From Feature Selection to Instance-wise Feature Acquisition\textsuperscript{1}

Tutorial @ SDM 2024

Daphney-Stavroula Zois\textsuperscript{1}  Charalampos Chelmis\textsuperscript{2}

\textsuperscript{1}Electrical and Computer Engineering Department

\textsuperscript{2}Computer Science Department

Saturday, April 20th, 2024

\textsuperscript{1}This research is based upon work supported by the National Science Foundation grants ECCS-1737443 & CNS-1942330 and a Google AI for Social Good award.
Advanced Topics
Advanced Topics

- Feature Acquisition
  - Interpretability (e.g., [liy23])
  - Dealing with structure (e.g., multidimensional Bayesian network classification [ELZ21, EZ23])
  - Reducing label uncertainty or learning to defer (e.g., dynamic classifier selection [EZC23b, EZC23a])

- Feature Selection
  - Incorporating fairness constraints (e.g., [GSSV22])
  - Feature selection for hierarchical classification (e.g., [ZH+19])
Using sparse set of features to classify data instances is essential for model interpretability

- Observe which features contribute to each model output
- Sparsity can be achieved
  - globally by incorporating regularizer to objective function
  - instance-level, e.g., evaluate features along different decision paths in decision trees

Goal: assess interpretability of IFCO [LZ21]
Interpretability of IFCO

- Model–based interpretability: humans can understand how model behaves and which factors influence its decision–making process
- Post–hoc interpretability: relationships learned by model from given dataset
For demonstration purpose, we use the **German credit–risk dataset**: classify people as high or low credit risk

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F₁</td>
<td>Checking account status</td>
<td>F₁₁</td>
<td>Present residence</td>
</tr>
<tr>
<td>F₂</td>
<td>Duration in months</td>
<td>F₁₂</td>
<td>Property</td>
</tr>
<tr>
<td>F₃</td>
<td>Credit history</td>
<td>F₁₃</td>
<td>Age in years</td>
</tr>
<tr>
<td>F₄</td>
<td>Purpose of the credit</td>
<td>F₁₄</td>
<td>Other installment plans</td>
</tr>
<tr>
<td>F₅</td>
<td>Credit amount</td>
<td>F₁₅</td>
<td>Housing</td>
</tr>
<tr>
<td>F₆</td>
<td>Savings account status</td>
<td>F₁₆</td>
<td>Existing credits</td>
</tr>
<tr>
<td>F₇</td>
<td>Present employment (years)</td>
<td>F₁₇</td>
<td>Job</td>
</tr>
<tr>
<td>F₈</td>
<td>Installment rate</td>
<td>F₁₈</td>
<td>Number of dependents</td>
</tr>
<tr>
<td>F₉</td>
<td>Personal status</td>
<td>F₁₉</td>
<td>Telephone</td>
</tr>
<tr>
<td>F₁₀</td>
<td>Other debtors</td>
<td>F₂₀</td>
<td>Foreign worker</td>
</tr>
</tbody>
</table>

**Standard interpretable models:**
- Logistic regression with L1–norm regularizer (LR)
- Decision tree (DT)
Model-based Interpretability

- **Sparsity**: use sparse set of features for classification
  - LR: global sparsity by using the L1–norm penalty
  - DT: instance-level sparsity by evaluating features along different branches (greedy learning of tree structure)
  - IFCO: instance-level sparsity by using feature acquisition cost $\sum_{k=1}^{R} e(F_{\sigma(k)})$
- **Sparsity stability**: interpretations are meaningless if sparsity varies drastically due to small perturbation in training dataset
Model–based Interpretability

- **Simulatability**: human can reason about decision–making process
  - LR: dot product between feature vector and weight vector
  - DT: hierarchical decision–making
  - IFCO:
Model–based Interpretability

- **Modularity**: ability to interpret meaningful portions of decision–making process independently
  - LR: affine transformation of input feature space (i.e., $w_i F_i$)
  - DT: each tree node is modular block that contributes to final classification decision
  - IFCO: sequential decision–making process based on sufficient statistic

\[
\pi_{\sigma^*}(k) = \frac{\Delta(F_{\sigma^*(k)}|F_{\sigma^*(1)}, \ldots, F_{\sigma^*(k-1)}, C)}{\Delta T(F_{\sigma^*(k)}|F_{\sigma^*(1)}, \ldots, F_{\sigma^*(k-1)}, C)\pi_{\sigma^*(k-1)}}
\]

- **Conditional independence** assumption helps to decompose $\pi_{\sigma^*(k)}$ into simple and meaningful portions in terms of $P(F_{\sigma^*(k)}|C)$
Post–hoc Interpretability: Dataset–level Interpretations

- **Partial dependence**: marginal effects of individual feature on output of machine learning model

\[
PD(F_i) \approx \frac{1}{N} \sum_{n=1}^{N} \hat{f}(F_i, \bar{F}_i^{(n)})
\]

![Graph showing partial dependence of credit risk on savings account status and age](image-url)
Post-hoc Interpretability: Dataset-level Interpretations

- **Feature importance**: number of times specific feature contributes to specific classification decision
Post–hoc Interpretability: Dataset–level Interpretations

- **Accuracy stability**: test accuracy should be stable for any perturbations in training data

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Feat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFCO</td>
<td>0.754±0.040</td>
<td>5.85</td>
</tr>
<tr>
<td>DT</td>
<td>0.702±0.044</td>
<td>6.78</td>
</tr>
<tr>
<td>LR</td>
<td>0.740±0.034</td>
<td>14.0</td>
</tr>
<tr>
<td>XGB</td>
<td>0.755±0.037</td>
<td>19.9</td>
</tr>
</tbody>
</table>

- Gradient boosted trees (XGB) *(black box)* requires 3.4 times more features for a just 0.1% improvement
Post–hoc Interpretability: Prediction–level interpretations

- **Incorrectly predicted**

<table>
<thead>
<tr>
<th>Feature</th>
<th>High Risk</th>
<th>Low Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>π₀</strong></td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td><strong>F₁</strong></td>
<td>0.38</td>
<td>0.62</td>
</tr>
<tr>
<td><strong>F₃</strong></td>
<td>0.23</td>
<td>0.77</td>
</tr>
<tr>
<td><strong>F₆</strong></td>
<td>0.13</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Correctly predicted \(\rightarrow\) **low credit–risk**

- **Correctly predicted**

<table>
<thead>
<tr>
<th>Feature</th>
<th>High Risk</th>
<th>Low Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>π₀</strong></td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td><strong>F₁</strong></td>
<td>0.38</td>
<td>0.62</td>
</tr>
<tr>
<td><strong>F₃</strong></td>
<td>0.74</td>
<td>0.26</td>
</tr>
<tr>
<td><strong>F₂</strong></td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td><strong>F₁₂</strong></td>
<td>0.88</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Correctly predicted \(\rightarrow\) **high credit–risk**

- **bad checking account status**
- **good credit history**
- **good savings account status**

**Correctly predicted**

- **bad checking account status**
- **bad credit history**
- **credit history of 36 months**
- **no known property**

**Correctly predicted** \(\rightarrow\) **high credit–risk**
Many real-world applications (e.g., medical diagnosis, behavioral analysis)
  • Bayesian networks used to describe relationships between variables
  • Variables not directly observable but can be inferred via features

Multi-dimensional Bayesian network classification [GBBL21] learns underlying unknown Bayesian network structure between variables in $X$ and features in $F$, and then performs inference to compute the values of variables in $X$. 
Problem Statement

- What happens if features are acquired at a cost?
- **Goal**: accurately classify each data instance during testing, while keeping total feature acquisition cost minimum when data instance label corresponds to known Bayesian network of multiple class variables.
Optimization Setup

- $\mathcal{G} = (X, E)$: known Bayesian network structure
- $X \triangleq \{X_1, X_2, \ldots, X_n\}$: set of nodes corresponding to categorical variables
- $E$: set of directed edges to represent relationships between categorical variables
- $F \triangleq \{F^{X_1}_{K_1}, \ldots, F^{X_2}_{K_2}, \ldots, F^{X_n}_{K_n}\}$: set of features, where $F^{X_i}_{k}$ is $k$th feature associated with variable $X_i$
- $e^i_k$: cost of acquiring $k$th feature associated with variable $X_i$
- $C^{X_i}_l$: class value for variable $X_i$
Optimization Setup

- Introduce random variables
  - \( R_i \in \{0, \ldots, K_i\} \): last feature acquired before classification decision for variable \( X_i \)
  - \( D_{R_i} \in \{1, \ldots, N_i\} \): classification decision based on \( R_i \) features for variable \( X_i \)

\[
\min_{R, D_R} J(R, D_R)
\]

\[
J(R, D_R) = \mathbb{E} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{R_i} e_i^k + \sum_{j} \sum_{m} M_{mj} P(D_R = j, C = c_m) \right\}
\]

- The computational complexity of directly solving the above problem is high
Alternative Approach

- Determine features to be acquired and classification decision for each categorical variable $X_i$ in $G$

\[
J(R_i, D_{R_i}) = \mathbb{E} \left[ \sum_{k=1}^{R_i} e_k^i + \sum_{l=1}^{N_i} \sum_{m=1}^{N_i} M_{lm}^i P(D_{R_i} = l, C_i = C_m X_i) \right],
\]

- How to account for relationships between categorical variables? Propagate decisions across $G$
  - Initially, acquire features and make classification decisions for in–degree 0 nodes
  - Use such decisions to drive feature acquisition and classification decisions for each in–degree greater than 0 node


Figure 6: (a) Original Bayesian network; (b) Feature acquisition and classification for variables of in-degree 0; (c) Feature acquisition and classification for variables of in-degree > 0
### Some Results

TABLE II: Comparison of global accuracy (GA), mean accuracy (MA), and the average number of features (AF). The highest and the second highest accuracy values are bolded and gray-shaded, and gray-shaded, respectively. The smallest and the second smallest AF values are bolded and gray-shaded, and gray-shaded, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>ISEC</th>
<th>IC-NB</th>
<th>IC-ETANA</th>
<th>PC-NB</th>
<th>RCC</th>
<th>MD-KNN</th>
<th>IC-SVM</th>
<th>PC-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edm</td>
<td>GA</td>
<td>0.5985</td>
<td>0.3890</td>
<td>0.4668</td>
<td>0.5443</td>
<td>0.3905</td>
<td>0.3864</td>
<td>0.3578</td>
<td>0.4483</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.7401</td>
<td>0.6491</td>
<td>0.6500</td>
<td>0.7101</td>
<td>0.6952</td>
<td>0.6209</td>
<td>0.6755</td>
<td>0.7013</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>5.8654</td>
<td>16.0000</td>
<td>8.6333</td>
<td>16.0000</td>
<td>16.0000</td>
<td>16.0000</td>
<td>16.0000</td>
<td>16.0000</td>
</tr>
<tr>
<td>Voice</td>
<td>GA</td>
<td>0.8753</td>
<td>0.6897</td>
<td>0.8224</td>
<td>0.6824</td>
<td>0.2735</td>
<td>0.8359</td>
<td>0.7663</td>
<td>0.7220</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.9364</td>
<td>0.8243</td>
<td>0.8748</td>
<td>0.8345</td>
<td>0.5210</td>
<td>0.9142</td>
<td>0.8780</td>
<td>0.8514</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>2.5127</td>
<td>19.0000</td>
<td>2.2719</td>
<td>19.0000</td>
<td>19.0000</td>
<td>19.0000</td>
<td>19.0000</td>
<td>19.0000</td>
</tr>
<tr>
<td>Jura</td>
<td>GA</td>
<td>0.4402</td>
<td>0.3026</td>
<td>0.5381</td>
<td>0.4410</td>
<td>0.1388</td>
<td>0.2591</td>
<td>0.2562</td>
<td>0.2293</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.6352</td>
<td>0.5405</td>
<td>0.5815</td>
<td>0.6016</td>
<td>0.4764</td>
<td>0.4889</td>
<td>0.5397</td>
<td>0.4830</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>7.0517</td>
<td>9.0000</td>
<td>8.2394</td>
<td>9.0000</td>
<td>9.0000</td>
<td>9.0000</td>
<td>9.0000</td>
<td>9.0000</td>
</tr>
<tr>
<td>Song</td>
<td>GA</td>
<td>0.3299</td>
<td>0.2114</td>
<td>0.2509</td>
<td>0.2611</td>
<td>0.3082</td>
<td>0.4229</td>
<td>0.3471</td>
<td>0.3548</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.7134</td>
<td>0.6012</td>
<td>0.6709</td>
<td>0.6360</td>
<td>0.6802</td>
<td>0.7565</td>
<td>0.6728</td>
<td>0.6724</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>16.3172</td>
<td>98.0000</td>
<td>16.6072</td>
<td>98.0000</td>
<td>98.0000</td>
<td>98.0000</td>
<td>98.0000</td>
<td>98.0000</td>
</tr>
<tr>
<td>Flare</td>
<td>GA</td>
<td>0.8173</td>
<td>0.0277</td>
<td>0.7800</td>
<td>0.0463</td>
<td>0.8204</td>
<td>0.7802</td>
<td>0.8202</td>
<td>0.8202</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.9205</td>
<td>0.2194</td>
<td>0.8906</td>
<td>0.5736</td>
<td>0.9226</td>
<td>0.9035</td>
<td>0.9225</td>
<td>0.9225</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>1.3040</td>
<td>10.0000</td>
<td>7.0573</td>
<td>10.0000</td>
<td>10.0000</td>
<td>10.0000</td>
<td>10.0000</td>
<td>10.0000</td>
</tr>
<tr>
<td>Student</td>
<td>GA</td>
<td>0.6099</td>
<td>0.5742</td>
<td>0.5914</td>
<td>0.5815</td>
<td>0.5469</td>
<td>0.5208</td>
<td>0.5334</td>
<td>0.5021</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.7409</td>
<td>0.7227</td>
<td>0.5529</td>
<td>0.5918</td>
<td>0.6322</td>
<td>0.6346</td>
<td>0.6340</td>
<td>0.6084</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>8.4940</td>
<td>40.0000</td>
<td>14.0858</td>
<td>30.0000</td>
<td>30.0000</td>
<td>30.0000</td>
<td>30.0000</td>
<td>30.0000</td>
</tr>
<tr>
<td>Emotion</td>
<td>GA</td>
<td>0.3212</td>
<td>0.1820</td>
<td>0.2378</td>
<td>0.2731</td>
<td>0.0000</td>
<td>0.1164</td>
<td>0.2631</td>
<td>0.3203</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.7783</td>
<td>0.7391</td>
<td>0.7641</td>
<td>0.7700</td>
<td>0.6885</td>
<td>0.7026</td>
<td>0.7934</td>
<td>0.7718</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>8.5983</td>
<td>72.0000</td>
<td>15.3432</td>
<td>72.0000</td>
<td>72.0000</td>
<td>72.0000</td>
<td>72.0000</td>
<td>72.0000</td>
</tr>
<tr>
<td>Child</td>
<td>GA</td>
<td>0.5620</td>
<td>0.5209</td>
<td>0.5350</td>
<td>0.4800</td>
<td>0.3910</td>
<td>0.5098</td>
<td>0.3909</td>
<td>0.3909</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.8197</td>
<td>0.8156</td>
<td>0.8069</td>
<td>0.7783</td>
<td>0.7106</td>
<td>0.7799</td>
<td>0.7106</td>
<td>0.7106</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>4.4293</td>
<td>17.0000</td>
<td>5.8147</td>
<td>17.0000</td>
<td>17.0000</td>
<td>17.0000</td>
<td>17.0000</td>
<td>17.0000</td>
</tr>
<tr>
<td>Hepar2</td>
<td>GA</td>
<td>0.4200</td>
<td>0.0000</td>
<td>0.4170</td>
<td>0.0350</td>
<td>0.4180</td>
<td>0.4150</td>
<td>0.4230</td>
<td>0.4150</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.7807</td>
<td>0.4260</td>
<td>0.7757</td>
<td>0.4193</td>
<td>0.7813</td>
<td>0.7792</td>
<td>0.7813</td>
<td>0.7747</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>12.6213</td>
<td>67.0000</td>
<td>31.9470</td>
<td>67.0000</td>
<td>67.0000</td>
<td>67.0000</td>
<td>67.0000</td>
<td>67.0000</td>
</tr>
<tr>
<td>Sachs</td>
<td>GA</td>
<td>0.7920</td>
<td>0.7770</td>
<td>0.6000</td>
<td>0.5000</td>
<td>0.7920</td>
<td>0.7880</td>
<td>0.7920</td>
<td>0.7920</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.8420</td>
<td>0.8345</td>
<td>0.7250</td>
<td>0.5765</td>
<td>0.8426</td>
<td>0.8390</td>
<td>0.8420</td>
<td>0.8420</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>1.8575</td>
<td>8.0000</td>
<td>8.2285</td>
<td>9.0000</td>
<td>9.0000</td>
<td>9.0000</td>
<td>9.0000</td>
<td>9.0000</td>
</tr>
<tr>
<td>Insurance</td>
<td>GA</td>
<td>0.8270</td>
<td>0.0920</td>
<td>0.8100</td>
<td>0.6150</td>
<td>0.4320</td>
<td>0.6062</td>
<td>0.7240</td>
<td>0.7310</td>
</tr>
<tr>
<td></td>
<td>MA</td>
<td>0.9050</td>
<td>0.8350</td>
<td>0.9030</td>
<td>0.7840</td>
<td>0.5870</td>
<td>0.7841</td>
<td>0.8540</td>
<td>0.8520</td>
</tr>
<tr>
<td></td>
<td>AF</td>
<td>2.9115</td>
<td>25.0000</td>
<td>5.4565</td>
<td>25.0000</td>
<td>25.0000</td>
<td>25.0000</td>
<td>25.0000</td>
<td>25.0000</td>
</tr>
</tbody>
</table>
ML models cannot accurately predict all test instances

Problematic, especially in risk-sensitive applications (e.g., autonomous vehicles, medical diagnosis)

To the best of our knowledge, instance-wise feature acquisition assumes single loss function

How to jointly acquire the subset of features based on which each example is to be classified and the appropriate classifier to be used for this task?

Assess difficulty of classifying data instances to guide decision making process

Easy-to-classify data instances: few features and simple classifier

Hard-to-classify data instances: more features and powerful classifier
Problem Description

- \( X \triangleq [X_1, \ldots, X_F]^\top \): feature vector containing \( F \) features
- \( c_f \): cost of acquiring \( f \)th feature
- \( Y \in \{1, \ldots, N\} \): label
- \( C \triangleq \{C_1, \ldots, C_Z\} \): set of \( Z \) classifiers

**Objective**: jointly determine subset of features to be acquired, classifier to be used and the label of each example
Optimization Setup

- Introduce random variables
  - \( S \in \{0, \ldots, F\} \): last feature acquired before label assignment
  - \( U_S \in \{0, \ldots, Z\} \): classifier selected after \( S \) features have been acquired
  - \( D_S \in \{1, \ldots, N\} \): classification decision for data instance under consideration based on \( S \) features

\[
\min_{S, U_S, D_S} L(S, U_S, D_S)
\]

\[
L(S, U_S, D_S) = \mathbb{E} \left\{ \sum_{f=1}^{S} c_f + \sum_{z=1}^{Z} \lambda_z \mathbb{I}_{\{U_S = z\}} h \tilde{Z} + \gamma \mathbb{I}_{\{U_S = 0\}} \right. \\
\left. \times \sum_{j=1}^{N} \sum_{i=1}^{N} \Omega_{ij} P(D_S = j, Y = i) \right\},
\]
Optimum Solution

- $\phi_f \triangleq [\phi_f^1, \ldots, \phi_f^N]^T$: posterior probability vector with $\phi_f^i \triangleq P(Y = i|x_1, \ldots, x_f)$
- Optimum label assignment strategy
  \[ D_S^* = \min_{1 \leq j \leq N} [\Omega_j^T \phi_S]. \]
- Optimum classifier selection strategy
  \[ U_S^* = \min_{0 \leq t \leq Z} [\lambda_t H_S^t(\phi_S)]. \]
- Optimum feature acquisition strategy via dynamic programming
  \[ \bar{L}_f(\phi_f) = \min \left[ l(\phi_f), \bar{I}_f(\phi_f) \right] \]
  \[ l(\phi_f) = \min_{0 \leq t \leq Z} [\lambda_t H_f^t(\phi_f)] \]
  \[ \bar{I}_f(\phi_f) = c_{f+1} + \sum_{x_{f+1}} \bar{L}_{f+1}(\phi_{f+1}) \Pi_{f+1}^T(x_{f+1}) \phi_f \]
Intuition

Figure 7: Illustration of classifier selection and label assignment processes in the case of two label values (i.e., $N = 2$), a simple classifier (region A), and a single powerful classifier (region B).
SFCS Algorithm

Feature Acquisition

Start $f = 1$

Update $\phi_f$

Check $l(\phi_f) > \bar{I}(\phi_f)$

$f = f + 1$

Classifier Selection

$S^*$

Simple

$C_1$

$C_Z$

Compute $U_S^*$ use Eq. (8)

Label Assignment

$U_S^* = 0$

$U_S^* = 1$

$U_S^* = Z$

$U_S^* = D_S^*$

Predicted labels

Testing Set

Varying number of acquired features

Selected classifiers
Some Results

- Good balance between accuracy and average number of acquired features
- Classifier selection in instance-wise feature acquisition enhances accuracy, but in most cases, increases average number of acquired features
- Why does SFCS–DT performs worse than DT?
Some Results

Figure 8: Distribution of average Gini impurity reduction (GIR) per example. “All” denotes baselines that use all features (e.g., SVM, DT)

- Feature with higher GIR is more significant than a feature with lower GIR, since latter cannot be used to effectively separate labels
Some Results

Figure 9: Distribution of number of acquired features during testing for the Spambase dataset using SFCS–3X (NB, SVM, DT).

- Classify most instances using simple classifier with few features
- When number of acquired features increases, SFCS switches to other classifiers (difficult-to-classify instances)
Causal Feature Selection for Algorithmic Fairness [GSSV22]

- Algorithmic fairness is critical when supervised classification models are used to support decisions in high-stake domains
- Not discrimination-aware feature selection methods prefer features that improve accuracy
- Goal: identify subset of new features to include in a dataset without worsening its biases against protected groups
  - Meant to be used during training dataset creation time
  - Key challenge: one or more non-protected features can facilitate reconstruction of protected information (e.g., infer race from zip code)
  - Main idea: perform conditional independence tests between different subsets of features
Causal Feature Selection for Algorithmic Fairness [GSSV22]

- Input dataset comprises:
  - Target variable $Y$ (e.g., credit score)
  - Set of protected/sensitive features $S$ (e.g., gender and race)
  - Set of admissible features $A$ (e.g., expected monthly usage)
    - Protected variables can affect the outcome through admissible features
  - Features that are neither admissible nor sensitive (e.g., age and education)

- Two-phase method using conditional independence tests
  - Identify features that do not capture information about sensitive attributes
  - Ensure fairness even if features capture some information about sensitive attributes
Causal Feature Selection for Algorithmic Fairness [GSSV22]

- Find variables $X_i$ independent of $S$ by performing conditional independence test
- Variables whose paths from $S$ are blocked by $A$ do not provide any new information about $S$
  - Check if $X_i$ is conditionally independent of $S$ given $A$
- Variables $X_i$ not independent of $S$ even given $A$ can leak sensitive information
  - If independent of $Y$ given $A$, no effect on the classifier
- Any variable that is not independent of $S$ and $Y$ even after intervening on $A$ should not be added

\textbf{Algorithm 1} SeqSel

1: \textbf{Input:} Variables $A, S, X, Y$
2: $C_1 \leftarrow \phi$
3: \textbf{for} $X \in X$ \textbf{do}
4: \hspace{1em} \textbf{if} $\exists A \subseteq A \text{ such that } (X \perp S | A)$ \textbf{then}
5: \hspace{2em} $C_1 \leftarrow C_1 \cup \{X\}$
6: $C_2 \leftarrow \phi$
7: $X \leftarrow X \setminus C_1$
8: \textbf{for} $X \in X$ \textbf{do}
9: \hspace{1em} \textbf{if} $(X \perp Y | A \cup C_1)$ \textbf{then}
10: \hspace{2em} $C_2 \leftarrow C_2 \cup \{X\}$
11: \textbf{return} $C_1 \cup C_2$
Causal DAG $G$ captures functional dependencies between variables

- Variable $X_1$ causes $X_2$ iff $X_1 \rightarrow X_2$ in $G$
- Joint probability distribution can be decomposed similar to Bayesian networks

- Variables $X$ and $Y$ are $d$–separated given $Z$, if all paths between $X$ and $Y$ are blocked by $Z$
  - Ideally, the prediction and protected attributes should be $d$–separated in $G$

- do-operator: assign value $x$ to variable $X$ ($do(X) = x$) in $G'$ induced by $G$, with the difference that all incoming edges of $X$ have been removed

- A classifier is considered fair if for any collection of values $\alpha$ of $A$ and output $y'$
  $$P(Y' = y|do(S) = s, do(A) = \alpha) = P(Y' = y|do(S) = s', do(A) = \alpha), \forall A, S, Y'$$

- Testing for causal fairness requires fully specified causal graphs (not available in practice)
  - Use conditional mutual information instead
Causal Feature Selection for Algorithmic Fairness [GSSV22]

- Given $A$, $D = A \cup T$ is causally fair if the Bayes optimal predictor $Y'$, trained on $D$ satisfies causal fairness with respect to sensitive attributes $S$
- Goal: identify largest subset $T$ such that $Y'$, trained using these variables is fair
- New node $Y'$ is added to $G$
- All features that impact the classifier output are made parents of $Y'$
Large-scale classification tasks comprise hundreds, thousands, or even tens of thousands of class labels.

Class labels are structured (often in a tree).
- Class hierarchy divides the classification task into small and easy subtasks.

**Goal:** Feature selection for hierarchical classification tasks.
- Relevant features may differ among classes.
- Need to select different features for different subtasks.
Feature Selection for Hierarchical Classification [ZH$^{+}$19]

- Feature selection as penalized optimization
  
  \[
  \min_{W} L(XW, Y) + \lambda R(W)
  \]

  - Empirical loss $L$ (e.g., logistic, hinge, cross-entropy loss)
  
  - Regularizer $R$ and positive constant $\lambda$
  
  - Structural sparsity with $\ell_{2,1}$-norm

- **Goal**: minimize $\sum_{i=0}^{N} (\|X_i W_i - Y_i\|_F^2 + \lambda \|W_i\|_{2,1})$

  - Closed form solution obtained for least squares loss

- Feature weight matrix $W_i$ is computed for each internal node $i$

- Data instances of the $i$th node: $X_i = [x_1; x_2; \ldots; x_{m_i}]$
Feature Selection for Hierarchical Classification [ZHZN+19]

- **Top–down recursive strategy**
- **Node $i$th’s top–ranked (w.r.t $\|w^i_j\|_F$) features are selected**

**Algorithm 1. Hierarchical Feature Selection (Hier-FS)**

**Input**: Input data $X_i \in \mathbb{R}^{m_i \times n}$ and labels $Y \in \{0, 1\}^{m_i \times d}$, where $i = 0, 1, \ldots, N$, and $N$ is the number of internal nodes. To facilitate the calculation, we let $d$ be the maximum number of classes of internal nodes. Regularization parameter is $\lambda$, and the maximal iteration number is $T$.

**Output**: Matrix $W \in \mathbb{R}^{n \times d(N+1)}$.

1. Set $t = 0$ and initialize $W_t = [W_0, W_1, \ldots, W_N]$;
2. $W = [W_0, W_1, \ldots, W_N];$
3. while $t < T$ do
4.   for $i = 0 : N$ do
5.     Compute the diagonal matrix $D^{(t)}$ according to $d_{jj}^{(t)} = \frac{1}{2\|w_j^{(t)}\|_F^2}$
6.   end for
7.   // Update the root node and internal nodes.
8.   for $i = 0 : N$ do
9.     Update $W_i$ by $W_i^{(t+1)} = (X_i^T X_i + \lambda D^{(t)})^{-1} (X_i^T Y_i)$;
10. end for
11. Update $W^{(t+1)} = [W_0, W_1, \ldots, W_N];$
12. $t = t + 1$;
13. end while
14. return $W$;
Hierarchical regularization with parent–child relationship
- Parent–child classes are similar to each other; should share common features
- Relationship is incorporated into regularizer: $\sum_{i=1}^{N} ||W_i - W_{p_i}||_F^2$

Hierarchical regularization with sibling relationship
- Siblings come from different subtrees
- Discriminative features must be selected for each sibling
- Hilbert–Schmidt Independence Criterion to penalize dependence between selected features at sibling nodes

Hierarchical regularization with family relationship
- Both parent–child and sibling relationships between categories incorporated into the optimization problem
Summary and Conclusion
Still about Feature Selection vs Feature Acquisition?

- **Global Feature Selection**
  - Identify, during training, a sub-set of features (common across instances)
  - Online/streaming methods when full feature set unavailable at training

- **Instance–wise Feature Selection**
  - Identify, during testing, small sub-set of features for each data instance (varies between instances)
  - Given a test instance, all of its features must be available

- **Active Feature Acquisition**
  - During training (related to feature selection with missing values)
  - During testing, learned model is used

- **Instance–wise Feature Acquisition**
  - Different features acquired, during testing, for each data instance
  - Classification with costly features / Dynamic instance–wise feature acquisition
Feature Selection vs Feature Acquisition Visualized

(a) global feature selection

(b) streaming feature selection

(c) static instance-wise feature selection

(d) active feature acquisition

(e) classification with costly features
Key Takeaways

- Traditional feature selection is conducted during training
- Feature acquisition ≠ feature selection
  - can be performed either during training or testing
- Instance–wise feature selection ≠ instance–wise feature acquisition
- Both feature selection and feature acquisition approaches face significant challenges
- Instance–wise feature acquisition has broader implications to ML
(Non Exhaustive List of) Topics This Tutorial Didn’t Cover

- Feature acquisition in both training and testing \[\text{DMW10}\]
- Group feature acquisition during testing \[\text{AJD24}\]
- Multiview/multimodal feature selection \[\text{YGSC15, LMF16, KAH20}\] and acquisition \[\text{NZC20}\]
- Active feature acquisition for time series data \[\text{LO21, BBS22, KCV}^+23\]
- Feature selection (prompting) for large language models
- Knowledge–driven feature acquisition
- Causality and feature selection
- Feature selection/acquisition for non–linear models
  - Quantifying feature importance is difficult
  - Interpreting findings becomes challenging
Our coverage of state-of-the-art and challenges we identify are not exhaustive

The slides can be found at: https://www.cs.albany.edu/~cchelmis/tutorials/sdm/2024/

Suggested citation:
Daphney-Stavroula Zois, Charalampos Chelmis, “From Feature Selection to Instance-wise Feature Acquisition”, Minitutorial at SIAM International Conference on Data Mining (SDM), Houston, TX, April 2024.
Acknowledgements

- We thank our collaborators and funding agencies for making this work possible
- We thank authors who graciously agreed to provide us with material to include in this tutorial
- We thank the conference tutorial committee for giving us the opportunity to present at SDM
- We thank you, the audience, for your attention
- We welcome your feedback and suggestions
References

Vedang Asgaonkar, Aditya Jain, and Abir De.
Generator Assisted Mixture of Experts For Feature Acquisition in Batch.

Josh Attenberg, Prem Melville, Foster Provost, and Maytal Saar-Tsechansky.
Selective data acquisition for machine learning.

Andrew Arnt and Shlomo Zilberstein.
Attribute measurement policies for time and cost sensitive classification.

Maik Büttner, Christian Beyer, and Myra Spiliopoulou.
Reducing Missingness in a Stream through Cost-Aware Active Feature Acquisition.
In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2022.

Valentina Bayer-Zubek and Thomas G Dietterich.
Integrating learning from examples into the search for diagnostic policies.

Gabriella Contardo, Ludovic Denoyer, and Thierry Artières.
Recurrent neural networks for adaptive feature acquisition.
Xiaoyong Chai, Lin Deng, Qiang Yang, and Charles X Ling.  
Test-cost sensitive naive Bayes classification.  

Mumin Cebe and Cigdem Gunduz-Demir.  
Test-cost sensitive classification based on conditioned loss functions.  

Ziheng Chen, Jin Huang, Hongshik Ahn, and Xin Ning.  
Costly features classification using Monte Carlo tree search.  

Koby Crammer, Alex Kulesza, and Mark Dredze.  
Adaptive regularization of weight vectors.  

Ian Connick Covert, Wei Qiu, Mingyu Lu, Na Yoon Kim, Nathan J White, and Su-In Lee.  
Learning to maximize mutual information for dynamic feature selection.  

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan.  
Learning to explain: An information-theoretic perspective on model interpretation.  
Datum-wise classification: a sequential approach to sparsity.

Sequential approaches for learning datum-wise sparse representations.

Mark Dredze, Koby Crammer, and Fernando Pereira.
Confidence-weighted linear classification.

Marie Desjardins, James MacGlashan, and Kiri L Wagstaff.
Confidence-based feature acquisition to minimize training and test costs.

Sachini Piyoni Ekanayake, Yasitha Warahena Liyanage, and Daphney-Stavroula Zois.
Dynamic feature selection for classification in structured environments.

Sachini Piyoni Ekanayake and Daphney-Stavroula Zois.
Datum–Wise Inference in Structured Environments.


Causal feature selection for algorithmic fairness.  

Icebreaker: Element-wise efficient information acquisition with a bayesian deep latent gaussian model.  

He He, Hal Daumé III, and Jason Eisner.  
Cost-sensitive dynamic feature selection.  
In ICML Inferning Workshop, 2012.

Xuegang Hu, Peng Zhou, Peipei Li, Jing Wang, and Xindong Wu.  
A survey on online feature selection with streaming features.  

Shihao Ji and Lawrence Carin.  
Cost-sensitive feature acquisition and classification.  

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý.  
Classification with costly features using deep reinforcement learning.  
Jaromír Janisch, Tomáš Pevný, and Viliam Lisý.
Classification with costly features as a sequential decision-making problem. 

Majid Komeili, Narges Armanfard, and Dimitrios Hatzinakos.
Multiview feature selection for single-view classification.

Jannik Kossen, Cătălina Cangea, Eszter Vértes, Andrew Jaegle, Viorica Patraucean, Ira Ktena, Nenad Tomasev, and Danielle Belgrave.

David P Kao, James D Lewsey, Inder S Anand, Barry M Massie, Michael R Zile, Peter E Carson, Robert S McKelvie, Michel Komajda, John JV McMurray, and JoAnn Lindenfeld.
Characterization of subgroups of heart failure patients with preserved ejection fraction with possible implications for prognosis and treatment response.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra.
Planning and acting in partially observable stochastic domains.
Athresh Karanam and Sriraam Natarajan.  
On test-time active feature selection through tractable acquisition functions.  

Dugang Liu, Pengxiang Cheng, Hong Zhu, Xing Tang, Yanyu Chen, Xiaoting Wang, Weike Pan, Zhong Ming, and Xiuqiang He.  
DIWIFT: Discovering Instance-wise Influential Features for Tabular Data.  

Continuous control with deep reinforcement learning.  

Interpretability in the context of sequential cost-sensitive feature acquisition, author=Liyanage, Yasitha Warahena and Zois, Daphney–Stavroula.  

Scott M Lundberg and Su-In Lee.  
A unified approach to interpreting model predictions.  

Hongfu Liu, Haiyi Mao, and Yun Fu.  
Robust multi-view feature selection.  
In IEEE 16th International Conference on Data Mining (ICDM), pages 281–290, 2016.
Yang Li and Junier Oliva.  
Active feature acquisition with generative surrogate models.  

Yasitha Warahena Liyanage and Daphney-Stavroula Zois.  

Yasitha Warahena Liyanage, Daphney-Stavroula Zois, and Charalampos Chelmis.  
Dynamic instance-wise classification in correlated feature spaces.  

Yasitha Warahena Liyanage, Daphney-Stavroula Zois, and Charalampos Chelmis.  
Dynamic instance-wise joint feature selection and classification.  

Kachuee Mohammad, Goldstein Orpaz, Kärkkäinen Kimmo, Darabi Sajad, and Sarrafzadeh Majid.  
Opportunistic learning: Budgeted cost-sensitive learning from data streams.  

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare.  
Safe and efficient off-policy reinforcement learning.  
Prem Melville, Maytal Saar-Tsechansky, Foster Provost, and Raymond Mooney.  
An expected utility approach to active feature-value acquisition.  

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang.  
EDDI: Efficient Dynamic Discovery of High-Value Information with Partial VAE.  

Aria Masoomi, Chieh Wu, Tingting Zhao, Zifeng Wang, Peter Castaldi, and Jennifer Dy.  
Instance-wise feature grouping.  

Imara Nazar, Daphney-Stavroula Zois, and Charalampos Chelmis.  
Knowing When to Stop: Joint Heterogeneous Feature Selection and Classification.  

Simon Perkins, Kevin Lacker, and James Theiler.  
Grafting: Fast, incremental feature selection by gradient descent in function space.  

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin.  
"why should i trust you?" explaining the predictions of any classifier.  
Hajin Shim, Sung Ju Hwang, and Eunho Yang.  
Joint active feature acquisition and classification with variable-size set encoding.  

Victor S Sheng and Charles X Ling.  
Feature value acquisition in testing: a sequential batch test algorithm.  

Find it at http://pslcedatashop.web.cmu.edu/KDDCup/downloads.jsp.

Kirill Trapeznikov and Venkatesh Saligrama.  
Supervised sequential classification under budget constraints.  

Hado Van Hasselt, Arthur Guez, and David Silver.  
Deep reinforcement learning with double q-learning.  

Joseph Wang, Tolga Bolukbasi, Kirill Trapeznikov, and Venkatesh Saligrama.  
Model selection by linear programming.  
Steve Webb, James Caverlee, and Calton Pu.
Introducing the webb spam corpus: Using email spam to identify web spam automatically.
In CEAS, 2006.

Yue Wu, Steven CH Hoi, Tao Mei, and Nenghai Yu.
Large-scale online feature selection for ultra-high dimensional sparse data.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas.
Dueling network architectures for deep reinforcement learning.

Joseph Wang, Kirill Trapeznikov, and Venkatesh Saligrama.
An lp for sequential learning under budgets.

Xindong Wu, Kui Yu, Wei Ding, Hao Wang, and Xingquan Zhu.
Online feature selection with streaming features.

Jialei Wang, Peilin Zhao, Steven CH Hoi, and Rong Jin.
Online feature selection and its applications.
Zhixiang Xu, Matt J Kusner, Kilian Q Weinberger, Minmin Chen, and Olivier Chapelle.
Classifier cascades and trees for minimizing feature evaluation cost.

Qi Xiao, Hebi Li, Jin Tian, and Zhengdao Wang.
Group-wise feature selection for supervised learning.

Qi Xiao and Zhengdao Wang.
Mixture of deep neural networks for instancewise feature selection.

Wanqi Yang, Yang Gao, Yinghuan Shi, and Longbing Cao.
MRM-lasso: A sparse multiview feature selection method via low-rank analysis.

Jinsung Yoon, James Jordon, and Mihaela Van der Schaar.
In International conference on learning representations, 2018.

Kui Yu, Xindong Wu, Wei Ding, and Jian Pei.
Scalable and accurate online feature selection for big data.
ACM Transactions on Knowledge Discovery from Data (TKDD), 11(2):1–39, 2016.
Jing Zhou, Dean Foster, Robert Stine, and Lyle Ungar.  
Streaming feature selection using alpha-investing.  

Peng Zhou, Xuegang Hu, Peipei Li, and Xindong Wu.  
OFS-Density: A novel online streaming feature selection method.  

Peng Zhou, Xuegang Hu, Peipei Li, and Xindong Wu.  
Online streaming feature selection using adapted neighborhood rough set.  
Information Sciences, 481:258–279, 2019.

Odin: Optimal discovery of high-value information using model-based deep reinforcement learning.  
In ICML Real-world Sequential Decision Making Workshop, 2019.

Hong Zhao, Qinghua Hu, Pengfei Zhu, Yu Wang, and Ping Wang.  
A recursive regularization based feature selection framework for hierarchical classification.  
IEEE Transactions on Knowledge and Data Engineering, 33(7):2833–2846, 2019.

Peng Zhou, Shu Zhao, Yuanting Yan, and Xindong Wu.  
Online scalable streaming feature selection via dynamic decision.  
ACM Transactions on Knowledge Discovery from Data (TKDD), 16(5):1–20, 2022.