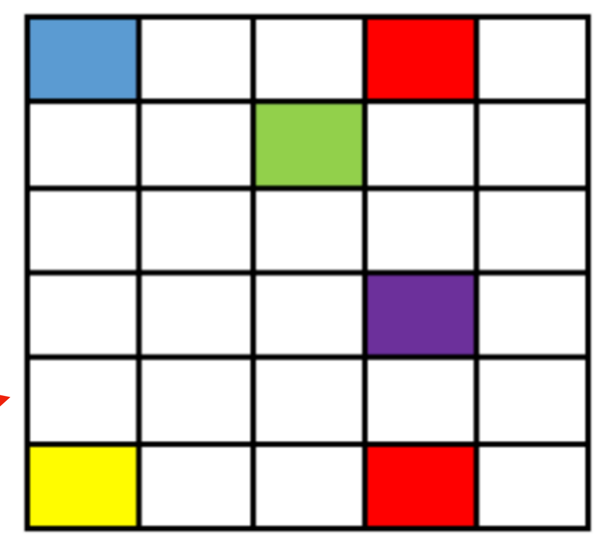
- Step 1: spectral clustering to find intrinsic cluster structure

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

intrinsic cluster indicator vector

- Step 2: perform Lasso on each cluster

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



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

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

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

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

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

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

- Embed cluster matrix into feature selection

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

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

- Feature score obtained from **W** (higher the value, the better)

Mila

Université de Montréal

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
  – With good model interpretability

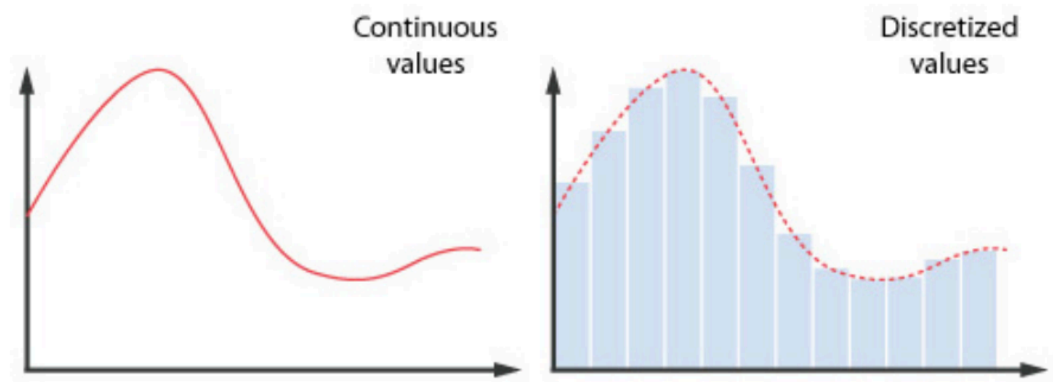
- Cons
  – The selected features may not be suitable for other tasks
  – Require solving non-smooth optimization problems, which is computational expensive

Traditional Feature Selection

Similarity based methods	Information Theoretical based methods	Sparse Learning based methods	Statistical based methods

Statistical based Methods

- This family of algorithms are based on different statistical measures 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

Mean value of samples from the first class

Mean value of samples from the second class

$$t\_score(f_i) = \frac{|\boxed{\mu_1} - \boxed{\mu_2}|}{\sqrt{\frac{\boxed{\sigma_1^2}}{n_1} + \frac{\boxed{\sigma_2^2}}{n_2}}}$$

Standard deviation value for samples from the first class

Standard deviation value for samples from the first class

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

Mila

Université de Montréal

Chi-Square Score [Liu and Setiono, 1995]

- Utilize independence test to assess whether the feature is independent of class label

- Given a feature f_i with r values, its feature score is

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

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

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

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

Traditional Feature Selection

Similarity based methods	Information Theoretical based methods	Sparse Learning based methods	Statistical based methods

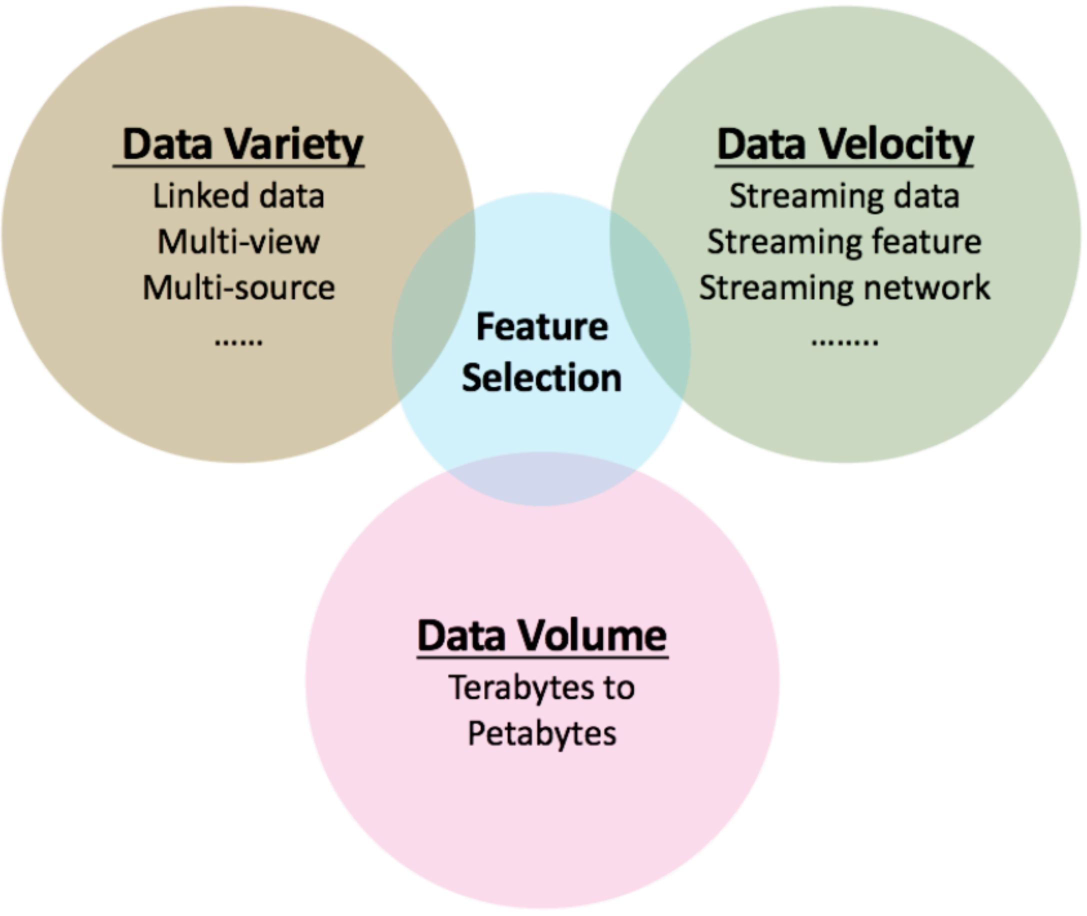
Other Types of Methods

- **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



Feature Selection with Structured Features

Feature Selection with Heterogeneous Data

Multi-Source Feature Selection

Feature Selection with Structured Features - A Framework

- A popular and successful approach is to minimize the fitting error penalized with structural regularization

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}}\ loss(\boxed{\mathbf{w}}; \mathbf{X}, \mathbf{y}) + \boxed{\alpha}\, penalty(\mathbf{w}, \boxed{\mathcal{G}})$$

Balancing parameter

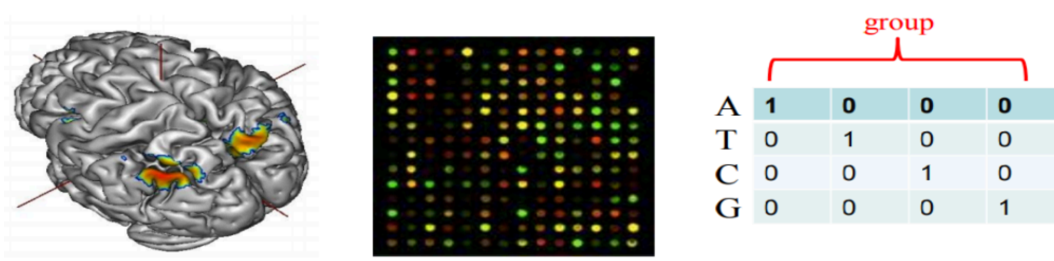
Feature weight

Structure among features is encoded in $\mathcal{G}$

- The above formulation is flexible in incorporating various types of structures among features

Group Structure – Group Lasso [Yuan and Lin, 2006]

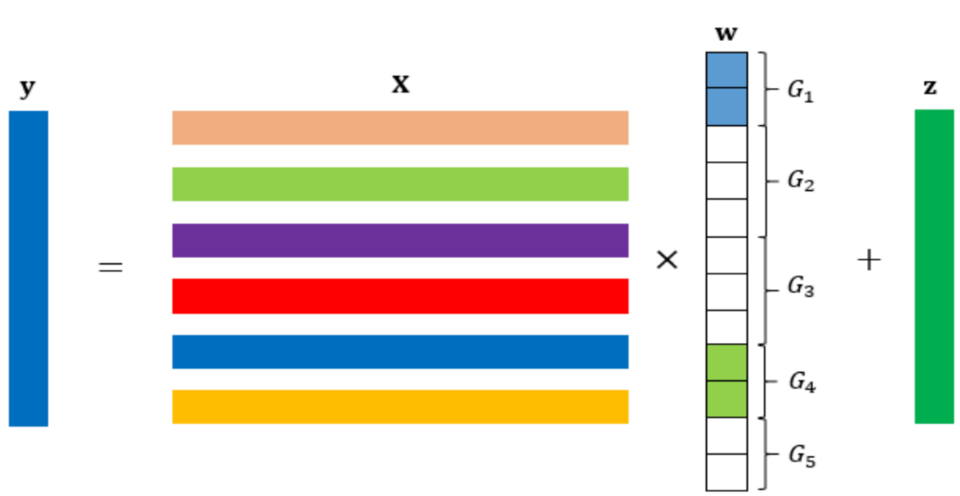
- Features form group structure in many applications



brain region functional group categorical variable

[Liu et al. SDM 2010]

- Group lasso selects or does not select a group as a whole

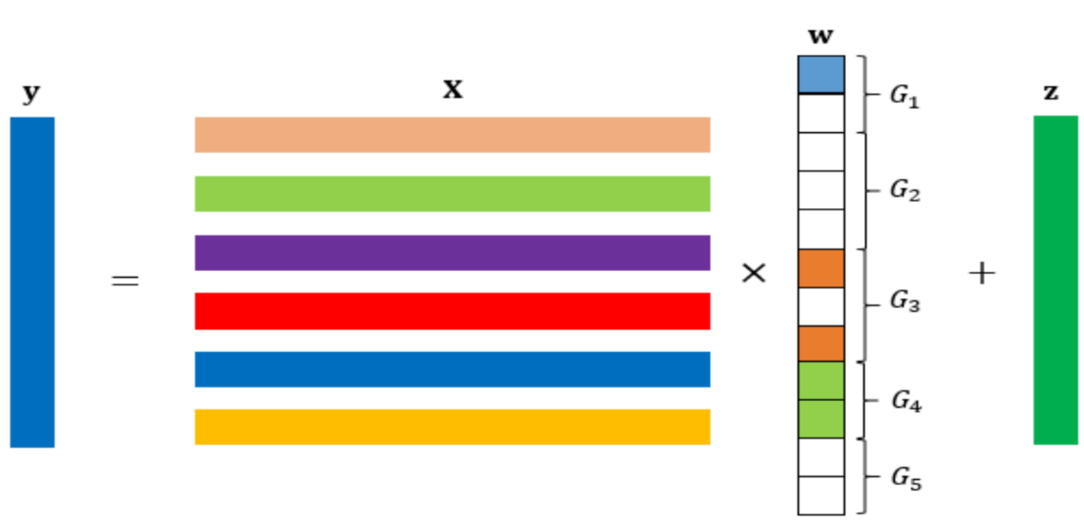


$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \alpha \sum_{i=1}^{k} h_i \|\mathbf{w}_{G_i}\|_2$$

Sparse Group Lasso [Friedman et al., 2010]

- For certain applications, it is desirable to select representative features from selected groups

- Sparse group lasso performs group selection and feature selection simultaneously

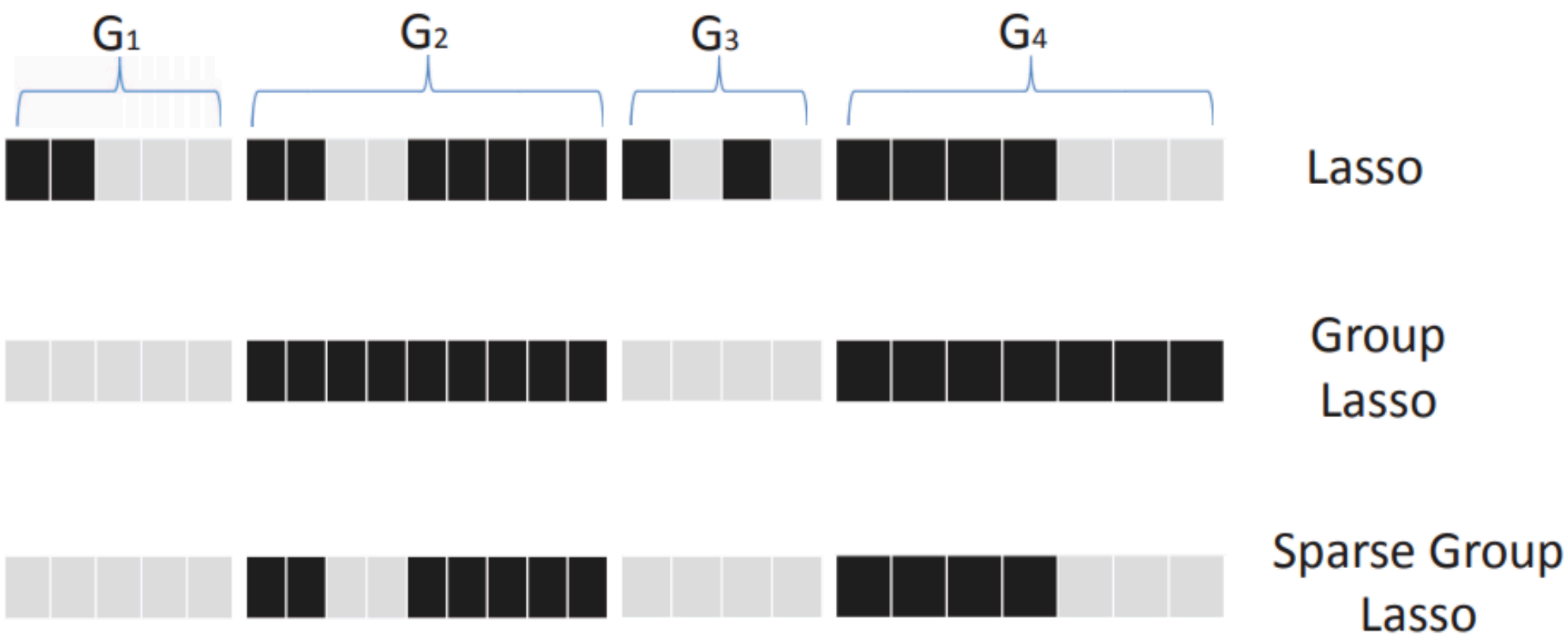


Joint group selection and feature selection

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \boxed{\alpha\|\mathbf{w}\|_1 + (1-\alpha)\sum_{i=1}^{k} h_i\|\mathbf{w}_{G_i}\|_2}$$

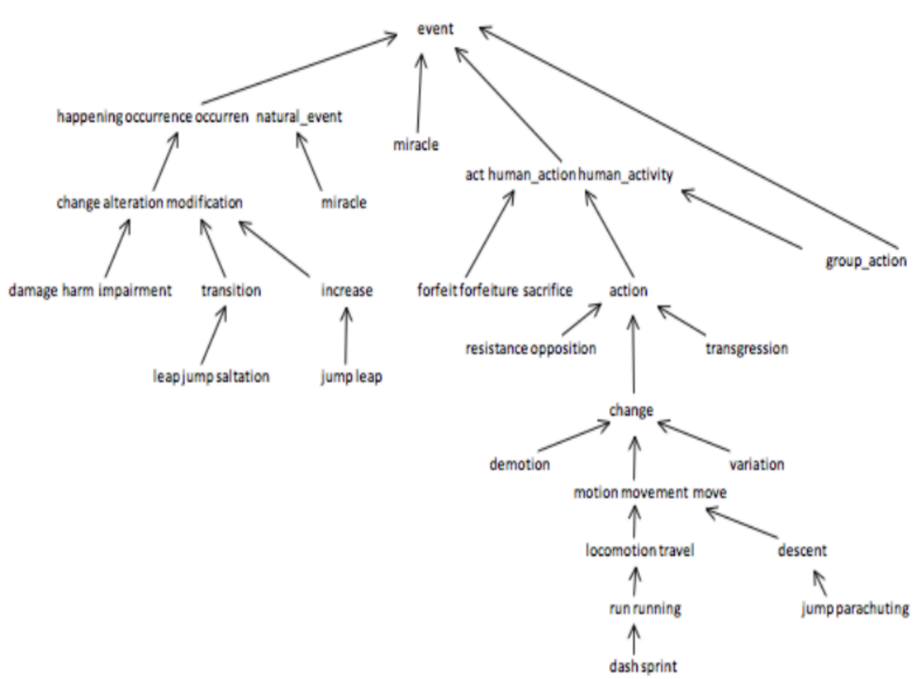
Group Structure - Summary

- Comparison between Lasso, Group Lasso and Sparse Group Lasso

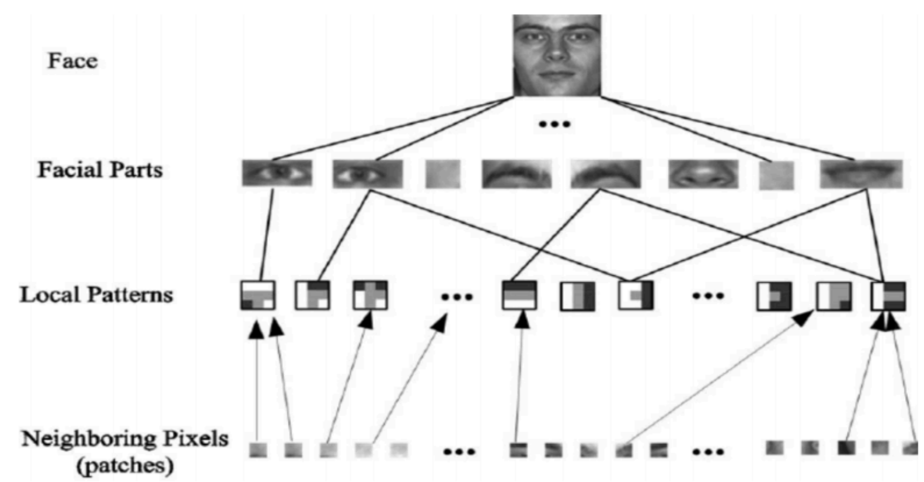


Tree Structure Among Features

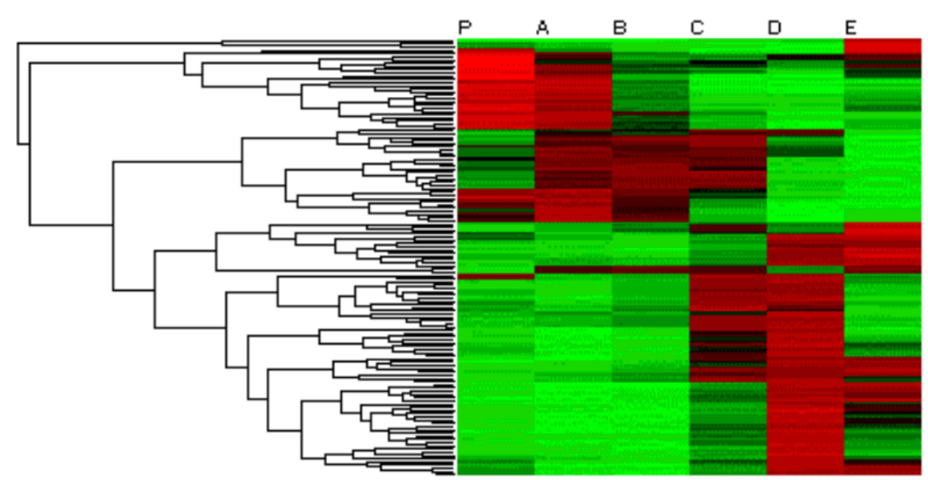
- Features can also exhibit tree (hierarchical) structure
  - – Pixels of face images
  - – Gene expression
  - – Words of documents



Words of documents



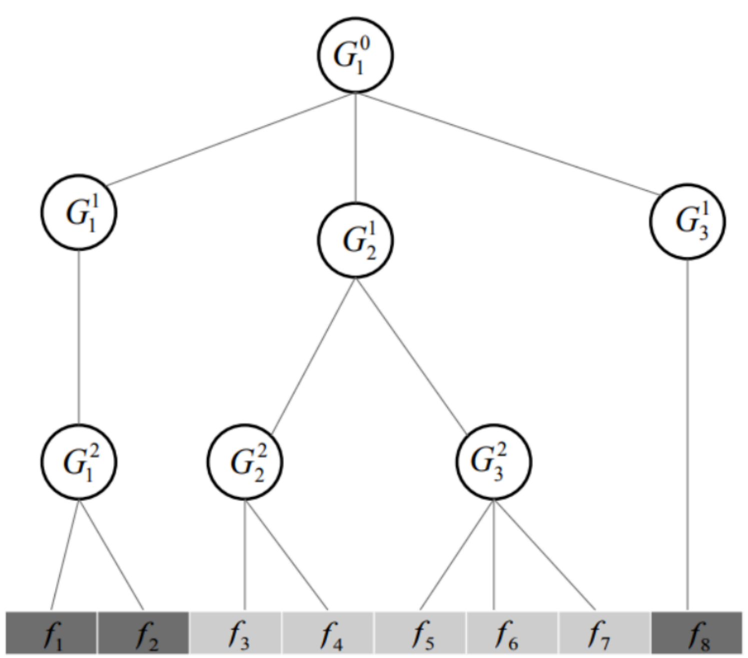
Pixels of face images



Gene expression

Tree-Guided Group Lasso [Liu and Ye, 2010]

- Leaf nodes are individual features
- Internal nodes are a group of features
- Each internal node has a weight indicates how tight the features in its subtree are correlated



$$G_1^0 = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$$

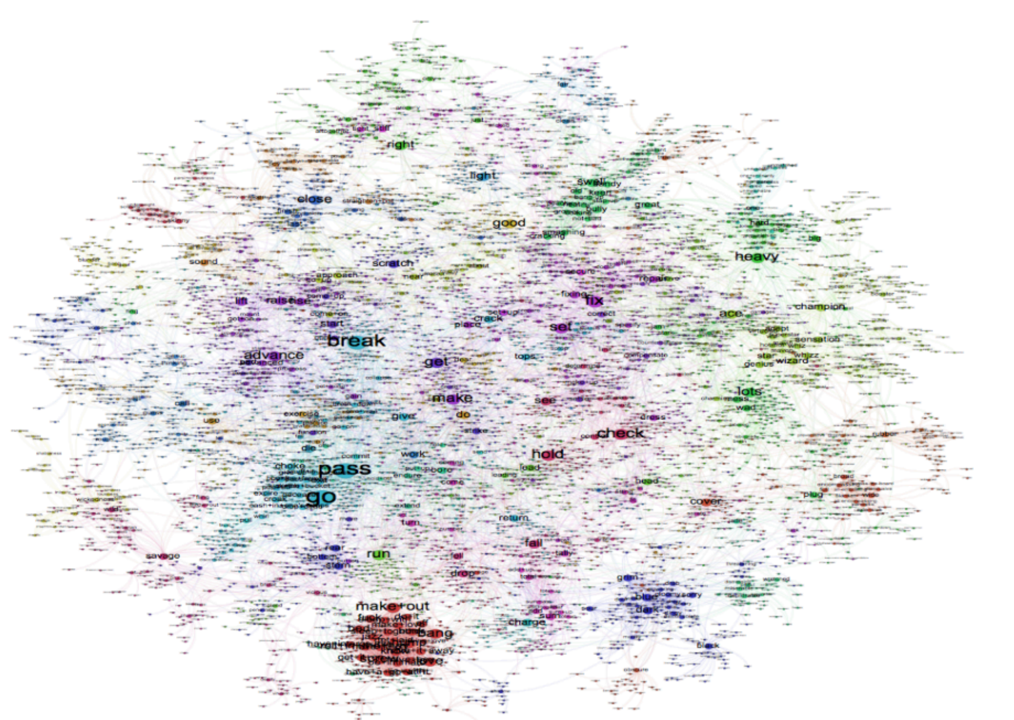
$$G_1^1 = \{f_1, f_2\}, G_2^1 = \{f_3, f_4, f_5, f_6, f_7\}, G_3^1 = \{f_8\}$$

$$G_1^2 = \{f_1, f_2\}, G_2^2 = \{f_3, f_4\}, G_3^2 = \{f_5, f_6, f_7\}$$

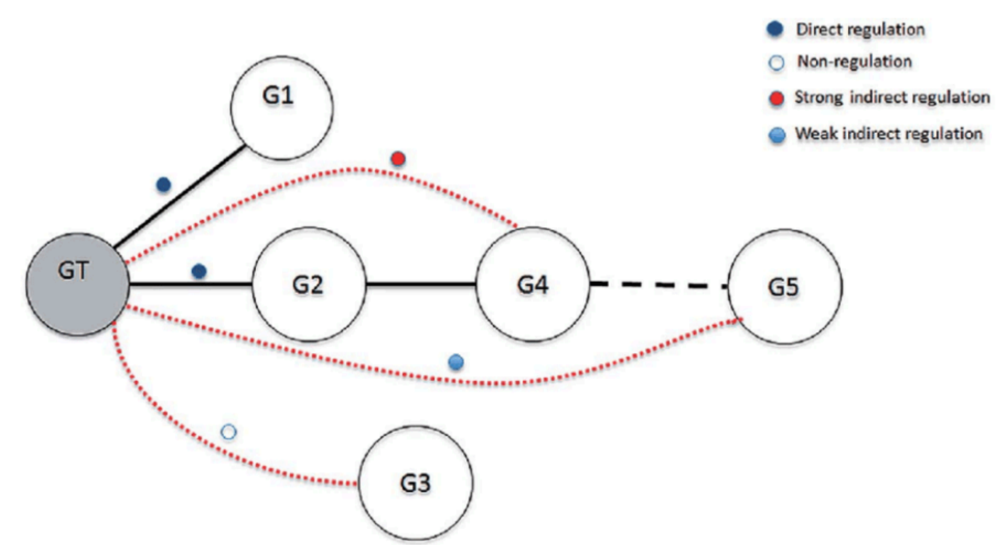
$$\min_{\mathbf{w}} \, loss(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \boxed{\sum_{i=0}^{d} \sum_{j=1}^{n_i} h_j^i ||\mathbf{w}_{G_j^i}||_2}$$

Graph Structure Among Features

- Features can also exhibit graph structure
  – Synonyms and antonyms between different words
  –Regulatory relationships between genes



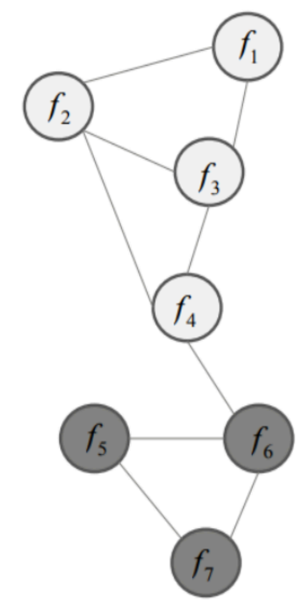
Synonyms and antonyms



Regulatory relations

Graph Lasso [Ye and Liu, 2012]

- Two nodes are connected if two features f_i and f_j tend to be selected together
- Their feature weights are similar
- Impose a regularization on the feature graph



	1	1				
1		1	1			
1	1		1			
	1	1			1	
					1	1
		1	1			1
				1	1	

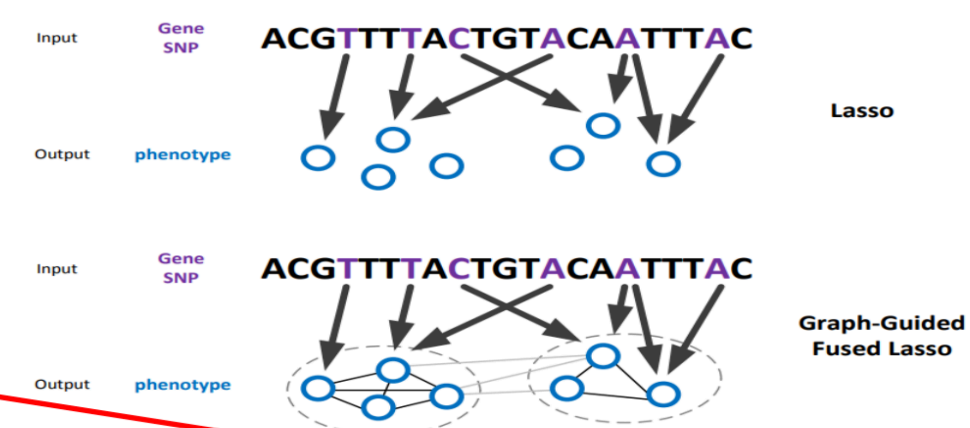
Graph Feature Representation: **A**

$$\min_{\mathbf{w}} loss(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha ||\mathbf{w}||_1 + (1-\alpha) \overbrace{\boxed{\sum_{i,j} \mathbf{A}(i,j)(\mathbf{w}_i - \mathbf{w}_j)^2}}^{\text{Graph Features}}$$

Graph Laplacian

Graph-Guided Fused Lasso (GFLasso) [Kim and Xing, 2009]

- Graph Lasso assume features connected together have similar feature coefficients
- However, features can also be negatively correlated
- GFLasso explicitly considers both positive and negative feature correlations



Positively correlated: $\mathbf{A}_{ij} = 1, r_{ij} > 0$
Negatively correlated: $\mathbf{A}_{ij} = 1, r_{ij} < 0$

$$\min_{\mathbf{w}} \; loss(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha ||\mathbf{w}||_1 + (1-\alpha) \sum_{i,j} \mathbf{A}(i,j) |\mathbf{w}_i - \boxed{\text{sign}(r_{i,j})} \mathbf{w}_j|$$

Feature Selection with Structured Features - Summary

- Incorporate feature structures as prior knowledge

- Pros
  – Improve the learning performance in many cases
  – Make the learning process more interpretable

- Cons
  –Require label information to guide feature selection
  –Require to solve complex non-smooth optimization problems

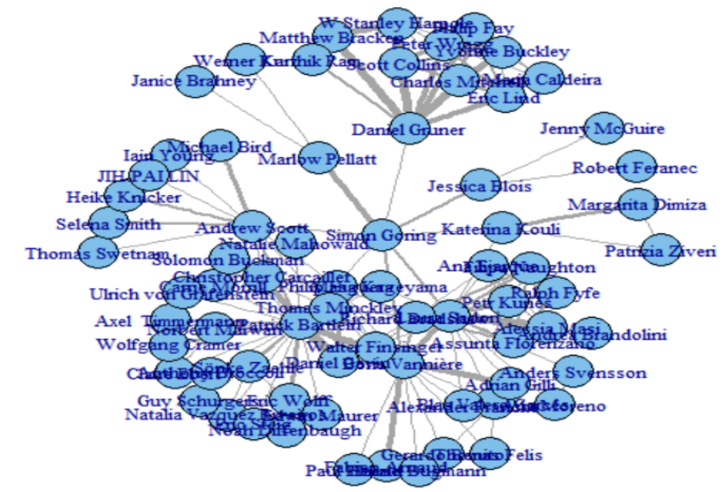
Feature Selection with Structured Features

Feature Selection with Heterogeneous Data

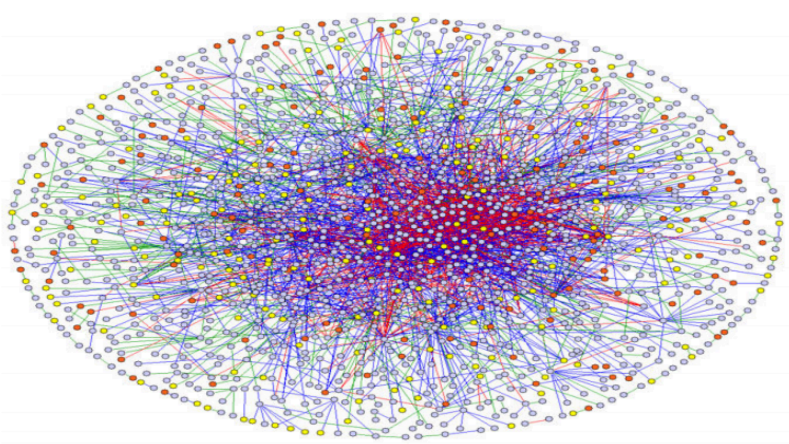
Multi-Source Feature Selection

Feature Selection with Heterogeneous Data

Social network



Coauthor network



Gene network

Transportation network

Feature Selection with Heterogeneous Data

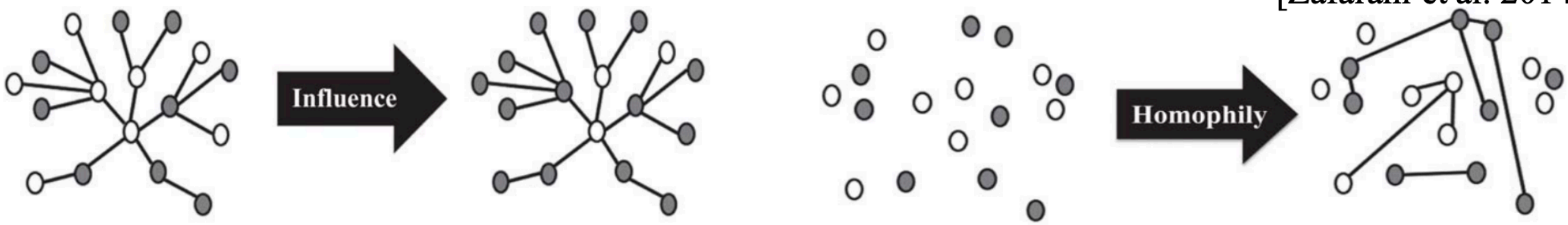
- Traditional feature selection algorithms are for a single source and are heavily based on the data i.i.d. assumption

- 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?

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



[Zafarani et al. 2014]

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

- But not all features are hinged with the network structure!

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

- 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**

Joint feature sparsity

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

Avoid overfitting

- Employ graph regularization to model links

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

undirected network

$$\mathbf{L} = \mathbf{D} - \boxed{\mathbf{A}}$$

Adjacency matrix

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

directed network

P: transition matrix of random walk
Π: stationary distribution

- Objective function of FSNet

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

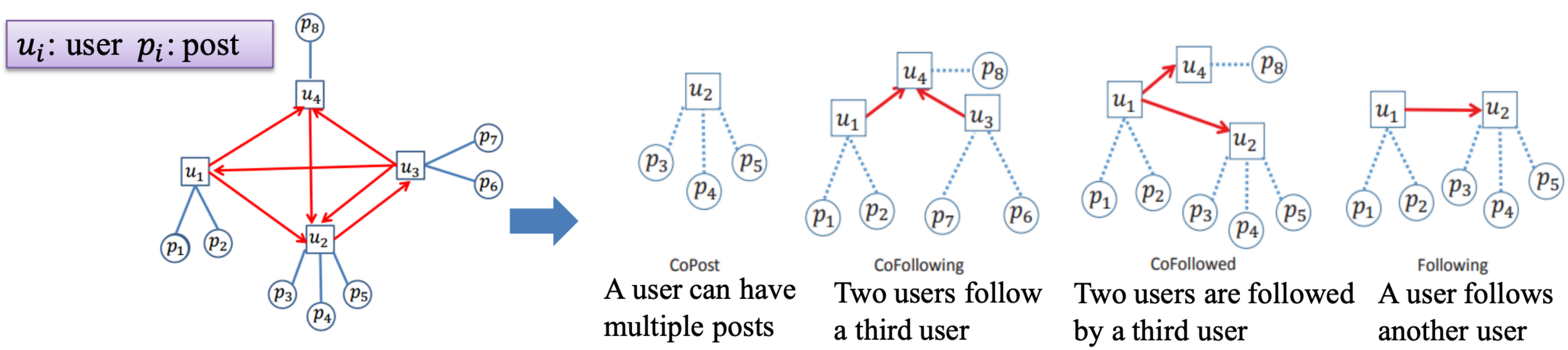
Feature scores are obtained from matrix **W**

Mila

université de Montréal

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

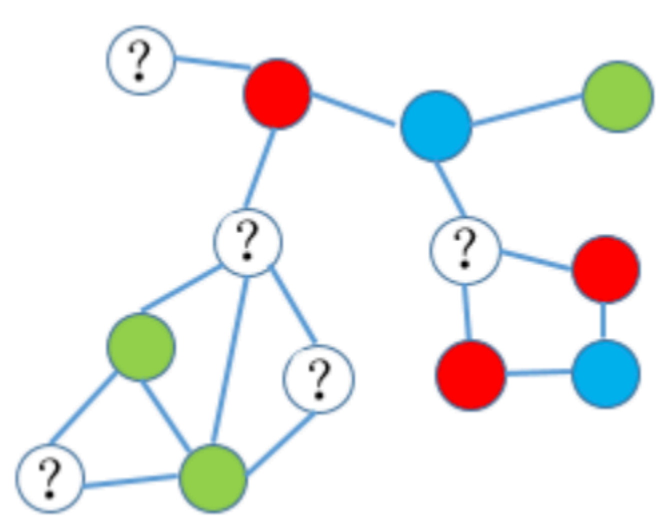
$$\min_{\mathbf{W}} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 + \alpha \|\mathbf{W}\|_{2,1} + \beta \sum_{u \in \mathbf{u}} \sum_{\underbrace{\{p_i, p_j\} \in \mathbf{p}_u}_{\text{CoPost relations}}} \overbrace{\|\mathbf{X}(i,:)\mathbf{W} - \mathbf{X}(j,:)\mathbf{W}\|_2^2}^{\text{CoPost hypothesis}}$$

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 price comes down! #apple” [Wu and Huang 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?

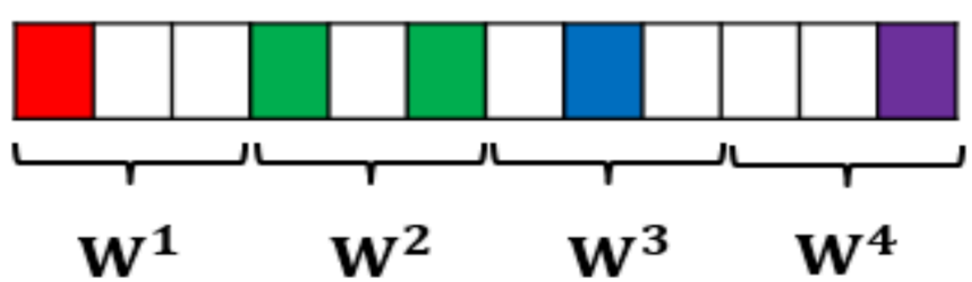


Personalized Feature Selection [Li et al., 2017]

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

$$\min_{\tilde{\mathbf{W}},\mathbf{W}^i} \sum_{i=1}^{n} \|\mathbf{x}_i(\underbrace{\tilde{\mathbf{W}}}_{\text{global feature weight for all nodes}} + \underbrace{\mathbf{W}^i}_{\text{local feature weight for the i-th node}}) - \mathbf{y}_i\|_2^2 + \beta \sum_{i=1}^{n} \underbrace{\|\mathbf{W}^i\|_{2,1}^2}_{\text{exclusive group lasso}} + \gamma \|\tilde{\mathbf{W}}\|_{2,1}$$

- 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

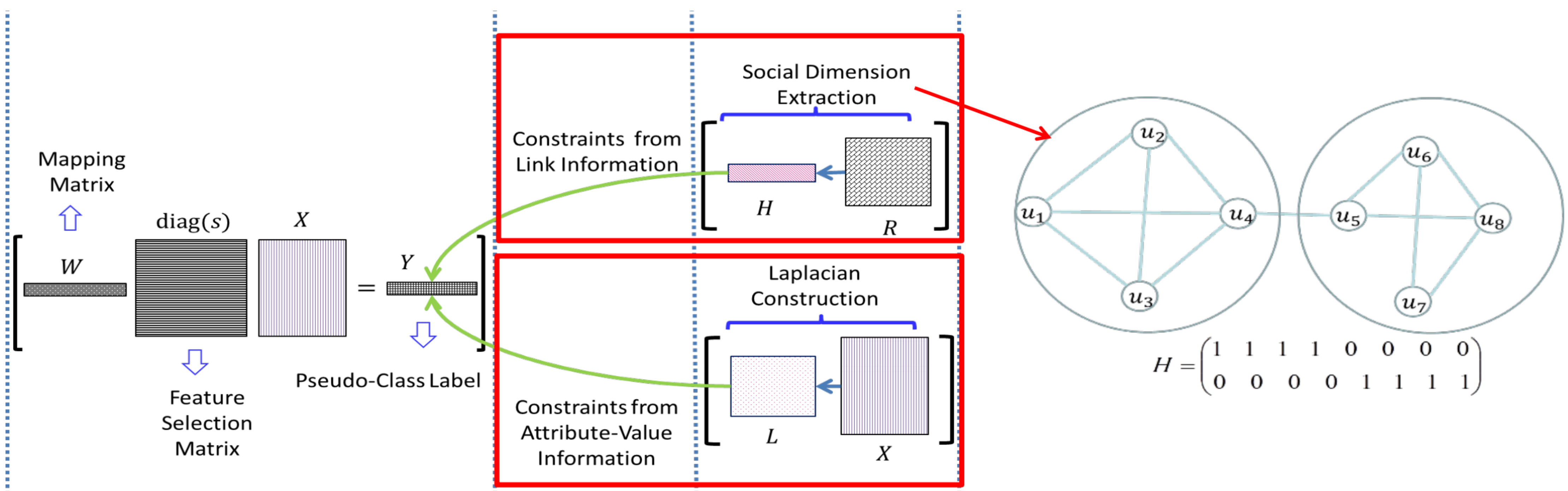
- The objective function

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

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

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 matrix $\boldsymbol{S}_w$, $\boldsymbol{S}_b$, and $\boldsymbol{S}_t$

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

Weighted social dimension matrix

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

- 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

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

Obtained from content similarity matrix using RBF

Mila

Université de Montréal

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

- Optimization framework of LUFS

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

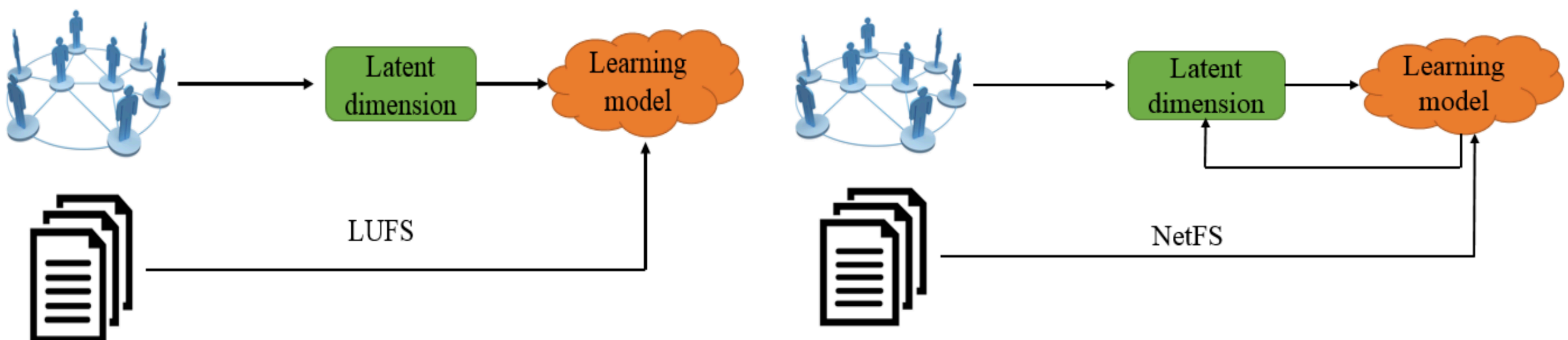
$$\boxed{\mathbf{Y} = \mathbf{W}'\mathrm{dig}(\mathbf{s})\mathbf{X}}$$

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

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

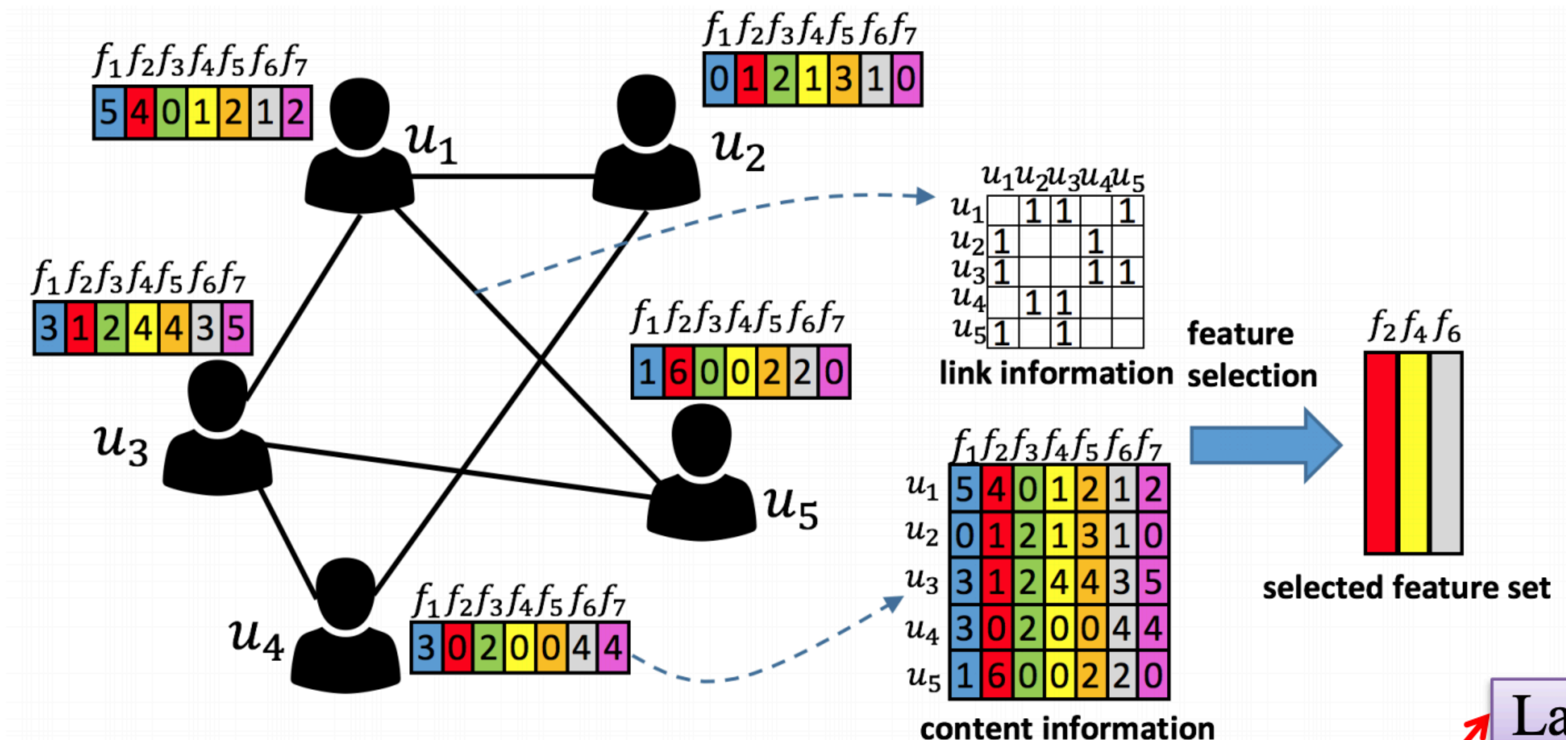
Robust Unsupervised on Networks (NetFS) [Li et al., 2016]

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



Difference between LUFS and NetFS

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



Latent representation

$$\min_{\mathbf{U}\geq 0,\mathbf{W}} ||\mathbf{X}\mathbf{W}-\boxed{\mathbf{U}}||_F^2 + \alpha||\mathbf{W}||_{2,1} + \frac{\beta}{2}||\boxed{\mathbf{A}}-\mathbf{U}\mathbf{U}'||_F^2$$

Latent representation as constraint to guide FS

Latent representation learning and feature selection complement each other

Feature Selection with Structured Features

Feature Selection with Heterogeneous Data

Multi-Source Feature Selection

Multi-Source Feature Selection [Zhao and Liu, 2008]

- Given multiple local geometric patterns in affinity matrix $\mathbf{S}_i$, the global $\mathbf{S} = \sum_{i=1}^{m} \mathbf{S}_i$
- Geometry-dependent sample covariance matrix for the target source $\mathbf{X}_i$ is

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

$$\mathbf{D}_{kk} = \sum_j \mathbf{S}_{kj}$$

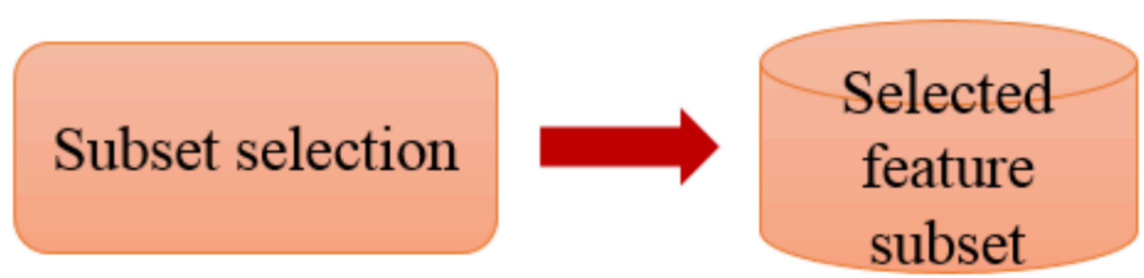
$$\mathbf{\Pi}_{jj} = \|\mathbf{D}^{0.5}\mathbf{X}_i(:,j)\|^{-1}$$

- Two ways to obtain relevant features from
  – Sort the diagonal of 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

Mila

Université de Montréal

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., NMI)

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

Challenges of Feature Selection

- **Scalability**
- **Stability Challenge**
-

Scalability Challange
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 Challange
Feature size

- Most existing feature selection algorithms have a time complexity proportional to $O(d^2)$ or even $O(d^3)$

- Data of ultrahigh-dimensionality emerges
  – Text mining
  – Information retrieval
  – Brain image

- For many feature selection algorithms, efficiency deteriorates quickly as could be very large

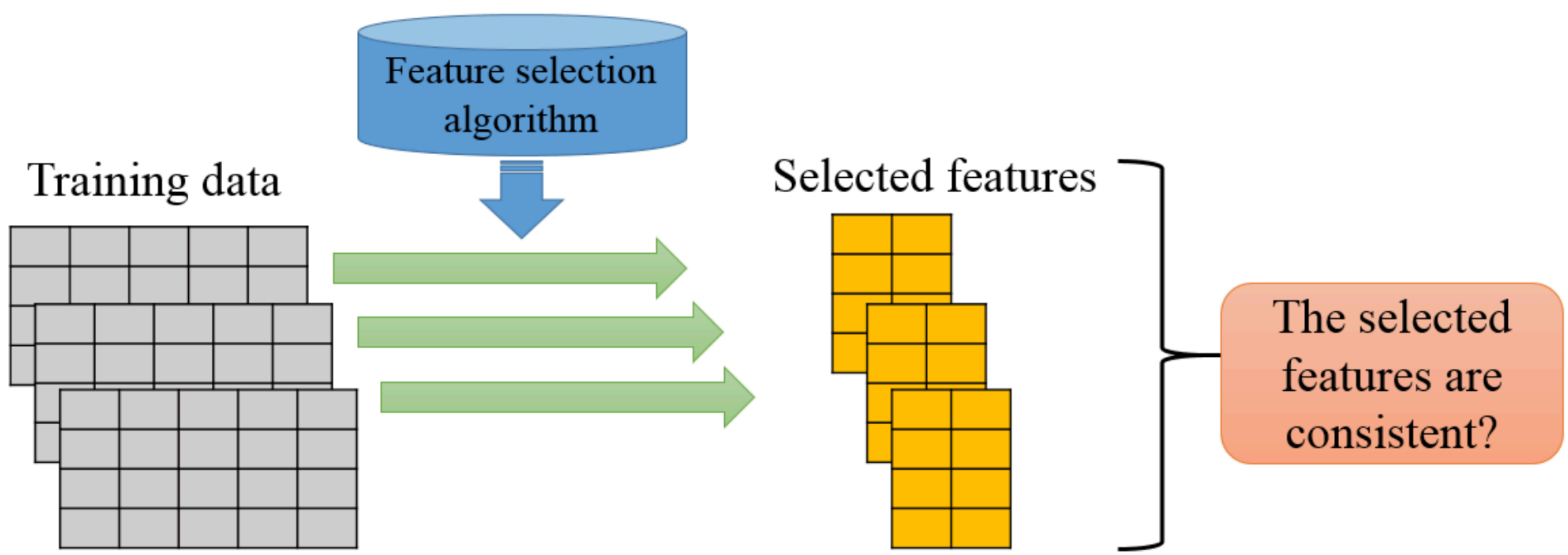
- Well-designed feature selection algorithms work in linear or sub-linear time are preferred

Mila

Université de Montréal

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?

- 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

- 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

"Your recent Amazon purchases, Tweet score and location history makes you 23.5% welcome here."

- 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

We will cover more in Week 14

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 streaming data

References

- Check all the references at the end of: http://www.public.asu.edu/~jundongl/tutorial/KDD17/KDD17.pdf