Topics

- Feature selection
  - motivation, examples
  - information value, greedy selection, regularization

- Combination methods
  - forward/backward fitting
  - boosting
Feature selection

- Suppose we consider only a finite collection of possible basis functions, $\phi_i(x)$, $i = 1, \ldots, m$, such as the input components $\phi_i(x) = x_i$.

- We try to find a small subset $S$ of basis functions that are sufficient to solve the (regression or) classification problem:

$$P(y = 1|x, w) = g\left(w_0 + \sum_{i=1}^{k} w_i \phi_{s_i}(x)\right)$$

where the indexes $S = \{s_1, \ldots, s_k\}$ identify the selected basis functions.
Feature selection

• Suppose we consider only a finite collection of possible basis functions, \( \phi_i(x), i = 1, \ldots, m \), such as the input components \( \phi_i(x) = x_i \).

• We try to find a small subset \( S \) of basis functions that are sufficient to solve the (regression or) classification problem:

\[
P(y = 1|x, w) = g\left(w_0 + \sum_{i=1}^{k} w_i \phi_{s_i}(x)\right)
\]

where the indexes \( S = \{s_1, \ldots, s_k\} \) identify the selected basis functions.

• There are many ways to find appropriate basis functions:
  – information value
  – greedy selection
  – regularization
Information value

- Let’s first try to select the basis functions independently of the classifier, i.e., gauge how “informative” they are in general about the class label.

- Text classification example: $x$ is a document and the basis functions $\phi_1(x), \ldots, \phi_m(x)$ are “word indicators”

$$
\phi_i(x) = \begin{cases} 
1, & \text{if document } x \text{ contains word } i \\
0, & \text{otherwise}
\end{cases}
$$

- each document is represented by a binary vector

$$
\phi(x) = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 1 \end{bmatrix}^T
$$

- we will derive a score for each feature (bit) based on how much information it contains about the class label
Let’s focus on a single feature, e.g., the first one

\[
\phi_1 : 0 \ 1 \ 0 \ \ldots \ 1 \\
y : -1 \ -1 \ 1 \ \ldots \ 1
\]

To assess how the feature values relate to the labels we can calculate the frequency of occurrence of different combinations of values: \(\hat{P}(y), \hat{P}(\phi_1), \hat{P}(\phi_1, y)\).

For example

\[
\hat{P}(\phi_1 = 0, y = 1) = \frac{\text{# of docs such that } \phi_1(x) = 0 \text{ and } y = 1}{n}
\]
Let’s focus on a single feature, e.g., the first one

\[ \phi_1 : 0 \ 1 \ 0 \ \ldots \ 1 \]

\[ y : -1 \ -1 \ 1 \ \ldots \ 1 \]

To assess how the feature values relate to the labels we can calculate the frequency of occurrence of different combinations of values: \( \hat{P}(y) \), \( \hat{P}(\phi_1) \), \( \hat{P}(\phi_1, y) \).

The \textit{mutual information} score for each feature is given by:

\[
I(\phi_1; y) = \sum_{\phi_1 \in \{0,1\}} \sum_{y \in \{-1,1\}} \hat{P}(\phi_1, y) \log_2 \frac{\hat{P}(\phi_1, y)}{\hat{P}(y) \hat{P}(\phi_1)}
\]

This score is zero if the label is independent of the feature value; large (but \( \leq 1 \)) if they are deterministically related.
Selection by information value

- We rank the features according to their mutual information scores (in the descending order of the score):

\[
I(\phi_1; y) = \sum_{\phi_1 \in \{0, 1\}} \sum_{y \in \{-1, 1\}} \hat{P}(\phi_1, y) \log_2 \frac{\hat{P}(\phi_1, y)}{\hat{P}(y) \hat{P}(\phi_1)}
\]

- how many features to include?
- redundant features?
- coordination among the features?
- which classifier can make use of these features?
Greedy selection

1. Find \( s_1 \) and \( \mathbf{w} = [w_0, w_1]^T \) such that

\[
P(y = 1|\mathbf{x}, \mathbf{w}) = g\left( w_0 + w_1 \phi_{s_1}(\mathbf{x}) \right)
\]

leads to the best classifier.
Greedy selection

1. Find $s_1$ and $\mathbf{w} = [w_0, w_1]^T$ such that

$$P(y = 1 | \mathbf{x}, \mathbf{w}) = g \left( w_0 + w_1 \phi_{s_1}(\mathbf{x}) \right)$$

leads to the best classifier.

2. Find $s_2$ and $\mathbf{w} = [w_0, w_1, w_2]^T$ such that

$$P(y = 1 | \mathbf{x}, \mathbf{w}) = g \left( w_0 + w_1 \phi_{s_1}(\mathbf{x}) + w_2 \phi_{s_2}(\mathbf{x}) \right)$$

gives the best performing classifier.
Greedy selection

1. Find $s_1$ and $\mathbf{w} = [w_0, w_1]^T$ such that

$$P(y = 1|x, \mathbf{w}) = g\left( w_0 + w_1 \phi_{s_1}(x) \right)$$

leads to the best classifier.

2. Find $s_2$ and $\mathbf{w} = [w_0, w_1, w_2]^T$ such that

$$P(y = 1|x, \mathbf{w}) = g\left( w_0 + w_1 \phi_{s_1}(x) + w_2 \phi_{s_2}(x) \right)$$

gives the best performing classifier.

3. Etc.

   - stopping criterion?
   - over-fitting?
Regularization

- We can also consider all of the basis functions at once

\[ P(y = 1|x, w) = g\left( w_0 + w_1\phi_1(x) + \ldots + w_m\phi_m(x) \right) \]

and introduce a regularization penalty that tries to set the weights to zero unless the corresponding basis functions are useful.
Regularization

- We can also consider all of the basis functions at once

\[ P(y = 1|x, \mathbf{w}) = g\left( w_0 + w_1 \phi_1(x) + \ldots + w_m \phi_m(x) \right) \]

and introduce a regularization penalty that tries to set the weights to zero unless the corresponding basis functions are useful.

\[
J(\mathbf{w}; \lambda) = \sum_{i=1}^{n} - \log P(y_i|x, \mathbf{w}) + \lambda \sum_{i=1}^{m} |w_i|
\]

In other words, we regularize the 1-norm (not Euclidean norm) of the weights; \( w_0 \) is not penalized.
Regularization

- The effect of the regularization penalty depends on its derivative at $w \approx 0$

$$J(w; \lambda) = \sum_{i=1}^{n} - \log P(y_i|x, w) + \lambda \sum_{i=1}^{m} |w_i|$$

$w^2/2$ versus $|w|$
Combination of methods

- Similarly to feature selection we can select simple “weak” classification or regression methods and combine them into a single “strong” method

- Example techniques
  - forward fitting (regression)
  - boosting (classification)
Combination of regression methods

- We want to combine multiple "weak" regression methods into a single "strong" method

\[ f(x) = f(x; \theta_1) + \ldots + f(x; \theta_m) \]

- Suppose we are given a family simple regression methods

\[ f(x; \theta) = w \phi_k(x) \]

where \( \theta = \{k, w\} \) specifies the identity of the basis function as well as the associated weight.

- **Forward-fitting**: sequentially introduce new simple regression methods to reduce the remaining prediction error
Forward fitting cont’d

Simple family: $f(x; \theta) = w\phi_k(x), \theta = \{k, w\}$

- We can fit each new component to reduce the prediction error; in each iteration we solve the same type of estimation problem

Step 1: $\hat{\theta}_1 \leftarrow \arg\min_{\theta} \sum_{i=1}^{n} (y_i - f(x_i; \theta))^2$
Forward fitting cont’d

Simple family: $f(x; \theta) = w\phi_k(x)$, $\theta = \{k, w\}$

- We can fit each new component to reduce the prediction error; in each iteration we solve the same type of estimation problem

  \[
  \text{Step 1: } \hat{\theta}_1 \leftarrow \arg\min_{\theta} \sum_{i=1}^{n} (y_i - f(x_i; \theta))^2
  \]

  \[
  \text{Step 2: } \hat{\theta}_2 \leftarrow \arg\min_{\theta} \sum_{i=1}^{n} (y_i - f(x_i; \hat{\theta}_1) - f(x_i; \theta))^2
  \]
Forward fitting cont’d

Simple family: \( f(x; \theta) = w \phi_k(x) \), \( \theta = \{k, w\} \)

- We can fit each new component to reduce the prediction error; in each iteration we solve the same type of estimation problem

  Step 1: \( \hat{\theta}_1 \leftarrow \arg\min_{\theta} \sum_{i=1}^{n} (y_i - f(x_i; \theta))^2 \)

  Step 2: \( \hat{\theta}_2 \leftarrow \arg\min_{\theta} \sum_{i=1}^{n} (y_i - f(x_i; \hat{\theta}_1) - f(x_i; \theta))^2 \)

  Step 3: \( \ldots \)

- The resulting combined regression method

\[
\hat{f}(x) = f(x; \hat{\theta}_1) + \ldots + f(x; \hat{\theta}_m)
\]

has much lower (training) error.
Forward fitting: example

\[ f(x; \theta) = w x^k, \text{ where } \theta = \{w, k\}. \]