Formule

- Cardinalità spazio delle ipotesi:

$$|X| = \prod |A_i|$$

- Cardinalità spazio dei concetti:

$$|C| = |\mathcal{P}(X)| = 2^{|X|}$$

- Cardinalità spazio delle ipotesi, semanticamente:

$$|H|_{sem} = 1 + \prod_{A} (|A_i| + 1)$$

- Cardinalità spazio delle ipotesi, sintatticamente:

$$|H|_{sint} = \prod (|A_i| + 2)$$

- Aspettativa di $G(X)$ su P ($Val(X)$ = range di valori di X):

$$E_P[g(X)] = \sum_{x \in Val(X)} g(x) \cdot P_X(x)$$

- Entropia di una variabile X:

$$H[X] = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i = E_P[\log_2(p)]$$

- Entropia di una distribuzione condizionale, con target T:

$$H[T|X = x_i] = -\sum_{j=1}^{m} P_{T|X}(t_j|x_i) \cdot \log_2 P_{T|X}(t_j|x_i)$$

- Entropia condizionale, con target T:

$$H[T|X] = \sum P(x) \cdot H(T|X = x)$$

- Information Gain su variabile X e target T:

$$IG[T|X] = H[T] - H[T|X]$$

Definizioni

- **learner**, la parte di programma che impara dagli esempi in modo automatico
- **trainer**, il *dataset* che fornisce esperienza al *learner*
- **esperienza diretta** dove il learner può acquisire informazione utile direttamente dagli esempi o dover inferire indirettamente da essi l'informazione necessaria (può essere chiaramente più complicato). Altrimenti è indiretta
- **apprendimento supervisionato**, dove vengono forniti a priori esempi di comportamento e si suppone che il *trainer* dia la risposta corretta per ogni input (mentre il learner usa gli esempi forniti per apprendere). L'esperienza è fornita da un insieme di coppie:

  $$S \equiv \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

  e, per ogni input ipotetico x_i l'ipotetico trainer restituisce il corretto y_i
- **apprendimento non supervisionato**, dove si riconosce *schemi* nell'input senza indicazioni sui valori in uscita. Non c'è target e si ha *libertà di classificazione*. Si cerca una *regolarità* e una *struttura* insita nei dati. In questo caso si ha:

  $$S \equiv \{x_1, x_2, \ldots, x_n\}$$

  Il clustering è un tipico problema di apprendimento non supervisionato. Non si ha spesso un metodo oggettivo per stabilire le prestazioni che vengono quindi valutate da umani
- **apprendimento per rinforzo**, dove bisogna apprendere, tramite il *learner* sulla base della risposta dell'ambiente alle proprie azioni. Si lavora con un*addestramento continuo*, aggiornando le ipotesi con l'arrivo dei dati (ad esempio per una macchina che deve giocare ad un gioco). Durante la fase di test bisogna conoscere le prestazioni e valutare la correttezza di quanto appreso. Il learner viene addestrato tramite *rewards* e quindi apprende una strategia per massimizzare i *rewards*, detta **strategia di comportamento** e per valutare la prestazione si cerca di massimizzare "a lungo termine" la ricompensa complessivamente ottenuta

- **apprendimento attivo**, dove il *learner* può "domandare" sui dati disponibili
- **apprendimento passivo**, dove il *learner* apprende solo a partire dai dati disponibili
- X, **spazio delle istanze**, ovvero la collezione di tutte le possibili **istanze** utili per qualche compito di *learning*. In termini statistici lo *spazio delle istanze* non è altro che lo **spazio campione** (ovvero lo spazio degli esiti fondamentali di un esperimento concettuale)
- $x \in X$, **istanza**, ovvero un singolo "oggetto" preso dallo **spazio delle istanze**. Ogni **istanza** è rappresentata tramite un **vettore di attributi unici** (un attributo per posizione del vettore)
- c, **concetto**, $c \subseteq X$, ovvero un sottoinsieme dello *spazio delle istanze* che descrive una *classe* di oggetti (ovvero di istanze) alla quale siamo interessati per costruire un modello di *machine learning*. In pratica raccolgo quel sottoinsieme di istanze che mi garantiscono, per esempio, uno o più attributi. La nozione statistica equivalente è quella di *evento* (ovvero un sottoinsieme dello *spazio campione*). Si ha quindi che, preso un concetto $A \subseteq X$:

$$f_A : X \to \{0, 1\}$$

$$f_a(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{altrimenti} \end{cases}$$

- h, **ipotesi**, $h \subseteq X$
- H, **spazio delle ipotesi**
- $(x, f(x))$, **esempio**, ovvero prendo un'istanza e la vado ad etichettare con la sua classe di appartenenza. La funzione f è detta **funzione target**
- $D = \{(x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$, **training set**, ovvero è la raccolta degli esempi. Qualora si avesse a che fare con un *training non supervisionato* si avrebbe: $D = \{x_1, \ldots, x_n\}$
- $\{(x'_1, f(x'_1)), \ldots, (x'_n, f(x'_n))\}$, **test**

- un **modello di machine learning** (dove *machine learning* viene anche definito come lo studio di diverse strategie, più precisamente di ottimizzazione, per cercare ipotesi soddisfacenti/efficienti nello spazio delle ipotesi) è quindi l'*ipotesi migliore*. Questo **modello predittivo** viene addestrato tramite il *training set* e servirà per inferire nuove informazioni mai state osservate nel *training set*. Lo *spazio delle ipotesi* può quindi essere chiamato anche **spazio dei modelli** (come del resto *ipotesi* e **modello** intendono la stessa cosa)

- **linguaggio delle ipotesi**, è il linguaggio che definisce lo *spazio delle ipotesi/modelli*

- **cross validation**, ovvero ripeto m volte la validazione su campioni diversi di input per evitare che un certo risultato derivi dalla fortuna

- **ipotesi H**, ovvero una congiunzione $\wedge$ di vincoli sugli attributi. Tale ipotesi è **consistente**, ovvero è coerente con tutti gli esempi

- **soddisfazione di un'ipotesi**: un'istanza x soddisfa un'ipotesi h sse tutti i vincoli espressi da h sono soddisfatti dai valori di x e si indica con:
$$h(x) = 1$$

- **concept learning** è la ricerca, nello spazio delle ipotesi, di funzioni che assumano valori all'interno di $\{0, 1\}$. In altre parole si parla di funzioni che hanno come dominio lo **spazio delle ipotesi** e come codominio $\{0, 1\}$:
$$f : X \to \{0, 1\}$$
Volendo si possono usare insiemi e non funzioni.
Si cerca quindi con opportune procedure la miglior ipotesi che si adatta meglio al concetto implicato dal *training set*. Valori del concept learning:
  -
  - specificato
  - non importante, che si indica con "?", e che può assumere qualsiasi valore. Avere un'ipotesi con tutti i valori del vettore pari a "?" implica avere l'ipotesi più generale, avendo classificato tutte le istanze solo come esempi positivi

  - nullo e si indica con $\emptyset$. Avere un'ipotesi con tutti i valori del vettore pari a $\emptyset$ implica avere l'ipotesi più specifica, avendo classificato tutte le istanze solo come esempi negativi

- **inductive learning** quando voglio apprendere una funzione da un esempio (banalmente una funzione target f con esempio $(x, f(x))$, ovvero una coppia). Si cerca quindi un'ipotesi h, a partire da un insieme d'esempi di apprendimento, tale per cui $h \approx f$

- **soddisfacibilità** quando un esempio x soddisfa un'ipotesi h, evento indicato con:
$$h(x) = 1$$
a priori sul fatto che x sia un esempio positivo o negativo del *target concept*. Si ha quindi che i valori x soddisfano i vincoli h

- Si dice che h è **consistente** con il training set D di concetti target sse:
$$Consistent(h, D) := h(x) = c(x),\ \forall \langle x, c(x) \rangle \in D$$

- Si definisce **version space**, rispetto ad H e D, come il sottoinsieme delle ipotesi da H consistenti con D e si indica con:
$$VS_{H,D} = \{h \in H | \, Consistent(h, D)$$

- Date $h_j, h_k \in H$ booleane e definite su X. Si ha che h_j è **più generale o uguale a** h_k (e si scrive con $h_j \geq h_k$) sse:
$$(h_k(x) = 1) \longrightarrow (h_j(x) = 1),\ \forall x \in X$$
Si impone quindi un ordine parziale.
Si ha che h_j è **più generale di** h_k (e si scrive con $h_j > h_k$) sse:
$$(h_j \geq h_k) \wedge (h_k \ngeq h_j)$$
Riscrivendo dal punto di vista insiemistico si ha che h_j è **più generale o uguale a** h_k sse:
$$h_k \supseteq h_j$$
e che è **più generale di** h_k sse:
$$h_k \supset h_j$$
Dal punto di vista logico si ha che h_j è **più generale di** h_k sse impone meno vincoli di h_k

- **Find-S** permette di partire dall'ipotesi più specifica (attributi nulli, indicati con $\emptyset$) e generalizzarla, trovando ad ogni passo un'ipotesi più specifica e consistente con il training set D. L'ipotesi in uscita sarà anche consistente con gli esempi negativi dando prova che il target è effettivamente in H. Con questo algoritmo non si può dimostrare di aver trovato l'unica ipotesi consistente con gli esempi e, ignorando gli esempi negativi non posso capire se D contiene dati inconsistenti. Inoltre non ho l'ipotesi più generale

- Il **bias induttivo** (con **bias** che normalmente denota una *distorsione* o un *scostamento* dei dati) di L è un insieme minimale di asserzioni B tale che, per ogni concetto target c e D_c corrispondente si ha che:

  $$[B \wedge D_c \wedge x_i] \vdash L(x_i, D_c),\ \forall x_i \in X$$

  con $\vdash$ che rappresenta l'implicazione logica

  Possiamo quindi distinguere:

  - **sistema induttivo**, dove si hanno in input gli esempi di training e la nuova istanza, viene usato l'algoritmo *candidate eliminate* con H e si ottiene o la classificazione della nuova istanza nulla
  - **sistema deduttivo** equivalente al sistema induttivo sopra descritto dove in input si aggiunge l'asserzione "H contiene il concetto target" e si produce lo stesso output tramite un **prover di teoremi**

-

Procedimenti comodi

- Per **find-S** parto da tutti $\emptyset$, prendo solo esempi positivi e procedo sistemando attributo per attributo