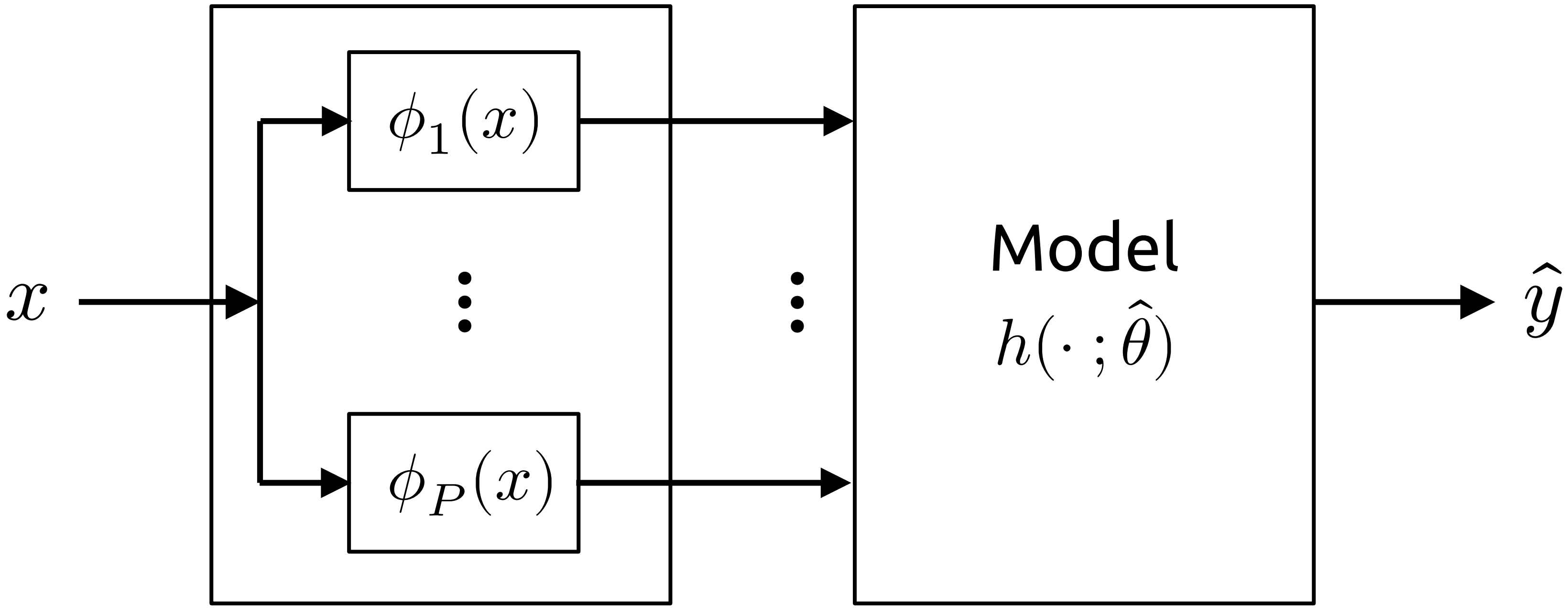
Statistics and Data Science for Engineers
E178 / ME276DS

# Lab04:
# Feature selection




$\mathcal{P} = \{\ \phi_1\ , \ldots ,\ \phi_P\ \}$ … set of all feature functions

# Exhaustive subset selection

subsets of size 1: $\{\{\phi_1\}, \{\phi_2\}, \dots, \{\phi_P\}\}$

subsets of size 2: $\{\{\phi_1, \phi_2\}, \{\phi_1, \phi_3\}, \dots, \{\phi_{P-1}, \phi_P\}\}$

⋮

subsets of size P: $\{\{\phi_1, \phi_2, \dots, \phi_P\}\}$

In total, train $2^P$ models and choose the best.

**Pseudocode**: Exhaustive search

for $k = 1 \ldots P$ :

for $\mathcal{A}_\kappa$ in $\{$all $k$-sized subsets of $\mathcal{P}\}$:

$$\underline{\hat{\theta}}_\kappa = \text{train}(\mathcal{A}_\kappa, \mathcal{D}_{\text{train}})$$

$$\ell_\kappa = \text{perf}(\mathcal{A}_\kappa, \underline{\hat{\theta}}_\kappa, \mathcal{D}_{\text{val}})$$

$$\kappa^* = \text{argbest}(\{\ell_\kappa\})$$

$$\mathcal{S}_k = \mathcal{A}_{\kappa^*}$$

... $\mathcal{S}_k =$ best set of $k$ features.

$$\mathcal{S}^* = \text{best of } \{\mathcal{S}_k\}_P$$

$$\underline{\hat{\theta}}^* = \text{train}(\mathcal{S}^*, \mathcal{D}_{\text{train}})$$

$$\ell^* = \text{perf}(\mathcal{S}^*, \underline{\hat{\theta}}^*, \mathcal{D}_{\text{test}})$$

## Forward stepwise selection

$\boxed{\text{Assumption: } \mathcal{S}_k \subset \mathcal{S}_{k+1}}$

Initialize: $\mathcal{S}_0 = \{\}$

| | Unchosen features $(\mathcal{P} \backslash \mathcal{S}_{k-1})$ | Best subset of size $k$ |
|---|---|---|
| $k=1$ | $\{\phi_1, \phi_2, \phi_3, \dots, \phi_{P-1}, \phi_P\}$ | $\mathcal{S}_1 = \{\phi_3\}$ |
| $k=2$ | $\{\phi_1, \phi_2, \dots, \phi_{P-1}, \phi_P\}$ | $\mathcal{S}_2 = \{\phi_3, \phi_{P-1}\}$ |
| $k=3$ | $\{\phi_1, \phi_2, \dots, \phi_P\}$ | $\mathcal{S}_3 = \{\phi_3, \phi_{P-1}, \phi_2\}$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $k=P$ | $\{\}$ | $\mathcal{S}_P = \mathcal{P}$ |

# **Pseudocode**: Forward stepwise selection

$\mathcal{S}_0 = \{\}$

for $k = 1 \dots P$ :

for $\kappa, \phi_p \in$ enumerate$(\mathcal{P} \backslash \mathcal{S}_{k-1})$

$\mathcal{A}_\kappa = \mathcal{S}_{k-1} \cup \phi_p$

$\hat{\theta}_\kappa =$ train$(\mathcal{A}_\kappa, \mathcal{D}_{\text{train}})$

$\ell_\kappa =$ perf$(\mathcal{A}_\kappa, \hat{\theta}_\kappa, \mathcal{D}_{\text{val}})$

$\kappa^* =$ argbest$(\{\ell_\kappa\})$

$\mathcal{S}_k = \mathcal{A}_{\kappa^*}$

$\mathcal{S}^* =$ best of $\{\mathcal{S}_k\}_P$

$\hat{\theta}^* =$ train$(\mathcal{S}^*, \mathcal{D}_{\text{train}})$

$\ell^* =$ perf$(\mathcal{S}^*, \hat{\theta}^*, \mathcal{D}_{\text{test}})$

```
curlyS = [set() for i in range(P+1)]
ellk = np.full(P+1,np.inf)

for k in range(1,P+1):

    curlyA = [set() for i in range(P-k+1)]
    ellkappa = np.full(P-k+1,np.inf)

    for kappa, phip in enumerate(curlyP-curlyS[k-1]):
        curlyA[kappa] = curlyS[k-1].union({phip})
        theta0hat, theta1hat = train( curlyA[kappa] , Dtrain)
        ellkappa[kappa] = perf(curlyA[kappa], theta0hat, theta1hat,
                               Dvalidate)

    kappastar = ellkappa.argmin()
    curlyS[k] = curlyA[kappastar]
    ellk[k] = ellkappa[kappastar]

kstar = ellk.argmin()
Sstar = curlyS[kstar]
theta0star, theta1star = train(Sstar, Dtrain)
ellstar = perf(Sstar, theta0star, theta1star, Dtest)

# Store the results
f_ellk = ellk
f_ellstar = ellstar
f_kstar = kstar
```

## Backward stepwise selection

$\boxed{\text{Assumption: } \mathcal{S}_k \subset \mathcal{S}_{k+1}}$

Initialize: $\mathcal{S}_P = \mathcal{P}$

| | Removable features $(\mathcal{S}_{k+1})$ | Best to remove |
|---|---|---|
| $k = P-1$ | $\{\phi_1, \phi_2, \phi_3, \dots, \phi_{P-1}, \phi_P\}$ | $\phi_1$ |
| $k = 2$ | $\{\phi_2, \phi_3, \dots, \phi_{P-1}, \phi_P\}$ | $\phi_P$ |
| $k = 3$ | $\{\phi_1, \phi_2, \dots, \phi_P\}$ | $\phi_2$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $k = 0$ | $\{\phi_3\}$ | $\phi_3$ |

## **Pseudocode**: Backward stepwise selection

$\mathcal{S}_P = \mathcal{P}$

for $k = P-1 \dots 1$ :

  for $\kappa, \phi_p \in$ enumerate$(\mathcal{S}_{k+1})$

    $\mathcal{A}_\kappa = \mathcal{S}_{k+1} \setminus \phi_p$

    $\widehat{\theta}_\kappa =$ train$(\mathcal{A}_\kappa, \mathcal{D}_{\text{train}})$

    $\ell_\kappa =$ perf$(\mathcal{A}_\kappa, \widehat{\theta}_\kappa, \mathcal{D}_{\text{val}})$

  $\kappa^* =$ argbest$(\{\ell_\kappa\})$

  $\mathcal{S}_k = \mathcal{A}_{\kappa^*}$

$\mathcal{S}^* =$ best of $\{\mathcal{S}_k\}_P$

$\widehat{\theta}^* =$ train$(\mathcal{S}^*, \mathcal{D}_{\text{train}})$

$\ell^* =$ perf$(\mathcal{S}^*, \widehat{\theta}^*, \mathcal{D}_{\text{test}})$

## Forward stepwise selection

$\mathcal{S}_0 = \{\}$

for $k = 1 \ldots P$ :

for $\kappa, \phi_p \in$ enumerate$(\mathcal{P} \backslash \mathcal{S}_{k-1})$

$\mathcal{A}_\kappa = \mathcal{S}_{k-1} \cup \phi_p$

$\hat{\theta}_\kappa =$ train$(\mathcal{A}_\kappa, \mathcal{D}_{\text{train}})$

$\ell_\kappa =$ perf$(\mathcal{A}_\kappa, \hat{\theta}_\kappa, \mathcal{D}_{\text{val}})$

$\kappa^* =$ argbest$(\{\ell_\kappa\})$

$\mathcal{S}_k = \mathcal{A}_{\kappa^*}$

$\mathcal{S}^* =$ best of $\{\mathcal{S}_k\}_P$

$\hat{\theta}^* =$ train$(\mathcal{S}^*, \mathcal{D}_{\text{train}})$

$\ell^* =$ perf$(\mathcal{S}^*, \hat{\theta}^*, \mathcal{D}_{\text{test}})$

## Backward stepwise selection

$\mathcal{S}_P = \mathcal{P}$

for $k = P{-}1 \ldots 1$ :

for $\kappa, \phi_p \in$ enumerate$(\mathcal{S}_{k+1})$

$\mathcal{A}_\kappa = \mathcal{S}_{k+1} \setminus \phi_p$

$\hat{\theta}_\kappa =$ train$(\mathcal{A}_\kappa, \mathcal{D}_{\text{train}})$

$\ell_\kappa =$ perf$(\mathcal{A}_\kappa, \hat{\theta}_\kappa, \mathcal{D}_{\text{val}})$

$\kappa^* =$ argbest$(\{\ell_\kappa\})$

$\mathcal{S}_k = \mathcal{A}_{\kappa^*}$

$\mathcal{S}^* =$ best of $\{\mathcal{S}_k\}_P$

$\hat{\theta}^* =$ train$(\mathcal{S}^*, \mathcal{D}_{\text{train}})$

$\ell^* =$ perf$(\mathcal{S}^*, \hat{\theta}^*, \mathcal{D}_{\text{test}})$