

## 1 Maschine Learning

Algorithm learns class of tasks, measured by loss function, from experience.

**supervised learning** learn  $h : \Delta^* \rightarrow \Sigma^*$ ,  $h = t$ ; example:  $(x, y) \in \Delta^* \times \Sigma^*$ ,  $t(x) = y$ .

**unsupervised learning** learn  $h : \Delta^* \rightarrow \Sigma^*$ ,  $\ker(h) = \ker(t)$ ; example:  $x \in \Delta^*$ .

**reinforcement learning** learn strategy based on feedback from environment.

## 2 Supervised Learning

- model function  $t : \mathcal{M} \rightarrow \mathcal{R}$

-  $\text{supp}(t) = \{m \in \mathcal{M} \mid t(m) \neq 0\}$

-  $\bar{m} \in \text{supp}(t) \Leftrightarrow t(\bar{m}) = 1$

**Hypothesis** of A: potential result of A

**Hypothesis space**  $\mathcal{H}_A$  of A: set of all hypotheses

**h fits D** if  $h(x_i) = y_i$  for all  $(x_i, y_i) \in D$

**Version space**  $\mathcal{V}_A(D)$  of A: all hypotheses that fit D

**Inductive bias** of A: set of assumptions that A uses to predict outputs of unseen data

### 2.1 Conjunctive Clause

$\theta = (\theta_1, \dots, \theta_k), \theta_i \in M_i \cup \{\star, \perp\}$

-  $\theta_\perp = (\perp, \dots, \perp)$  most specific

-  $\theta_\star = (\star, \dots, \star)$  most general

-  $\text{supp}(h_{\theta_\perp}) = \emptyset, \text{supp}(h_{\theta_\star}) = \mathcal{M}$

-  $h_{\theta_\perp} = h_{(\theta_1, \dots, \perp, \dots, \theta_k)}$ ...

induced hypothesis  $h_\theta(m_1, \dots, m_k) = 1$  if  $\forall i : \theta_i \in \{m_i, \star\}$  else 0

$h \preceq h'$  if  $\text{supp}(h) \subseteq \text{supp}(h')$ .  $h$  is more specific (less general) than  $h'$

**Find-S Algorithm** finds most specific conjunctive clause that fits D

1. Start with  $\theta_\perp = (\perp, \dots, \perp)$

2. iterate over POSITIVE examples

3. min-generalize  $\theta$  to fit example

4.  $\perp \rightarrow a, a \rightarrow \star$

- maximal general hypothesis:

1. start at  $\theta_\star = (\star, \dots, \star)$

2. exclude every negative example

3.  $(\star, \dots) \rightarrow \{(b, \dots), (c, \dots)\}$

- If  $\mathcal{V}_A(D) \neq \emptyset$ , Find-S finds  $h \in \mathcal{V}_A(D)$

**disjunctive normal form**  $\Theta = \{\theta_1, \dots, \theta_m\}$

'finite set of conjunctive clauses'

induced hypothesis  $h_\Theta(\bar{m}) = 1$  if  $\exists \theta \in \Theta : h_\theta(\bar{m}) = 1$  else 0

-  $\text{supp}(\Theta) = \bigcup_{\theta \in \Theta} \text{supp}(\theta_\theta)$

- can represent all boolean functions

### Boundary sets of version space

maximally general hypotheses  $\mathcal{V}_A^\top(D) = \{h \in \mathcal{V}_A(D) \mid \nexists h' \in \mathcal{V}_A(D) : h \prec h'\}$

maximally specific hypotheses  $\mathcal{V}_A^\perp(D) = \{h \in \mathcal{V}_A(D) \mid \nexists h' \in \mathcal{V}_A(D) : h' \preceq h\}$

-  $h \in \mathcal{V}_A^\top$  maximal, weil:  $\forall x \in M \setminus \text{supp}(h) : \text{supp}(h) \cup \{x\} \notin \text{supp}(\mathcal{V}_A(D))$

Theorem:  $\mathcal{V}_A(D) = \{h \in \mathcal{H}_A \mid \exists h_\top \in \mathcal{V}_A^\top(D), \exists h_\perp \in \mathcal{V}_A^\perp(D) : h_\perp \preceq h \preceq h_\top\}$

$\rightarrow \mathcal{V}_A(D)$  det. by  $\mathcal{V}_A^\top(D)$  and  $\mathcal{V}_A^\perp(D)$

- only 1 lower bound (in  $\mathcal{V}_A^\perp(D)$ ), potentially multiple upper bounds (in  $\mathcal{V}_A^\top(D)$ )

### Candidate Elimination Algorithm

Output: DNF for  $\mathcal{V}_A^\top(D)$  and  $\mathcal{V}_A^\perp(D)$

1.  $S_\perp = \{\theta_\perp\}, S_\top = \{\theta_\star\}$

2. for  $1 \leq i \leq n : y_i = 1$  (pos. xmpls)

1. keep only fitting h from  $S_\top$

2.  $\forall \theta \in S_\perp : h_\theta(x_i) = 0$

- remove  $\theta$ , add all min generalizations  $\theta'$  of  $\theta$  that fit  $x_i$  to  $S_\perp$

3. keep only most specific h in  $S_\perp$

3. for  $1 \leq i \leq n : y_i = 0$  (neg. xmpls)

1. keep only fitting h from  $S_\perp$

2.  $\forall \theta \in S_\top : h_\theta(x_i) = 1$

- remove  $\theta$ , add all min specializations  $\theta'$  of  $\theta$  that fit  $x_i$  to  $S_\top$ , for which a more specific  $\theta_\perp \in S_\perp$  exists!

3. keep only most general h in  $S_\top$

-  $\mathcal{V}_A^\top = \{h_\theta \mid \theta \in S_\top\}, \mathcal{V}_A^\perp = \{h_\theta \mid \theta \in S_\perp\}$

- Concept identified if:  $S_\perp = S_\top$  and  $|S_\top| = 1$ .  $\mathcal{V}_A(D) = \emptyset$  if  $S_\perp = \emptyset \vee S_\top = \emptyset$

### 2.2 Decision Trees

**Splitting**  $\Pi = \{M_1, \dots, M_p\}$  is finite partition of (sub)feature Space  $\mathcal{M}'$

- induces splitting of  $\{1, \dots, n\}$  into  $I_{D'}(M_1), \dots, I_{D'}(M_p)$  (sets of indices)

- monotropic splits: based on 1 feature

- simple split: monotropic, into all realizations  $M = \{\bar{m} \in M \mid m_1 = a, \dots, b\}$

- binary split: monotropic, into 2 sets

$M = \{\bar{m} \in M \mid m_1 \in A\} \cup \{\bar{m} \in M \mid m_1 \notin A\}$

- induced hypothesis  $h_T(\bar{m}) = T(v)$ , where  $v$  is unique leaf s.t.  $\bar{m} \in M_v$

- simple decision trees can represent all hypotheses

### Decision Tree Quality Measures

- Number of leaves
- Height (max number of constraints to check)
- External path length (sum of all path lengths from root to leaf)

- Weighted external path length (sum of all path lengths from root to leaf, weighted by number of examples classified in that leaf)

Theorem: Given D and bound b, its NP hard to decide existence of decision tree T s.t.  $h_T$  fits D and T has ext. p.l.  $\leq b$

**Majority Class**  $\text{Maj}_D(M')$  maj.  $r \in R$

**Number of misclassifications**:  $\text{Err}_D(M', r)$  in feature subspace  $M'$  with majority class  $r$

-  $\text{Err}_D(T)$ : sum up all  $\text{Err}_D(M_v, T(v))$

**pure node** v if  $\text{Err}_D(M_v, T(v)) = 0$

- class distribution  $p_D^{M'}(r)$ :  $p(r)$  in  $M_v$

### impurity function

**loss functions (and derivatives)**

- $l(h, D) = \sum_{i=1}^n (1 - \delta_{y_i, h(x_i)})$
- $\delta_{ij} = 1$  if  $i = j$ , 0 otherwise.
- asd