Feature Selection

• In many applications, we often encounter a very large number of potential features that can be used.
• Which subset of features should be used for the best classification?
• Need for a small number of discriminative features
  • To avoid “curse of dimensionality”
  • To reduce feature measurement cost
  • To reduce computational burden
• Given an \( n \times d \) pattern matrix (\( n \) patterns in \( d \)-dimensional space), generate an \( n \times m \) pattern matrix, where \( m << d \)
Feature Selection vs. Extraction

- Both are collectively known as **dimensionality reduction**
- **Selection**: choose the **best** subset of size $m$ from the available $d$ features
- **Extraction**: given $d$ features (set $Y$), extract $m$ new features (set $X$) by **linear or non-linear combination** of all the $d$ features
  - Linear feature extraction: $X = TY$, where $T$ is a $m \times d$ matrix
  - Non-linear feature extraction: $X = f(Y)$
- New extracted features may not have physical interpretation or meaning
- Examples of linear feature extraction
  - PCA (unsupervised); LDA (supervised)
- Goals for feature selection & extraction: simplify classifier complexity and improve the classification accuracy,
Feature Selection

- How to find the best subset of size $m$?
- Recall, best means a classifier based on these $m$ features has the lowest probability of error.
- Simplest approach an exhaustive search: (i) select a criterion fn., (ii) evaluate ALL possible subsets.
- Computationally prohibitive
  - For $d=24$ and $m=12$, there are about 2.7 million possible feature subsets! Cover & Van Campenhout (IEEE SMC, 1977) showed that to guarantee the best subset of size $m$ from the available set of size $d$, one must examine all possible subsets of size $m$.
- Heuristics have been used to avoid exhaustive search.
- How to evaluate the subsets?
  - Error rate; but then which classifier should be used?
  - Distance measure; Mahalanobis, divergence,…
- Feature selection is an optimization problem.
• Feature selection enables combining features from different data models
• Potential difficulties in feature selection (i) small sample size, (ii) what criterion function to use
• Let $Y$ be the original set of features and $X$ is the selected subset
• Feature selection criterion for the set $X$ is $J(X)$; large value of $J$ indicates a better feature subset; problem is to find subset $X$ such that

$$J(X) = \max_{Z \subseteq Y, |Z|=d} J(Z)$$
Taxonomy of Feature Selection Algorithms

feature selection

SPR

ANN
node pruning

suboptimal

optimal
exhaustive search
branch-and-bound

single solution
many solutions

deterministic
stochastic
deterministic
stochastic

PTA(l,r)
SA
beam search
GA
floating
Max-Min

Fig. 1. A taxonomy of feature selection algorithms.
Deterministic Single-Solution Methods

• Begin with a single solution (feature subset) & iteratively add or remove features until some termination criterion is met
• Also known as sequential methods
  – Bottom up (forward method): begin with an empty set & add features
  – Top-down (backward method): begin with a full set & delete features
• These “greedy” methods do not examine all possible subsets, so no guarantee of finding the optimal subset
• Pudil introduced two floating selection methods: SFFS, SFBS
• 15 feature selection methods listed in Table 1 were evaluated

| TABLE 1 |
| Feature Selection Algorithms Used in Experimental Evaluation |
|---------|--------|---------|---------|
| SFS     | SBS    | GSFS(2) | GSBS(2) |
| GSFS(3) | GSBS(3)| SFFS    | SFBS    |
| PTA((1), (2)) | PTA((1), (3)) | PTA((2), (3)) |
| BB      | MM     | GA      | NP      |
Naiive Method

• Sort the given d features in order of their prob. of correct recognition

• Select the top m features from this sorted list

• *Disadvantage*: Feature correlation is not considered; best pair of features may not even contain the best individual feature
Sequential Forward Selection (SFS)

- Start with the empty set, $X=0$
- Repeatedly add the most significant feature with respect to $X$
- Disadvantage: Once a feature is retained, it cannot be discarded; nesting problem
Sequential Backward Selection (SBS)

- Start with the full set, $X=Y$
- Repeatedly delete the least significant feature in $X$
- **Disadvantage**: SBS requires more computation than SFS; Nesting problem
Generalized Sequential Forward Selection (GSFS\((m)\))

- Start with the empty set, \(X=0\)
- Repeatedly add the most significant \(m\)-subset of \((Y - X)\) (found through exhaustive search)
Generalized Sequential Backward Selection (GSBS($m$))

• Start with the full set, $X=Y$

• Repeatedly delete the least significant $m$-subset of $X$ (found through exhaustive search)
Sequential Forward Floating Search (SFFS)

- **Step 1: Inclusion.** Use the basic SFS method to select the most significant feature with respect to $X$ and include it in $X$. Stop if $d$ features have been selected, otherwise go to step 2.

- **Step 2: Conditional exclusion.** Find the least significant feature $k$ in $X$. If it is the feature just added, then keep it and return to step 1. Otherwise, exclude the feature $k$. Note that $X$ is now better than it was before step 1. Continue to step 3.

- **Step 3: Continuation of conditional exclusion.** Again find the least significant feature in $X$. If its removal will (a) leave $X$ with at least 2 features, and (b) the value of $J(X)$ is greater than the criterion value of the best feature subset of that size found so far, then remove it and repeat step 3. When these two conditions cease to be satisfied, return to step 1.
Experimental Results

• 20-dimensional 2-class Gaussian data with a common covariance matrix

• **Goodness of features** (criterion function) is measured by the Mahalanobis distance; recall that in the common covariance matrix case, the prob. of error is inversely proportional to the Mahalanobis distance

• Forward search methods are faster than backward methods

• Performance of floating method (SFFS) is comparable to branch & bound methods, but SFFS is significantly faster

• “Nesting” problem: The optimal three feature subset is not contained in the optimal four feature subset!

• Feature selection is an **off-line process**, so large cost during training can be afforded
Experimental Results

Fig. 2. Performance and execution times of selected algorithms on synthetic 2-class Gaussian data set.
Selection of Texture Features

• Selection of texture features for classifying Synthetic Aperture Radar (SAR) images
• A total of 18 different features/pixel from 4 different models
• Can classification error be reduced by feature selection
• 22,000 samples (pixels) from 5 classes; equally split for training/test
• Can the classification error be reduced by feature selection?

TABLE 2
<table>
<thead>
<tr>
<th>#</th>
<th>feature</th>
<th>model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mean</td>
<td>local statistics</td>
</tr>
<tr>
<td>2</td>
<td>$\theta_1$</td>
<td>MAR</td>
</tr>
<tr>
<td>3</td>
<td>$\theta_2$</td>
<td>MAR</td>
</tr>
<tr>
<td>4</td>
<td>$\theta_3$</td>
<td>MAR</td>
</tr>
<tr>
<td>5</td>
<td>$\sigma$ (variance)</td>
<td>MAR</td>
</tr>
<tr>
<td>6</td>
<td>mean (logarithmic)</td>
<td>MAR</td>
</tr>
<tr>
<td>7</td>
<td>angular second moment</td>
<td>GLCM</td>
</tr>
<tr>
<td>8</td>
<td>contrast</td>
<td>GLCM</td>
</tr>
<tr>
<td>9</td>
<td>inverse difference moment</td>
<td>GLCM</td>
</tr>
<tr>
<td>10</td>
<td>entropy</td>
<td>GLCM</td>
</tr>
<tr>
<td>11</td>
<td>inertia</td>
<td>GLCM</td>
</tr>
<tr>
<td>12</td>
<td>cluster shade</td>
<td>GLCM</td>
</tr>
<tr>
<td>13</td>
<td>power-to-mean ratio</td>
<td>local statistics</td>
</tr>
<tr>
<td>14</td>
<td>skewness</td>
<td>local statistics</td>
</tr>
<tr>
<td>15</td>
<td>kurtosis</td>
<td>local statistics</td>
</tr>
<tr>
<td>16</td>
<td>contrast (from Skriver, 1987)</td>
<td>fractal</td>
</tr>
<tr>
<td>17</td>
<td>lacunarity</td>
<td>local statistics</td>
</tr>
<tr>
<td>18</td>
<td>dimension</td>
<td>fractal</td>
</tr>
</tbody>
</table>

Fig. 3. Sample SAR images, with corresponding ground truth image at lower right.
Performance of SFFS on Texture Features

- Best individual texture model for this data is the MAR model
- Pooling features from different models and then applying feature selection results in an accuracy of 89.3% by the 1NN method
- Selected subset has representative feature from every model; 5-feature subset selected contains features from 3 different models

Fig. 4. Recognition rates of SFFS method on texture features.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Recognition rate (%)</th>
<th>Optimal number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1NN</td>
<td>89.3</td>
<td>12</td>
</tr>
<tr>
<td>3NN</td>
<td>88.4</td>
<td>11</td>
</tr>
</tbody>
</table>
Effect of Training Set Size on Feature Selection

• How reliable are the feature selection results in small sample size situations?
• Would the error in estimating covariance (for Mahalanobis distance) affect the feature selection performance?
• Feature selection on the Trunk data with varying sample size
• 20-dim data from distributions below; n in [10, 5000]
• Feature selection quality: no. of common features in the subset selected by SFFS and by the optimal method
• For n = 20, B&B selected the subset
  \{1,2,4,7,9,12,13,14,15,18\};
• optimal subset is \{1,2,3,4,5,6,7,8,9,10\}

\[
p(x | \omega_1) \sim N(\mu, I) \quad p(x | \omega_2) \sim N(-\mu, I)
\]

(2)

\[
\mu = \begin{bmatrix} \frac{1}{\sqrt{1}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \ldots \end{bmatrix}^T
\]

(3)
Fig. 5. Quality of selected feature subsets as a function of the size of training data.