Università Degli Studi Di Milano

Metodi di Ensemble Gerarchici per la Predizione Strutturata della Funzione delle Proteine

Relatore

Prof. Giorgio Valentini

Correlatore

Dr. Marco Notaro

Candidato

Marco Odore

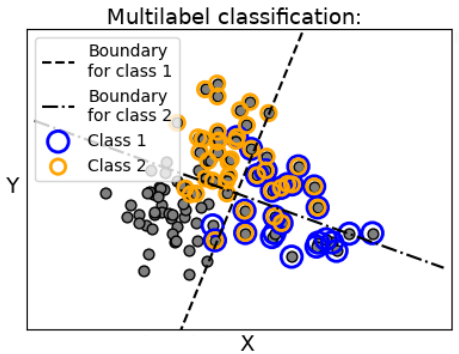
10 Luglio 2018

Il problema della predizione della funzione delle proteine

- Identificare la funzione delle proteine attraverso le analisi di laboratorio è **costosa** e richiede **molto tempo**

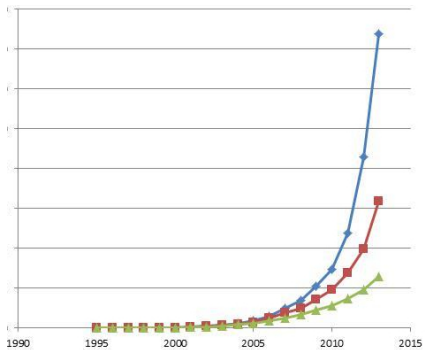
Il problema della predizione della funzione delle proteine

- Identificare la funzione delle proteine attraverso le analisi di laboratorio è **costosa** e richiede **molto tempo**
- Esistono migliaia di funzioni a cui poter associare un gene/proteina, anche contemporaneamente **(problema multiclasse e multietichetta)**



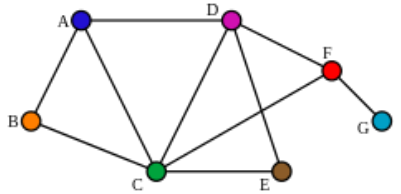
Il problema della predizione della funzione delle proteine

- Identificare la funzione delle proteine attraverso le analisi di laboratorio è **costosa** e richiede **molto tempo**
- Esistono migliaia di funzioni a cui poter associare un gene/proteina, anche contemporaneamente **(problema multiclasse e multietichetta)**
- Il quantitativo di dati genomici cresce molto rapidamente.



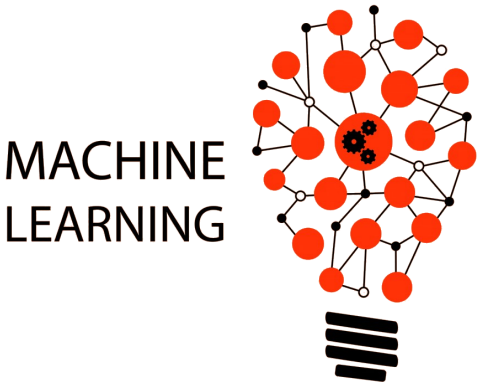
Il problema della predizione della funzione delle proteine

- Identificare la funzione delle proteine attraverso le analisi di laboratorio è **costosa** e richiede **molto tempo**
- Esistono migliaia di funzioni a cui poter associare un gene/proteina, anche contemporaneamente **(problema multiclasse e multietichetta)**
- Il quantitativo di dati genomici cresce molto rapidamente.
- Le classi che rappresentano le diverse funzioni delle proteine **non sono indipendenti**.



Il problema della predizione della funzione delle proteine

- Identificare la funzione delle proteine attraverso le analisi di laboratorio è **costosa** e richiede **molto tempo**
- Esistono migliaia di funzioni a cui poter associare un gene/proteina, anche contemporaneamente **(problema multiclasse e multietichetta)**
- Il quantitativo di dati genomici cresce molto rapidamente.
- Le classi che rappresentano le diverse funzioni delle proteine **non sono indipendenti**.
- La **classificazione manuale** delle proteine è quindi infattibile. È necessario quindi un approccio **automatico**.



Tassonomie per le funzioni delle proteine

- Esistono due tassonomie principali per l'organizzazione delle funzioni:

Tassonomie per le funzioni delle proteine

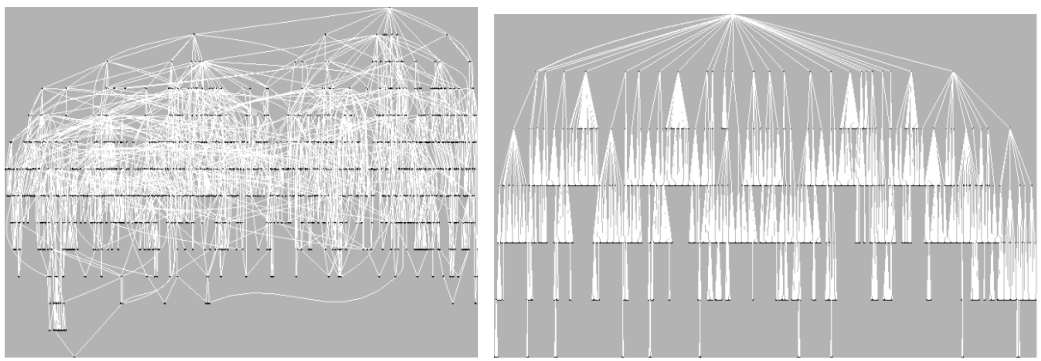
- Esistono due tassonomie principali per l'organizzazione delle funzioni:
  - **Gene Ontology** (GO): organizza le funzioni come un grafo diretto aciclico (DAG), ed è articolata in tre ontologie differenti: *Biological Process* (BP), *Molecular Function* (MF) e *Cellular Component* (CC).

Tassonomie per le funzioni delle proteine

- Esistono due tassonomie principali per l'organizzazione delle funzioni:
  - **Gene Ontology** (GO): organizza le funzioni come un grafo diretto aciclico (DAG), ed è articolata in tre ontologie differenti: *Biological Process* (BP), *Molecular Function* (MF) e *Cellular Component* (CC).
  - **Functional Catalogue** (FunCat): è organizzato invece come un albero, e descrive le funzioni in maniera più sintetica rispetto alla Gene Ontology.

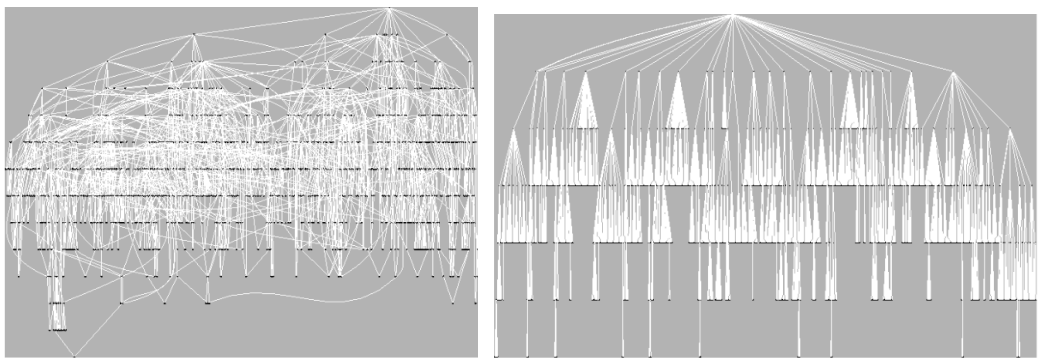
Tassonomie per le funzioni delle proteine

- Esistono due tassonomie principali per l'organizzazione delle funzioni:
  - **Gene Ontology** (GO): organizza le funzioni come un grafo diretto aciclico (DAG), ed è articolata in tre ontologie differenti: *Biological Process* (BP), *Molecular Function* (MF) e *Cellular Component* (CC).
  - **Functional Catalogue** (FunCat): è organizzato invece come un albero, e descrive le funzioni in maniera più sintetica rispetto alla Gene Ontology.



Tassonomie per le funzioni delle proteine

- Esistono due tassonomie principali per l'organizzazione delle funzioni:
  - **Gene Ontology** (GO): organizza le funzioni come un grafo diretto aciclico (DAG), ed è articolata in tre ontologie differenti: *Biological Process* (BP), *Molecular Function* (MF) e *Cellular Component* (CC).
  - **Functional Catalogue** (FunCat): è organizzato invece come un albero, e descrive le funzioni in maniera più sintetica rispetto alla Gene Ontology.



- Data la granularità e specificità superiori della GO e il suo largo utilizzo nella comunità scientifica, nella tesi si è utilizzata tale ontologia.

La predizione della funzione delle proteine tramite metodi automatici

Schematicamente i metodi per effettuare predizioni della funzione delle proteine in maniera automatica si possono classificare in [*Valentini, 2014*]:

La predizione della funzione delle proteine tramite metodi automatici

Schematicamente i metodi per effettuare predizioni della funzione delle proteine in maniera automatica si possono classificare in [*Valentini, 2014*]:

- Metodi basati sulla **comparazione di biosequenze**: si basano sull'idea che sequenze simili condividano funzioni simili.

La predizione della funzione delle proteine tramite metodi automatici

Schematicamente i metodi per effettuare predizioni della funzione delle proteine in maniera automatica si possono classificare in [*Valentini, 2014*]:

- Metodi basati sulla **comparazione di biosequenze**: si basano sull'idea che sequenze simili condividano funzioni simili.
- Metodi **basati su reti**: sono metodi applicati a dati rappresentati sotto forma di reti, che si basano sugli algoritmi di propagazione delle etichette.

La predizione della funzione delle proteine tramite metodi automatici

Schematicamente i metodi per effettuare predizioni della funzione delle proteine in maniera automatica si possono classificare in [*Valentini, 2014*]:

- Metodi basati sulla **comparazione di biosequenze**: si basano sull'idea che sequenze simili condividano funzioni simili.
- Metodi **basati su reti**: sono metodi applicati a dati rappresentati sotto forma di reti, che si basano sugli algoritmi di propagazione delle etichette.
- Metodi **flat supervisionati** (es. basati su metodi di apprendimento supervisionato).

La predizione della funzione delle proteine tramite metodi automatici

Schematicamente i metodi per effettuare predizioni della funzione delle proteine in maniera automatica si possono classificare in [*Valentini, 2014*]:

- Metodi basati sulla **comparazione di biosequenze**: si basano sull'idea che sequenze simili condividano funzioni simili.
- Metodi **basati su reti**: sono metodi applicati a dati rappresentati sotto forma di reti, che si basano sugli algoritmi di propagazione delle etichette.
- Metodi **flat supervisionati** (es. basati su metodi di apprendimento supervisionato).
- Metodi **Kernel per spazi di output strutturato**: sono metodi che sfruttano funzioni kernel congiunte per predire in spazi di output strutturato.

La predizione della funzione delle proteine tramite metodi automatici

Schematicamente i metodi per effettuare predizioni della funzione delle proteine in maniera automatica si possono classificare in [*Valentini, 2014*]:

- Metodi basati sulla **comparazione di biosequenze**: si basano sull'idea che sequenze simili condividano funzioni simili.
- Metodi **basati su reti**: sono metodi applicati a dati rappresentati sotto forma di reti, che si basano sugli algoritmi di propagazione delle etichette.
- Metodi **flat supervisionati** (es. basati su metodi di apprendimento supervisionato).
- Metodi **Kernel per spazi di output strutturato**: sono metodi che sfruttano funzioni kernel congiunte per predire in spazi di output strutturato.
- Metodi **Ensemble Gerarchici**: i metodi trattati in questa tesi.

Metodi Ensemble Gerarchici 1/2

I Metodi di Ensemble Gerarchici sono metodi caratterizzati da due step principali [*Valentini, 2014*]:

Metodi Ensemble Gerarchici 1/2

I Metodi di Ensemble Gerarchici sono metodi caratterizzati da due step principali [*Valentini, 2014*]:

1. **Predizione flat** delle diverse classi dell'ontologia, generando diversi predittori *indipendenti*.

Metodi Ensemble Gerarchici 1/2

I Metodi di Ensemble Gerarchici sono metodi caratterizzati da due step principali [*Valentini, 2014*]:

1. **Predizione flat** delle diverse classi dell'ontologia, generando diversi predittori *indipendenti*.
2. **Combinazione e correzione gerarchica delle predizioni** sfruttando il DAG dei termini della GO.

Metodi Ensemble Gerarchici 1/2

I Metodi di Ensemble Gerarchici sono metodi caratterizzati da due step principali [*Valentini, 2014*]:

1. **Predizione flat** delle diverse classi dell'ontologia, generando diversi predittori *indipendenti*.
2. **Combinazione e correzione gerarchica delle predizioni** sfruttando il DAG dei termini della GO.

Il secondo step rappresenta la componente *ensemble* del metodo. Tale step si rende necessario in quanto le predizioni flat non tengono in considerazione la struttura gerarchica dei DAG della GO, portando a risultati *inconsistenti*.

Metodi Ensemble Gerarchici 1/2

I Metodi di Ensemble Gerarchici sono metodi caratterizzati da due step principali [*Valentini, 2014*]:

1. **Predizione flat** delle diverse classi dell'ontologia, generando diversi predittori *indipendenti*.
2. **Combinazione e correzione gerarchica delle predizioni** sfruttando il DAG dei termini della GO.

Il secondo step rappresenta la componente *ensemble* del metodo. Tale step si rende necessario in quanto le predizioni flat non tengono in considerazione la struttura gerarchica dei DAG della GO, portando a risultati *inconsistenti*.

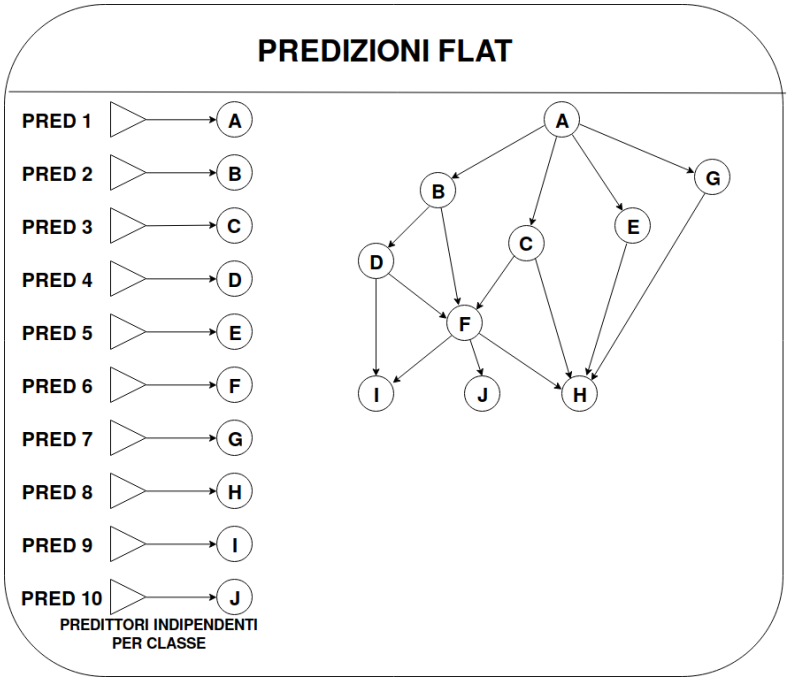
Consistenza & True Path Rule

Un insieme di predizioni $\hat{y} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|N|} >$, dove $|N|$ è la cardinalità dei termini della gerarchia, è definito *consistente*, se rispetta la *True Path Rule*, e cioè:

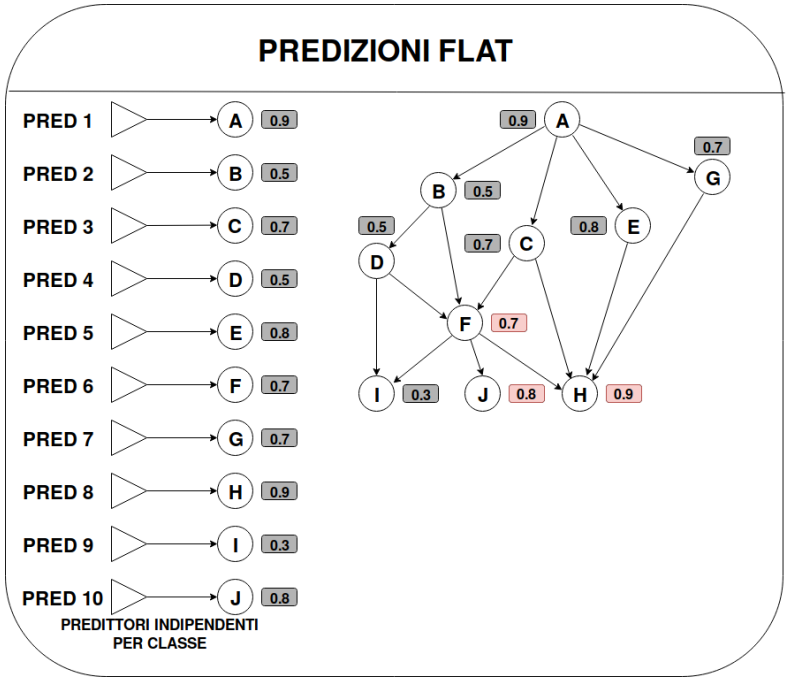
$$y \quad consistente \quad \leftrightarrow \forall i \in N, j \in par(i) \rightarrow y_j \geq y_i$$

Dove $par(i)$ indica l'insieme dei termini genitori del nodo i nella gerarchi

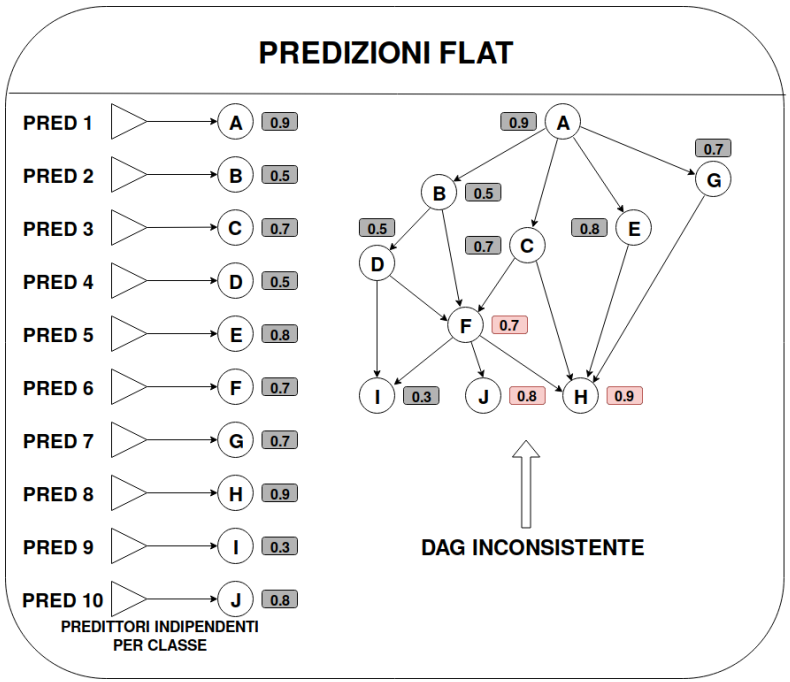
Metodi Ensemble Gerarchici (Esempio) 2/2



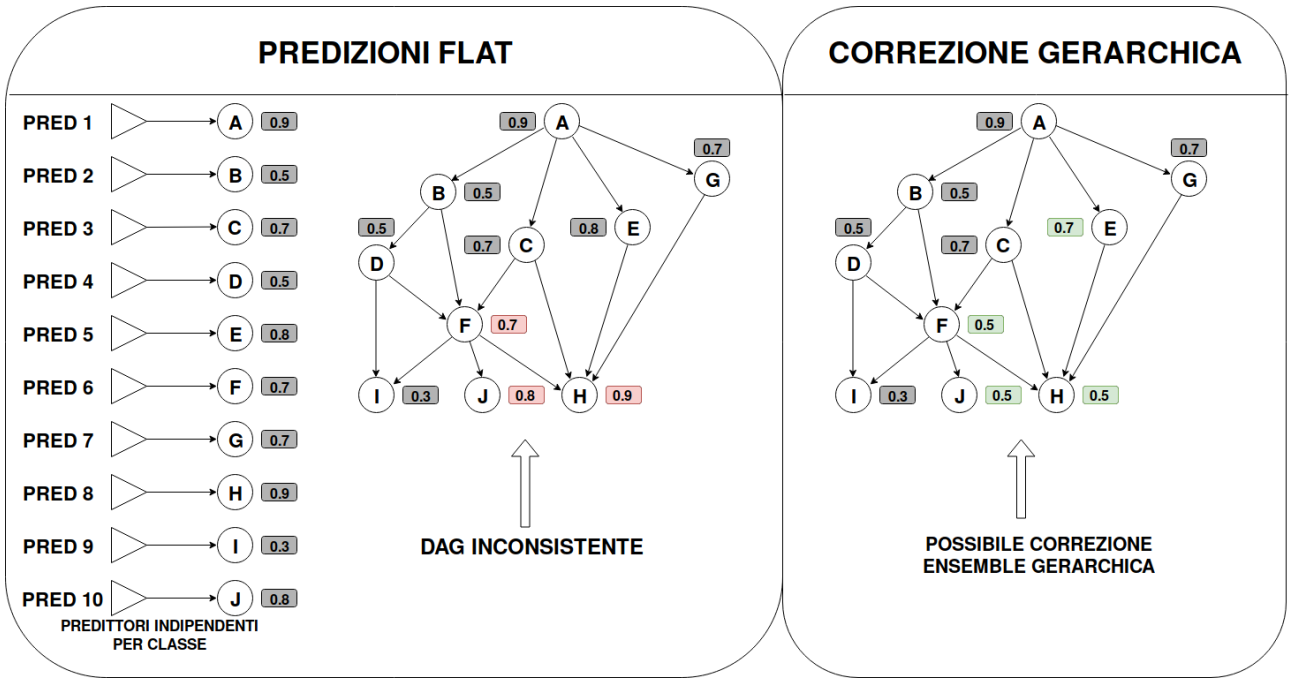
Metodi Ensemble Gerarchici (Esempio) 2/2



Metodi Ensemble Gerarchici (Esempio) 2/2



Metodi Ensemble Gerarchici (Esempio) 2/2



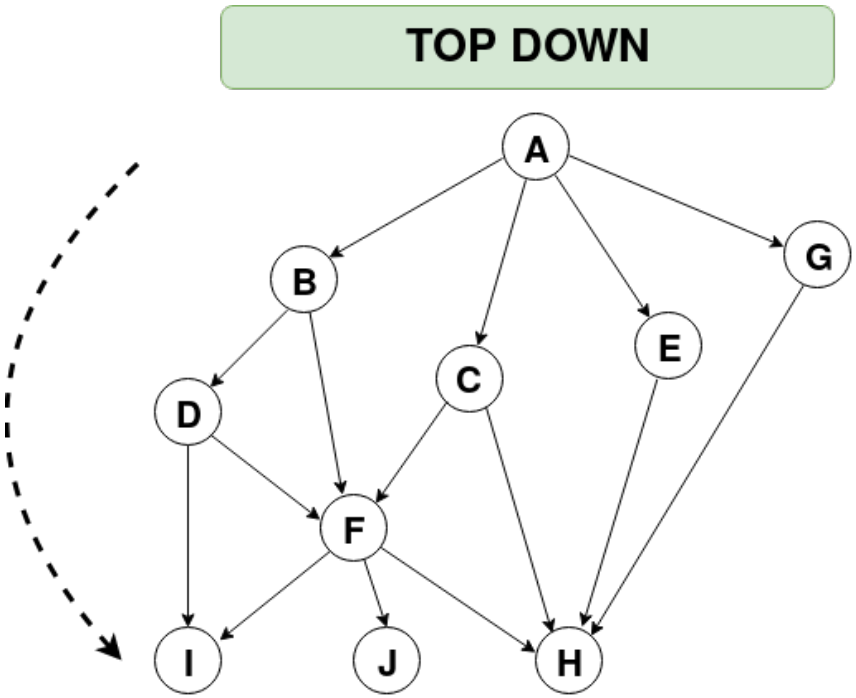
Metodi Ensemble Gerarchici: Approcci

Esistono fondamentalmente due approcci generali per la correzione [*Valentini, 2011*]:

Metodi Ensemble Gerarchici: Approcci

Esistono fondamentalmente due approcci generali per la correzione [*Valentini, 2011*]:

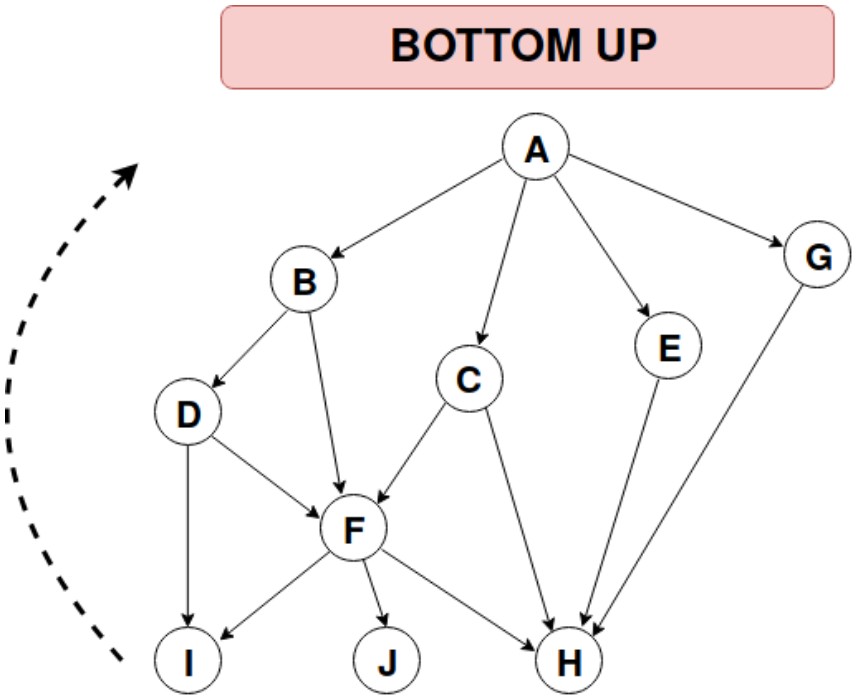
- *Top-down*: le predizioni vengono corrette dai nodi più generali a quelli più specifici.



Metodi Ensemble Gerarchici: Approcci

Esistono fondamentalmente due approcci generali per la correzione [*Valentini, 2011*]:

- *Top-down*: le predizioni vengono corrette dai nodi più generali a quelli più specifici.
- *Bottom-up*: Le predizioni vengono corrette dai nodi più specifici verso quelli più generali.



Metodo Top-Down Gerarchico (HTD-DAG)

- È un metodo che utilizza l'approccio *Top-Down* [*Notaro et al., 2017*].

[1] **Dove il livello è quello del cammino massimo dalla radice**

Metodo Top-Down Gerarchico (HTD-DAG)

- È un metodo che utilizza l'approccio *Top-Down* [*Notaro et al., 2017*].
- La correzione avviene ricorsivamente, percorrendo il grafo per *livelli*[1]. Più precisamente, dato il grafo $G = (N, E)$, gli score flat $f(x) = \hat{y}$ sono corretti gerarchicamente a $\bar{y}$, applicando la seguente regola:

Aggiornamento con HTD-DAG

$$\bar{y}_i := \begin{cases} \hat{y}_i & \text{if } \ i \in root(G) \\ min_{j \in par(i)} \bar{y}_j & \text{if } \ min_{j \in par(i)} \bar{y}_j < \hat{y}_i \\ \hat{y}_i & \text{altrimenti} \end{cases}$$

Dove $par(i)$ specifica i genitori del nodo i.

[1] **Dove il livello è quello del cammino massimo dalla radice**

Metodo True Path Rule per DAG (TPR-DAG)

- È un metodo che combina gli approcci top-down e bottom-up per la correzione delle predizioni flat ([*Valentini, 2011*], [*Cesa Bianchi et al., 2012*] per ontologie ad albero, [*Notaro et al., 2017*] per DAG).

Metodo True Path Rule per DAG (TPR-DAG)

- È un metodo che combina gli approcci top-down e bottom-up per la correzione delle predizioni flat ([*Valentini, 2011*], [*Cesa Bianchi et al., 2012*] per ontologie ad albero, [*Notaro et al., 2017*] per DAG).
- È suddiviso in due step sequenziali:

Metodo True Path Rule per DAG (TPR-DAG)

- È un metodo che combina gli approcci top-down e bottom-up per la correzione delle predizioni flat ([*Valentini, 2011*], [*Cesa Bianchi et al., 2012*] per ontologie ad albero, [*Notaro et al., 2017*] per DAG).
- È suddiviso in due step sequenziali:
  1. **Step bottom-up**: che partendo dai nodi più specifici del DAG, propaga quelle predizioni flat che sono considerate *positive*.

Metodo True Path Rule per DAG (TPR-DAG)

- È un metodo che combina gli approcci top-down e bottom-up per la correzione delle predizioni flat ([*Valentini, 2011*], [*Cesa Bianchi et al., 2012*] per ontologie ad albero, [*Notaro et al., 2017*] per DAG).
- È suddiviso in due step sequenziali:
  1. **Step bottom-up**: che partendo dai nodi più specifici del DAG, propaga quelle predizioni flat che sono considerate *positive*.
  2. **Step top-down**: È il medesimo step utilizzato dal metodo HTD-DAG.

Metodo True Path Rule per DAG (TPR-DAG)

- È un metodo che combina gli approcci top-down e bottom-up per la correzione delle predizioni flat ([*Valentini, 2011*], [*Cesa Bianchi et al., 2012*] per ontologie ad albero, [*Notaro et al., 2017*] per DAG).
- È suddiviso in due step sequenziali:
  1. **Step bottom-up**: che partendo dai nodi più specifici del DAG, propaga quelle predizioni flat che sono considerate *positive*.
  2. **Step top-down**: È il medesimo step utilizzato dal metodo HTD-DAG.
- Lo step top down si rende necessario in quanto la propagazione delle predizioni positive dal basso verso l'alto non garantisce la consistenza delle predizioni necessarie alla True Path Rule.

Metodo True Path Rule per DAG (TPR-DAG)

- È un metodo che combina gli approcci top-down e bottom-up per la correzione delle predizioni flat ([*Valentini, 2011*], [*Cesa Bianchi et al., 2012*] per ontologie ad albero, [*Notaro et al., 2017*] per DAG).
- È suddiviso in due step sequenziali:
  1. **Step bottom-up**: che partendo dai nodi più specifici del DAG, propaga quelle predizioni flat che sono considerate *positive*.
  2. **Step top-down**: È il medesimo step utilizzato dal metodo HTD-DAG.
- Lo step top down si rende necessario in quanto la propagazione delle predizioni positive dal basso verso l'alto non garantisce la consistenza delle predizioni necessarie alla True Path Rule.
- La selezione dei nodi considerati *positivi* può avvenire in diverse maniere: con *soglia adattiva*, *senza soglia* e con *soglia fissa*.

Generalized Pool Adjacent Violator (GPAV) (1/5)

- Per il passo Top-Down dell'algoritmo TPR-DAG (e HTD) è stato sviluppato un nuovo metodo all'interno di questa tesi, e cioè **Generalized Pool Adjacent Violator** (GPAV) [*Burdakov et al., 2006*], un algoritmo che permette di risolvere i problemi di **Isotonic Regression**, definiti come:

Isotonic Regression (caso generale con ordinamento parziale)

Dato un DAG, $G(N, E)$, con il set di nodi $N = \{1, 2, ..., n\}$, si deve trovare il vettore $x^* \in R^n$ tale che:

$$min \sum_{i=1}^{n} w_i (x_i - a_i)^2$$

$$\text{such that } x_i \leq x_j \ \forall (i, j) \in E$$

Generalized Pool Adjacent Violator (GPAV) (1/5)

- Per il passo Top-Down dell'algoritmo TPR-DAG (e HTD) è stato sviluppato un nuovo metodo all'interno di questa tesi, e cioè **Generalized Pool Adjacent Violator** (GPAV) [*Burdakov et al., 2006*], un algoritmo che permette di risolvere i problemi di **Isotonic Regression**, definiti come:

Isotonic Regression (caso generale con ordinamento parziale)

Dato un DAG, $G(N, E)$, con il set di nodi $N = \{1, 2, ..., n\}$, si deve trovare il vettore $x^* \in R^n$ tale che:

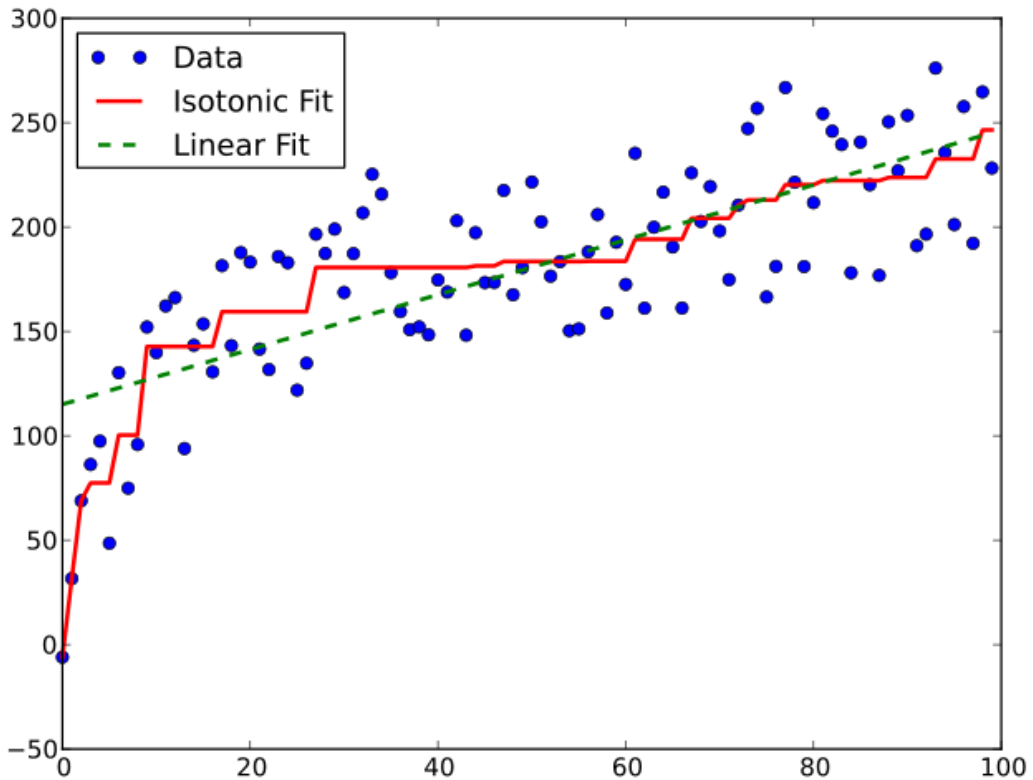
$$min \sum_{i=1}^{n} w_i (x_i - a_i)^2$$

$$\text{such that } x_i \leq x_j \ \forall (i, j) \in E$$

- Con una complessità pari a $O(n^2)$.

Generalized Pool Adjacent Violator (GPAV) (2/5)

Esempio di Isotonic Regression con *ordinamento totale*:



Generalized Pool Adjacent Violator (GPAV) (3/5)

- L'algoritmo richiede un *ordinamento topologico* del grafo.

Generalized Pool Adjacent Violator (GPAV) (3/5)

- L'algoritmo richiede un *ordinamento topologico* del grafo.
- L'algoritmo genera uno split del set di nodi N del DAG, in un insieme di *blocchi disgiunti* (inizialmente il numero di blocchi è uguale a $|N|$).

Generalized Pool Adjacent Violator (GPAV) (3/5)

- L'algoritmo richiede un *ordinamento topologico* del grafo.
- L'algoritmo genera uno split del set di nodi N del DAG, in un insieme di *blocchi disgiunti* (inizialmente il numero di blocchi è uguale a $|N|$).
- Seguendo l'ordinamento topologico del grafo, un blocco *assorbe* un suo blocco *predecessore* se si verificano determinate condizioni.

Generalized Pool Adjacent Violator (GPAV) (3/5)

- L'algoritmo richiede un *ordinamento topologico* del grafo.
- L'algoritmo genera uno split del set di nodi N del DAG, in un insieme di *blocchi disgiunti* (inizialmente il numero di blocchi è uguale a $|N|$).
- Seguendo l'ordinamento topologico del grafo, un blocco *assorbe* un suo blocco *predecessore* se si verificano determinate condizioni.
- I nodi presenti nel medesimo blocco B_i *condividono il medesimo valore* x_i, e quindi a seguito dell'assorbimento sarà necessario un aggiornamento di tale valore.

Generalized Pool Adjacent Violator (GPAV e ISO-TPR) (4/5)

Algorithm 1 GPAV

Input:
$G = (N, E)$
$N = \{1, 2, ..., |N|\}$
$\hat{y} =< \hat{y}_1, \hat{y}_2, ..., \hat{y}_{|N|} >$
procedure GPAV
    for (each $i \in N$) **do**
        $B_i = \{i\}$
        $B_i^- = i^-$
        $x_i = \hat{y}_i$
        $W_i = w_i$
    for $k = 1, 2, ..., n$ **do**
        // finché esiste un predecessore di B_k che viola la monotonicità
        while $\{i \in B_k^- : x_i > x_k\} \neq 0$ **do**
            // Trova l'elemento che viola maggiormente il vincolo
            Find $j \in B_k^- : x_j = max\{x_i : i \in B_k^-\}$
            Absorb(k, j) *// j viene assorbito da B_k*
    //Aggiornamento per soluzione finale
    for each $k \in H$ **do**
        for each $i \in B_k$ **do**
            $\bar{y}_i = x_i$
Output:
$\bar{y} =< \bar{y}_1, \bar{y}_2, ..., \bar{y}_{|N|} >$

Generalized Pool Adjacent Violator (GPAV e ISO-TPR) (5/5)

- Riassumendo, l'algoritmo effettua degli assorbimenti di blocchi adiacenti, finché questi violano i vincoli del problema quadratico, generando di fatto una partizione dei nodi, in cui le parti condividono lo stesso valore.

Generalized Pool Adjacent Violator (GPAV e ISO-TPR) (5/5)

- Riassumendo, l'algoritmo effettua degli assorbimenti di blocchi adiacenti, finché questi violano i vincoli del problema quadratico, generando di fatto una partizione dei nodi, in cui le parti condividono lo stesso valore.
- Sostituendo GPAV allo step Top-Down dell'algoritmo TPR-DAG visto in precedenza (invece che HTD), si ottiene l'algoritmo **ISO-TPR**, un altro nuovo metodo utilizzato in questa tesi.

Generalized Pool Adjacent Violator (GPAV e ISO-TPR) (5/5)

- Riassumendo, l'algoritmo effettua degli assorbimenti di blocchi adiacenti, finché questi violano i vincoli del problema quadratico, generando di fatto una partizione dei nodi, in cui le parti condividono lo stesso valore.
- Sostituendo GPAV allo step Top-Down dell'algoritmo TPR-DAG visto in precedenza (invece che HTD), si ottiene l'algoritmo **ISO-TPR**, un altro nuovo metodo utilizzato in questa tesi.
- Il passo bottom-up di TPR permette di aumentare la sensibilità (Recall) e l'algoritmo GPAV nello step top-down assicura la consistenza, manetenendo gli score il più vicino possibile agli score ottenuti nel passo bottom-up (nel senso dei minimi quadrati).

Predizione della funzione delle proteine in C.elegans (WORM)

- Si è eseguita la sperimentazione sul genoma della specie *Caenorhabditis elegans* (WORM), utilizzando come insieme delle istanze e input del problema una matrice simmetrica generata dal network di interazione proteina-proteina *STRING* [*Szklarczyk et al., 2015*] [2].

[2] **Search Tool for the Retrieval of Interacting Genes/Proteins**

Predizione della funzione delle proteine in C.elegans (WORM)

- Si è eseguita la sperimentazione sul genoma della specie *Caenorhabditis elegans* (WORM), utilizzando come insieme delle istanze e input del problema una matrice simmetrica generata dal network di interazione proteina-proteina *STRING* [*Szklarczyk et al., 2015*] [2].
- Tale matrice STRING ha dimensione 15752×15752 (WORM). Il nostro problema ha quindi 15752 istanze.

[2] **Search Tool for the Retrieval of Interacting Genes/Proteins**

Predizione della funzione delle proteine in C.elegans (WORM)

- Si è eseguita la sperimentazione sul genoma della specie *Caenorhabditis elegans* (WORM), utilizzando come insieme delle istanze e input del problema una matrice simmetrica generata dal network di interazione proteina-proteina *STRING* [*Szklarczyk et al., 2015*] [2].
- Tale matrice STRING ha dimensione 15752×15752 (WORM). Il nostro problema ha quindi 15752 istanze.
- In base al tipo do ontologia, si hanno DAG con un quantitativo diverso di nodi(termini) e archi(relazioni):

ontologia	numero di termini	numero di archi
BP	4068	8066
MF	1163	1567
CC	578	1082

[2] **Search Tool for the Retrieval of Interacting Genes/Proteins**

Annotazioni per il dataset C.elegans (WORM)

Per evitare di avere problemi nella fase di cross-validazione, i DAG sono stati ridotti a quei termini per cui si hanno almeno 10 annotazioni. Un po' di statistiche a seguito della selezione:

Onto	**numero di termini**	**media**	**d.std.**	**massimo**	**minimo**
BP	1335	71,33	151,68	2597	10
MF	186	61,23	191,84	1806	10
CC	221	131,9	302,25	1924	10

Tabella: La colonna *numero di termini* indica il numero di termini ottenuti dopo la selezione, *media* la media delle annotazioni per classe, per l'ontologia di riferimento, *d.std.* la deviazione standard delle annotazioni per l'ontologia di riferimento, *massimo* e *minimo* rispettivamente il massimo e minimo numero di annotazioni.

Algoritmi di ML per le predizioni flat

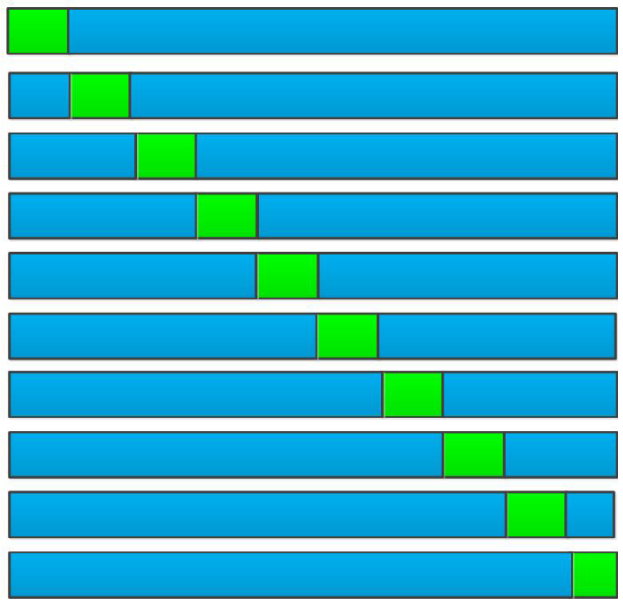
- Per quanto riguarda gli algoritmi di apprendmento automatico da utilizzare per le predizioni flat, si sono considerati i seguenti metodi:
  1. *K-Nearest Neighbors*
  2. *Logit Boost*
  3. *Linear Discriminant Analysis*
  4. *eXtreme Gradient Boosting*
  5. *C5.0* (Alberi di decisione)
  6. *Random Forest*
  7. *Multilayer Perceptron*
  8. *Support Vector Machine lineare*
  9. *Bagged CART* (Bagged ensemble di alberi di decisione)
  10. *AdaBoost.M1*
  11. *Naive Bayes*
  12. *Modelli Lineari Generalizzati*

Algoritmi di ML per le predizioni flat

- Per quanto riguarda gli algoritmi di apprendmento automatico da utilizzare per le predizioni flat, si sono considerati i seguenti metodi:
  1. *K-Nearest Neighbors*
  2. *Logit Boost*
  3. *Linear Discriminant Analysis*
  4. *eXtreme Gradient Boosting*
  5. *C5.0* (Alberi di decisione)
  6. *Random Forest*
  7. *Multilayer Perceptron*
  8. *Support Vector Machine lineare*
  9. *Bagged CART* (Bagged ensemble di alberi di decisione)
  10. *AdaBoost.M1*
  11. *Naive Bayes*
  12. *Modelli Lineari Generalizzati*
- Dato l'elevato numero di algoritmi selezionati per la sperimentazione, si è deciso di *non effettuare il tuning dei parametri*, questo per evitare di allungare ulteriormente i tempi dell'intero processo di valutazione e generazione degli score flat.

Cross-validazione e metriche usate per la valutazione degli algoritmi

- Per stimare le performance dei nostri predittori si è utilizzata la tecnica della *cross-validazione*.

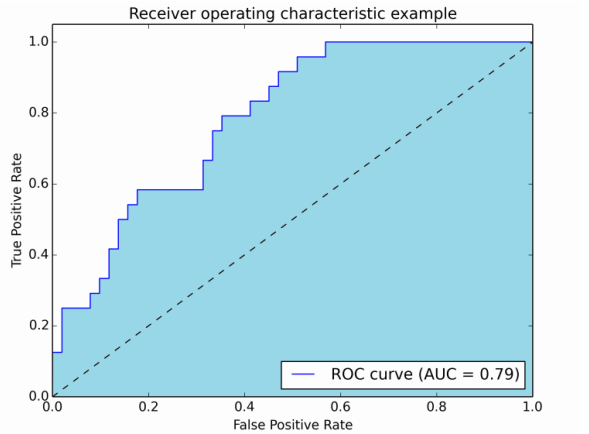


Cross-validazione e metriche usate per la valutazione degli algoritmi

- Per stimare le performance dei nostri predittori si è utilizzata la tecnica della *cross-validazione*.
- Come metriche si sono poi usate la:

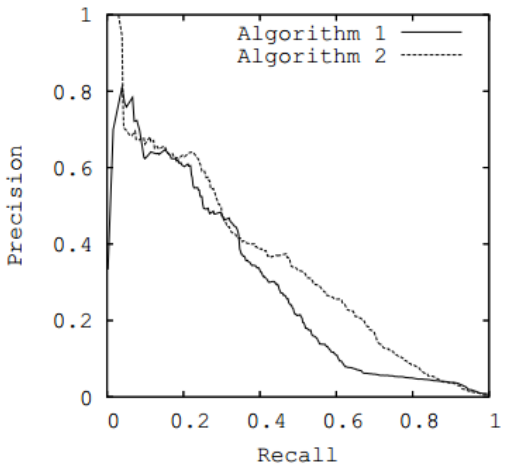
Cross-validazione e metriche usate per la valutazione degli algoritmi

- Per stimare le performance dei nostri predittori si è utilizzata la tecnica della *cross-validazione*.
- Come metriche si sono poi usate la:
  1. *AUROC*: mette in relazione le misure di Recall e False Positive Rate, al variare di una soglia applicata all'output del modello.



Cross-validazione e metriche usate per la valutazione degli algoritmi

- Per stimare le performance dei nostri predittori si è utilizzata la tecnica della *cross-validazione*.
- Come metriche si sono poi usate la:
  1. *AUROC*: mette in relazione le misure di Recall e False Positive Rate, al variare di una soglia applicata all'output del modello.
  2. *AUPRC*: è una misura che mette in relazione la variazione di Precision al variare della Recall.



Cross-validazione e metriche usate per la valutazione degli algoritmi

- Per stimare le performance dei nostri predittori si è utilizzata la tecnica della *cross-validazione*.
- Come metriche si sono poi usate la:
  1. *AUROC*: mette in relazione le misure di Recall e False Positive Rate, al variare di una soglia applicata all'output del modello.
  2. *AUPRC*: è una misura che mette in relazione la variazione di Precision al variare della Recall.
  3. *F-Score gerarchica*: è una metrica centrata sui geni, massimizzata al variare di una soglia t in (0, 1). Tale misura si basa sulla Precisione e Recall centrate sui geni e non su le classi.

$$Precision(t) = \frac{1}{n}\sum_{j=1}^{n}\frac{TruePositive_j(t)}{TruePositive_j(t) + FalsePositive_j(t)}$$

$$Recall(t) = \frac{1}{n}\sum_{j=1}^{n}\frac{TruePositive_j(t)}{TruePositive_j(t) + FalseNegative_j(t)}$$

$$Fmax = \max_t \frac{2Precision(t)Recall(t)}{Precision(t) + Recall(t)}$$

Stima preliminare dei tempi di calcolo degli algoritmi flat

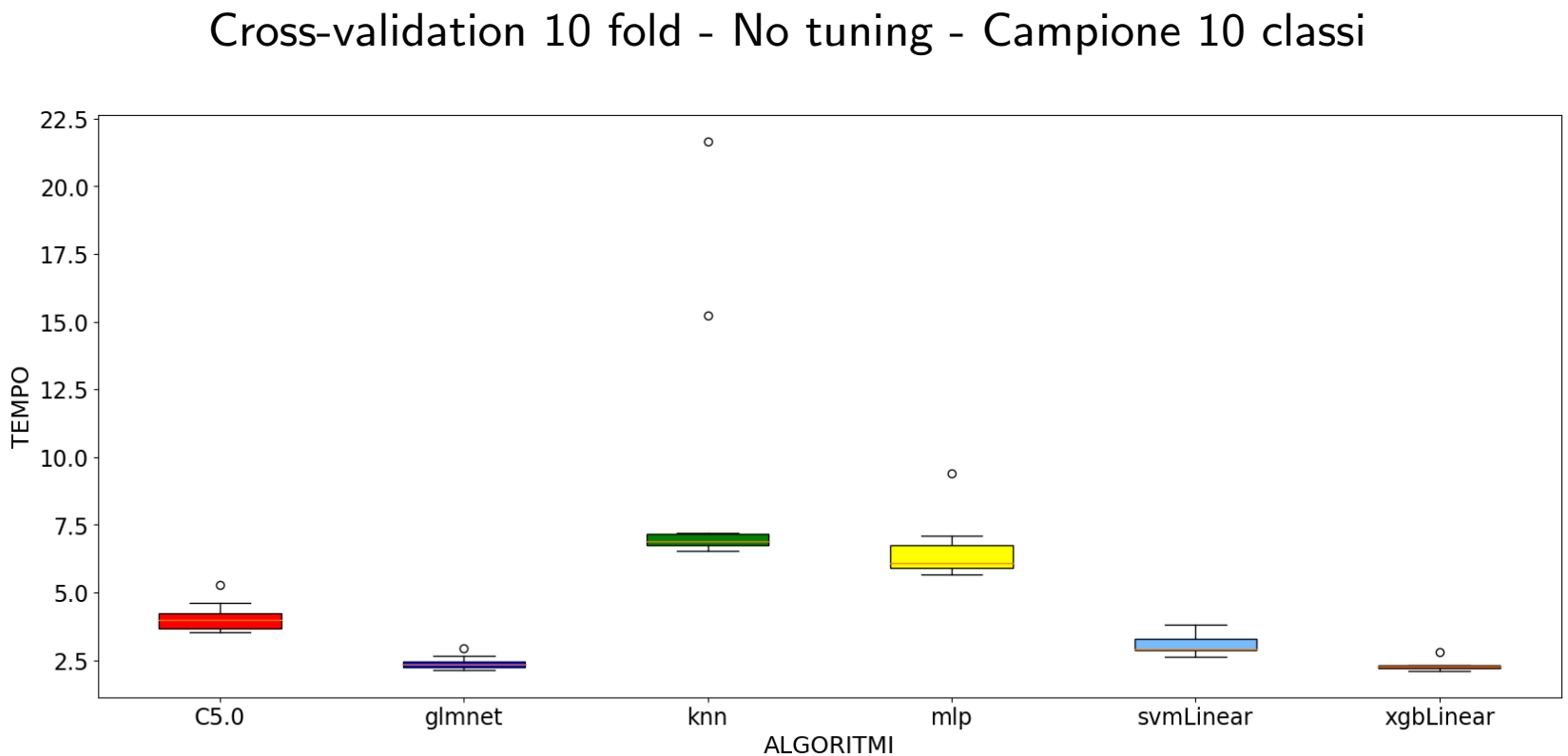
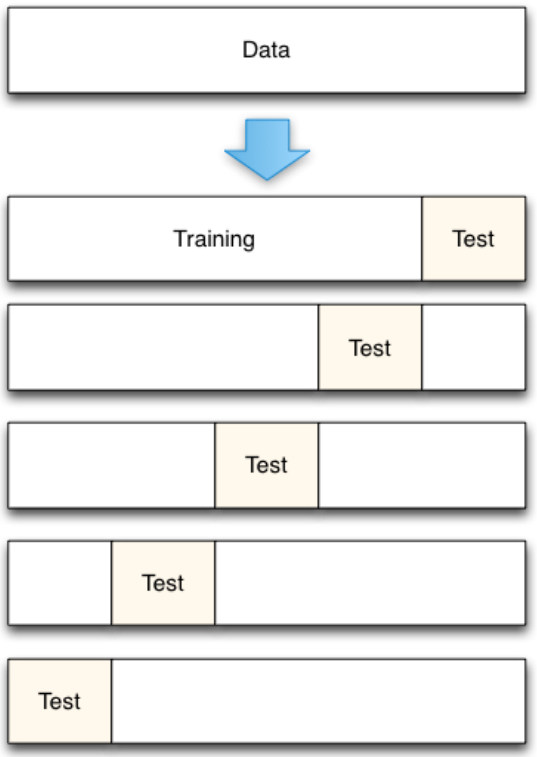


Figura: Il box plot dei tempi di esecuzione, con cross-validation a 10 fold per l' ontologia BP. I tempi sono da intendersi in ore e per classe, per un campione di 10 classi.

Riduzione dei tempi di calcolo del problema

- Si è prima di tutto ridotto il numero di fold da 10 a 5 nella cross-validation



Riduzione dei tempi di calcolo del problema

- Si è prima di tutto ridotto il numero di fold da 10 a 5 nella cross-validation
- Si sono provati due metodi di riduzione della dimensionalità:

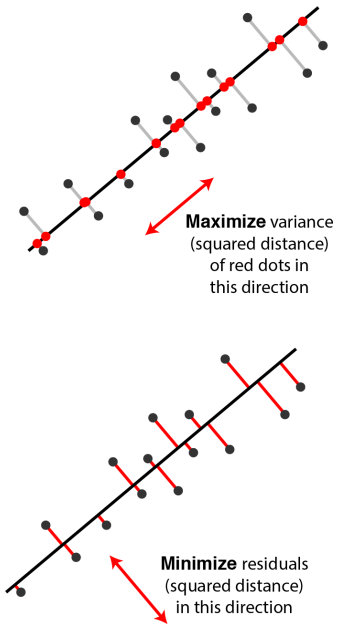
Riduzione dei tempi di calcolo del problema

- Si è prima di tutto ridotto il numero di fold da 10 a 5 nella cross-validation
- Si sono provati due metodi di riduzione della dimensionalità:
  1. **Selezione delle feature con Correlazione di Pearson** (FS).

$$corr_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y}$$

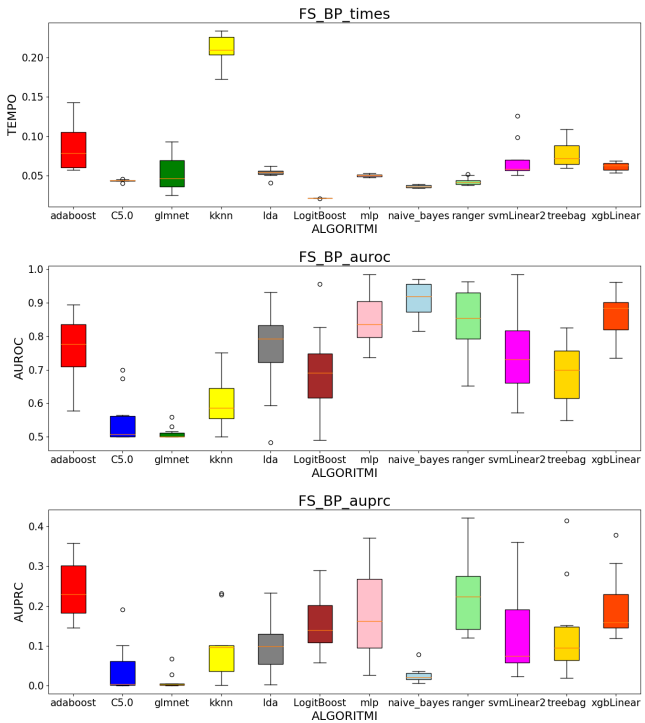
Riduzione dei tempi di calcolo del problema

- Si è prima di tutto ridotto il numero di fold da 10 a 5 nella cross-validation
- Si sono provati due metodi di riduzione della dimensionalità:
  1. **Selezione delle feature con Correlazione di Pearson** (FS).
  2. **Selezione delle componenti con la Principal Component Analysis** (PCA)



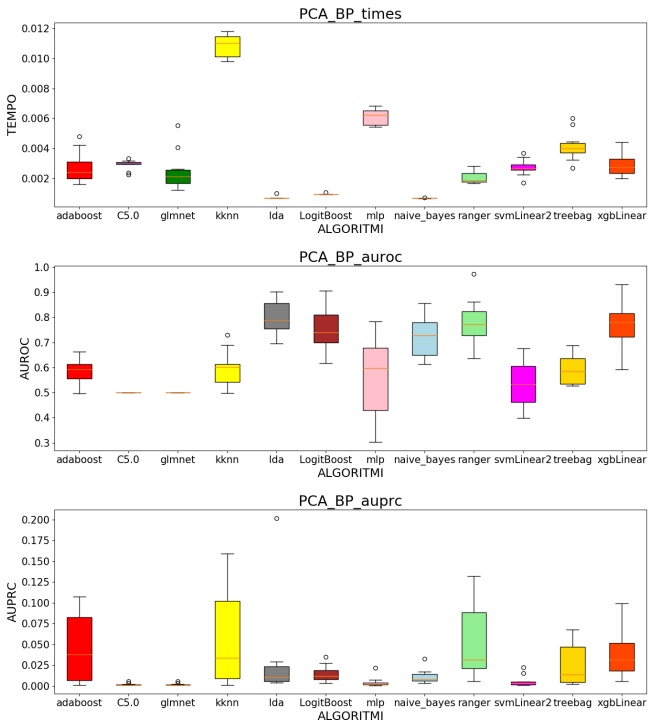
Riduzione dei tempi di calcolo del problema

- Si è prima di tutto ridotto il numero di fold da 10 a 5 nella cross-validation
- Si sono provati due metodi di riduzione della dimensionalità:
  1. **Selezione delle feature con Correlazione di Pearson** (FS).
  2. **Selezione delle componenti con la Principal Component Analysis** (PCA)
- Sempre facendo una stima sul medesimo campione di 10 classi, si sono testate:
  1. configurazioni per la FS per le prime 1000, 500 e 100 feature nel ranking ottenuto (a destra BoxPlot per le prime 100 feature selezionate)



Riduzione dei tempi di calcolo del problema

- Si è prima di tutto ridotto il numero di fold da 10 a 5 nella cross-validation
- Si sono provati due metodi di riduzione della dimensionalità:
  1. **Selezione delle feature con Correlazione di Pearson** (FS).
  2. **Selezione delle componenti con la Principal Component Analysis** (PCA)
- Sempre facendo una stima sul medesimo campione di 10 classi, si sono testate:
  1. configurazioni per la FS per le prime 1000, 500 e 100 feature nel ranking ottenuto (a destra BoxPlot per le prime 100 feature selezionate)
  2. configurazioni per la PCA, per le componenti che selezionano rispettivamente il 90%, il 70% e il 50% della varianza spiegata (a destra BoxPlot per le prime 15 componenti = 50% varianza spiegata)



Organizzazione degli esperimenti

Data l'analisi precedente effettuata sugli algoritmi di apprendimento flat (su di un campione di 10 classi), si è deciso di adottare il seguente set-up sperimentale:

- **cross-validation 5 fold + Selezione delle prime 100 feature con Correlazione di Pearson**.
- **cross-validation 5 fold + Selezione delle prime 15 componenti della PCA**.

Le quali hanno evidenziato un buon compromesso tra tempi di esecuzione e performance.

Organizzazione degli esperimenti

Data l'analisi precedente effettuata sugli algoritmi di apprendimento flat (su di un campione di 10 classi), si è deciso di adottare il seguente set-up sperimentale:

- **cross-validation 5 fold + Selezione delle prime 100 feature con Correlazione di Pearson**.
- **cross-validation 5 fold + Selezione delle prime 15 componenti della PCA**.

Le quali hanno evidenziato un buon compromesso tra tempi di esecuzione e performance.

Per quanto riguarda i metodi ensemble gerarchici, si sono utilizzati i seguenti metodi:

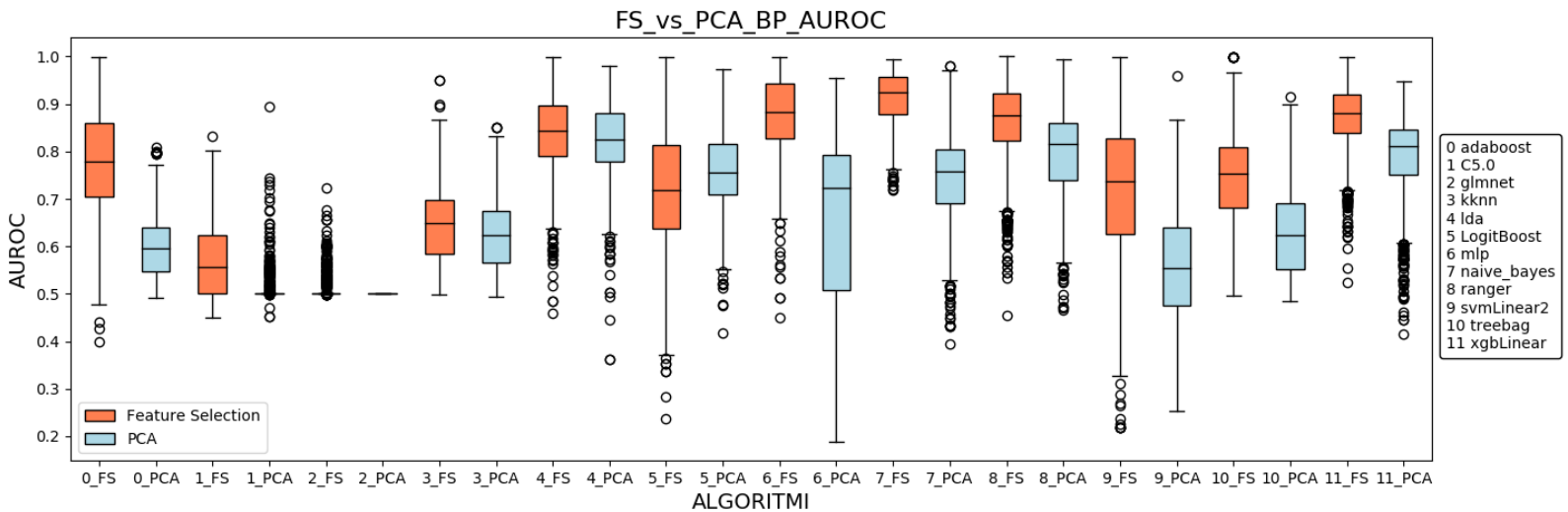
1. **HTD-DAG** [*Notaro et al., 2017*]
2. **TPR-DAG** (con variante senza soglia, con soglia adattiva, con pesi nell'aggiornamento) [*Notaro et al., 2017*]
3. **GPAV** (nuovo metodo introdotto in questa tesi)
4. **ISO-TPR** (nuovo metodo introdotto in questa tesi, con variante senza soglia e con soglia adattiva)

Performance degli algoritmi flat con FS e PCA

La generazione di tutti gli score flat, per entrambi i metodi di riduzione della complessità, ha evidenziato delle performance generalmente migliori per la selezione delle feature effettuta con la correlazione di Pearson (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$). Le performance migliori sono probabilmente da ricondurre al fatto che il tipo di riduzione per la correlazione è *supervisionato* a differenza della PCA.

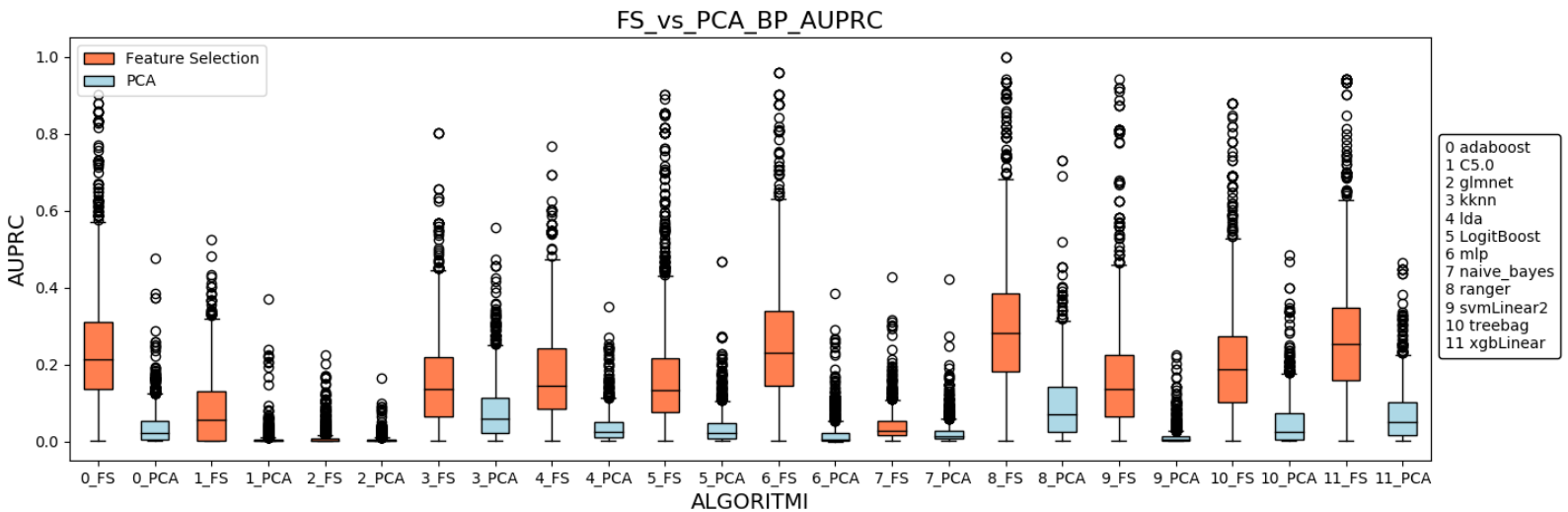
Performance degli algoritmi flat con FS e PCA

La generazione di tutti gli score flat, per entrambi i metodi di riduzione della complessità, ha evidenziato delle performance generalmente migliori per la selezione delle feature effettuta con la correlazione di Pearson (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$). Le performance migliori sono probabilmente da ricondurre al fatto che il tipo di riduzione per la correlazione è *supervisionato* a differenza della PCA.



Performance degli algoritmi flat con FS e PCA

La generazione di tutti gli score flat, per entrambi i metodi di riduzione della complessità, ha evidenziato delle performance generalmente migliori per la selezione delle feature effettuta con la correlazione di Pearson (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$). Le performance migliori sono probabilmente da ricondurre al fatto che il tipo di riduzione per la correlazione è *supervisionato* a differenza della PCA.



Risultati Metodi Ensemble

I metodi ensemble risultano essere più efficaci e riescono a migliorare in maniera statisticamente significativa i metodi flat (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$) quando è utilizzata la selezione delle feature con correlazione di Pearson come tecnica di riduzione della dimensionalità. (HTD è un'eccezione)

Risultati Metodi Ensemble

I metodi ensemble risultano essere più efficaci e riescono a migliorare in maniera statisticamente significativa i metodi flat (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$) quando è utilizzata la selezione delle feature con correlazione di Pearson come tecnica di riduzione della dimensionalità. (HTD è un'eccezione)

Confronto fra metodi gerarchici e metodi flat (AUROC).

(a) **[AUROC] [BP] [FS]**

METODI GERARCHICI	WIN-TIE-LOSS
GPAV	12-0-0
HTD	1-4-7
TPR-TF	11-1-0
ISO-TPR-TF	12-0-0
TPR-AT	5-1-6
ISO-TPR-AT	11-1-0
TPR-W	11-1-0

(b) **[AUROC] [BP] [PCA]**

METODI GERARCHICI	WIN-TIE-LOSS
GPAV	8-1-3
HTD	7-0-5
TPR-TF	8-1-3
ISO-TPR-TF	6-2-4
TPR-AT	6-2-4
ISO-TPR-AT	6-1-5
TPR-W	10-0-2

Risultati Metodi Ensemble

I metodi ensemble risultano essere più efficaci e riescono a migliorare in maniera statisticamente significativa i metodi flat (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$) quando è utilizzata la selezione delle feature con correlazione di Pearson come tecnica di riduzione della dimensionalità. (HTD è un'eccezione)

Confronto fra metodi gerarchici e metodi flat (AUPRC).

(a) **[AUPRC] [BP] [FS]**

METODI GERARCHICI	WIN-TIE-LOSS
GPAV	12-0-0
HTD	3-1-8
TPR-TF	12-0-0
ISO-TPR-TF	12-0-0
TPR-AT	9-2-1
ISO-TPR-AT	12-0-0
TPR-W	11-1-0

(b) **[AUPRC] [BP] [PCA]**

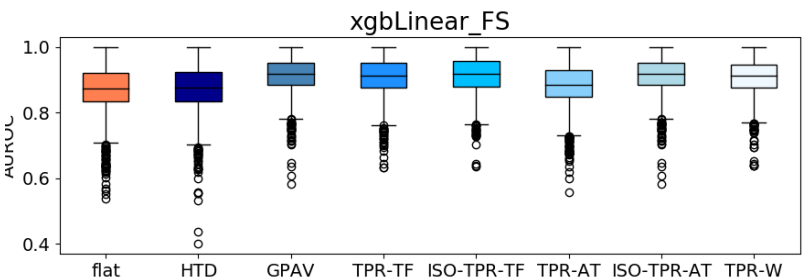
METODI GERARCHICI	WIN-TIE-LOSS
GPAV	8-3-1
HTD	6-2-4
TPR-TF	10-1-1
ISO-TPR-TF	7-2-3
TPR-AT	8-2-2
ISO-TPR-AT	6-2-4
TPR-W	10-0-2

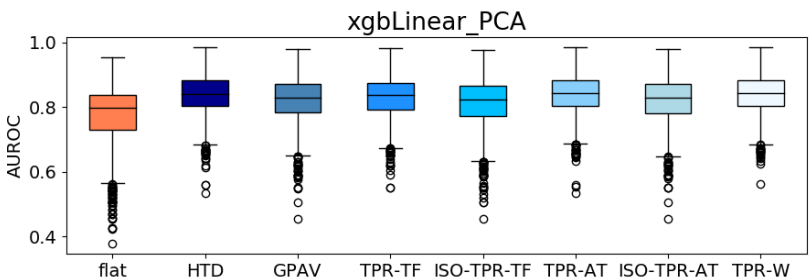
Risultati Metodi Ensemble

I metodi ensemble risultano essere più efficaci e riescono a migliorare in maniera statisticamente significativa i metodi flat (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$) quando è utilizzata la selezione delle feature con correlazione di Pearson come tecnica di riduzione della dimensionalità. (HTD è un'eccezione)

Risultati della AUROC ottenuti per Extreme gradient boosting (FS)

onto	flat	HTD	GPAV	TPR-TF	ISO-TPR-TF	TPR-AT	ISO-TPR-AT	TPR-W
BP	0.8699±0.002	0.8729±0.0021	**0.9121±0.0016**	**0.9081±0.0016**	**0.9118±0.0016**	**0.8818±0.0019**	**0.9122±0.0016**	**0.9069±0.0016**
MF	0.8535±0.0066	0.8475±0.0078	**0.8875±0.0059**	**0.8848±0.0061**	**0.887±0.006**	**0.8563±0.0071**	**0.8875±0.0059**	**0.8818±0.0062**
CC	0.86±0.0053	0.8593±0.0058	**0.8946±0.0049**	**0.8897±0.0049**	**0.8913±0.005**	**0.8652±0.0054**	**0.8946±0.0049**	**0.8912±0.0047**



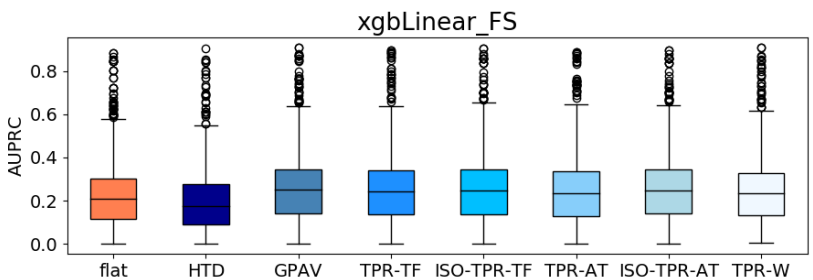


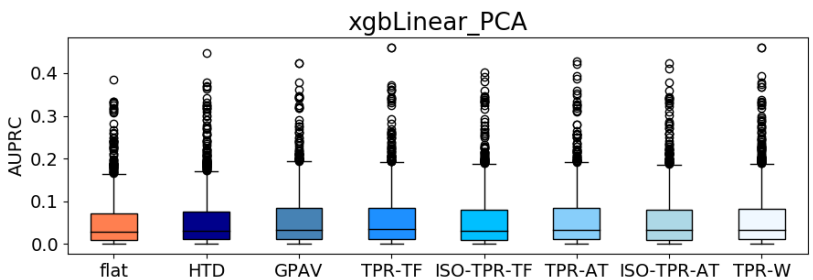
Risultati Metodi Ensemble

I metodi ensemble risultano essere più efficaci e riescono a migliorare in maniera statisticamente significativa i metodi flat (Test dei ranghi con segno di Wilcoxon, con significatività $\alpha = 0.01$) quando è utilizzata la selezione delle feature con correlazione di Pearson come tecnica di riduzione della dimensionalità. (HTD è un'eccezione)

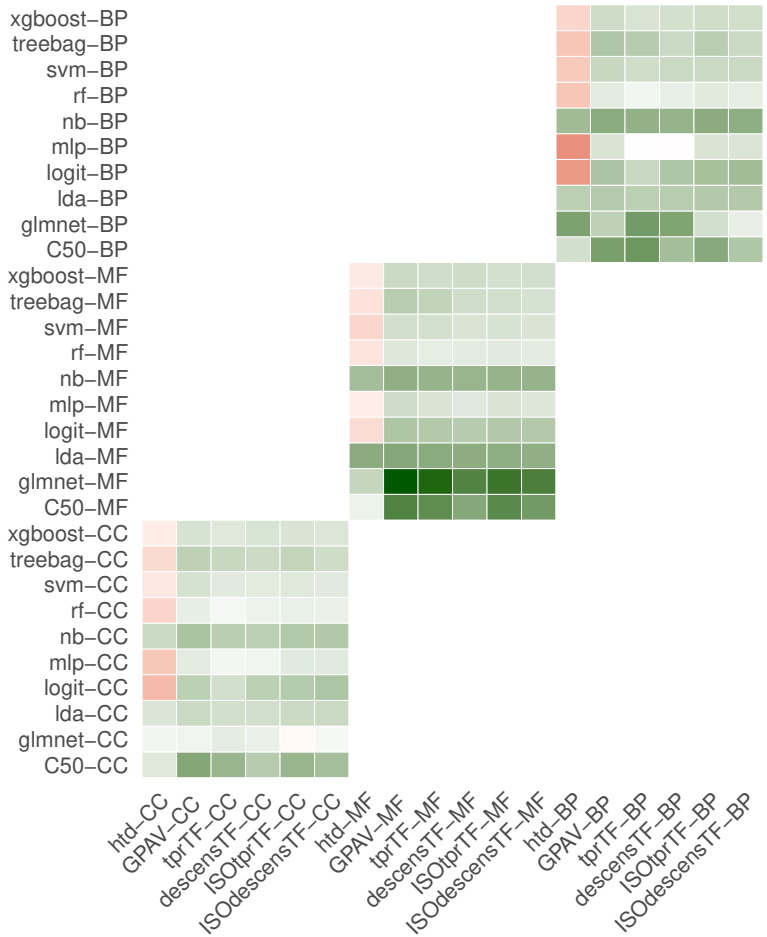
Risultati della AUPRC ottenuti per Extreme gradient boosting (FS)

onto	flat	HTD	GPAV	TPR-TF	ISO-TPR-TF	TPR-AT	ISO-TPR-AT	TPR-W
BP	0.2255±0.0041	0.1971±0.0039	**0.259±0.0044**	**0.2511±0.0043**	**0.2565±0.0043**	**0.2475±0.0043**	**0.2574±0.0043**	**0.2463±0.0042**
MF	0.1853±0.0094	0.1808±0.0104	**0.2114±0.01**	**0.2055±0.0101**	**0.2072±0.0098**	**0.197±0.0101**	**0.2065±0.0098**	**0.1976±0.0102**
CC	0.25±0.0116	0.2305±0.0118	**0.2739±0.012**	**0.2635±0.0119**	**0.2751±0.0122**	**0.2545±0.0119**	**0.2765±0.0121**	**0.2558±0.0118**





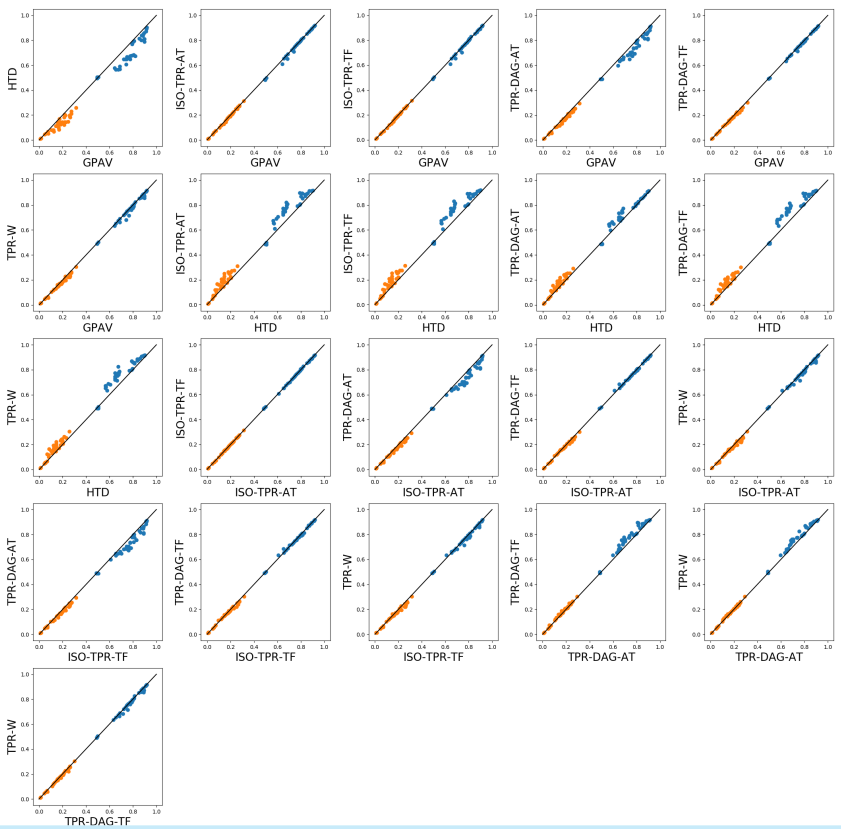
Heat Map metodi flat vs metodi gerarchici



Confronto tra metodi ensemble

AUPRC (arancio) AUROC(Blu) FS (36 punti = 12 algo * 3 onto)

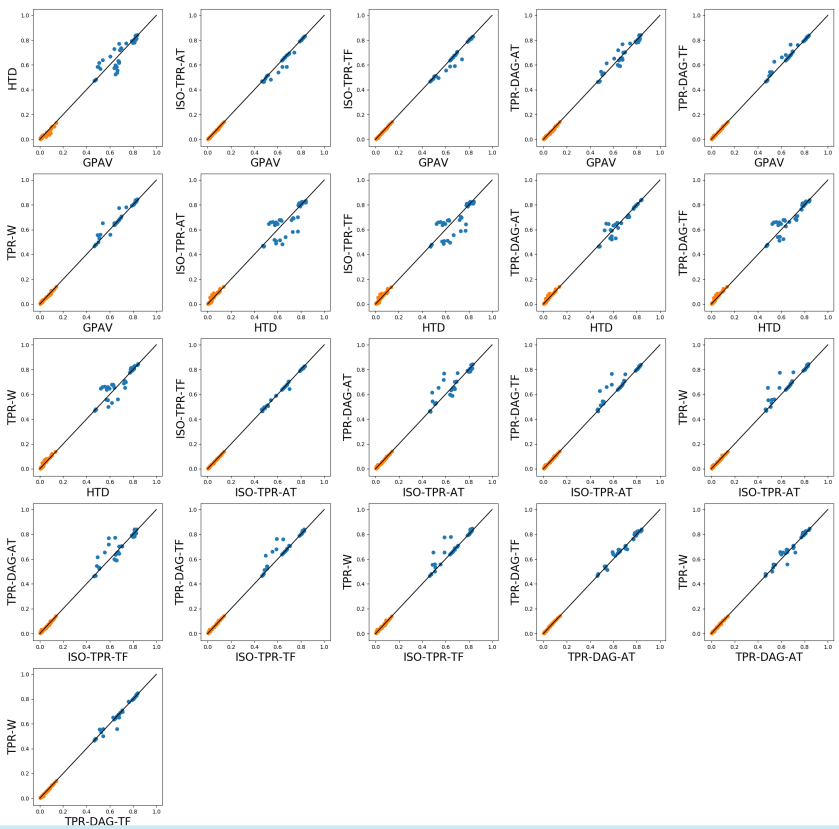
SCATTER FS AUROC & AUPRC



Confronto tra metodi ensemble

AUPRC (arancio) AUROC(Blu) PCA (36 punti = 12 algo * 3 onto)

SCATTER PCA AUROC & AUPRC



Conclusioni

- I metodi ensemble gerarchici riescono ad apportare miglioramenti statisticamente significativi alle performance dei predittori flat a patto di avere delle predizioni di partenza sufficientemente accurate.

Conclusioni

- I metodi ensemble gerarchici riescono ad apportare miglioramenti statisticamente significativi alle performance dei predittori flat a patto di avere delle predizioni di partenza sufficientemente accurate.
- La FS con correlazione di Pearson richiede tempi più lunghi della selezione con PCA, ma determina un vantaggio statisticamente significativo a livello di performance rispetto a quest'ultima (sia flat che ensemble).

Conclusioni

- I metodi ensemble gerarchici riescono ad apportare miglioramenti statisticamente significativi alle performance dei predittori flat a patto di avere delle predizioni di partenza sufficientemente accurate.
- La FS con correlazione di Pearson richiede tempi più lunghi della selezione con PCA, ma determina un vantaggio statisticamente significativo a livello di performance rispetto a quest'ultima (sia flat che ensemble).
- Il tuning degli algoritmi flat svolge un ruolo importante. Alcuni algoritmi hanno prodotto predittori flat scadenti, che non hanno consentito ai metodi di ensemble gerarchici di migliorare le performance (es. i modelli lineari generalizzati, addestrati con il package glmnet in R).

Conclusioni

- I metodi ensemble gerarchici riescono ad apportare miglioramenti statisticamente significativi alle performance dei predittori flat a patto di avere delle predizioni di partenza sufficientemente accurate.
- La FS con correlazione di Pearson richiede tempi più lunghi della selezione con PCA, ma determina un vantaggio statisticamente significativo a livello di performance rispetto a quest'ultima (sia flat che ensemble).
- Il tuning degli algoritmi flat svolge un ruolo importante. Alcuni algoritmi hanno prodotto predittori flat scadenti, che non hanno consentito ai metodi di ensemble gerarchici di migliorare le performance (es. i modelli lineari generalizzati, addestrati con il package glmnet in R).
- I nuovi metodi ensemble gerarchici introdotti in questa tesi (GPAV e ISO-TPR), si sono dimostrati competitivi rispetto ai metodi ensemble gerarchici allo stato dell'arte, e in alcuni casi hanno prodotto risultati significativamente migliori rispetto ad algoritmi gerarchici allo stato dell'arte.

Lavori futuri

- Ampliare il numero di specie su cui effettuare l'analisi.

Lavori futuri

- Ampliare il numero di specie su cui effettuare l'analisi.
- Effettuare un tuning più accurato degli algoritmi di machine learning selezionati, tenendo in considerazione che le classi dell'output strutturato sono spesso sbilanciate all'interno dei dataset per la GO.

Lavori futuri

- Ampliare il numero di specie su cui effettuare l'analisi.
- Effettuare un tuning più accurato degli algoritmi di machine learning selezionati, tenendo in considerazione che le classi dell'output strutturato sono spesso sbilanciate all'interno dei dataset per la GO.
- Introdurre nuove tecniche di riduzione della dimensionalità, sia per ridurre i tempi di calcolo (complessità) che per migliorare i risultati dei classificatori base (performance).

Lavori futuri

- Ampliare il numero di specie su cui effettuare l'analisi.
- Effettuare un tuning più accurato degli algoritmi di machine learning selezionati, tenendo in considerazione che le classi dell'output strutturato sono spesso sbilanciate all'interno dei dataset per la GO.
- Introdurre nuove tecniche di riduzione della dimensionalità, sia per ridurre i tempi di calcolo (complessità) che per migliorare i risultati dei classificatori base (performance).
- Utilizzare altre tipologie di selezione dei figli per lo step bottom-up della True Path Rule, in combinazione e non (a esempio effettuando una selezione dei figli pesata con soglia adattiva).

Lavori futuri

- Ampliare il numero di specie su cui effettuare l'analisi.
- Effettuare un tuning più accurato degli algoritmi di machine learning selezionati, tenendo in considerazione che le classi dell'output strutturato sono spesso sbilanciate all'interno dei dataset per la GO.
- Introdurre nuove tecniche di riduzione della dimensionalità, sia per ridurre i tempi di calcolo (complessità) che per migliorare i risultati dei classificatori base (performance).
- Utilizzare altre tipologie di selezione dei figli per lo step bottom-up della True Path Rule, in combinazione e non (a esempio effettuando una selezione dei figli pesata con soglia adattiva).
- Analizzare in maniera più approfondita le performance, ad esempio in relazione ad altre metriche, come la F-score gerarchica.

Lavori futuri

- Ampliare il numero di specie su cui effettuare l'analisi.
- Effettuare un tuning più accurato degli algoritmi di machine learning selezionati, tenendo in considerazione che le classi dell'output strutturato sono spesso sbilanciate all'interno dei dataset per la GO.
- Introdurre nuove tecniche di riduzione della dimensionalità, sia per ridurre i tempi di calcolo (complessità) che per migliorare i risultati dei classificatori base (performance).
- Utilizzare altre tipologie di selezione dei figli per lo step bottom-up della True Path Rule, in combinazione e non (a esempio effettuando una selezione dei figli pesata con soglia adattiva).
- Analizzare in maniera più approfondita le performance, ad esempio in relazione ad altre metriche, come la F-score gerarchica.
- Confrontare i metodi ensemble con i diversi metodi per la predizione su output strutturati, come ad esempio i metodi Kernel con output strutturato.

Grazie!

Domande?

Bibliografia

- G. Valentini, *Hierarchical Ensemble Methods for Protein Function Prediction*, ISRN Bioinformatics, anno 2014.
- G. Valentini, *True path rule hierarchical ensembles for genome-wide gene function prediction*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol.8, no. 3, pp. 832–847, anno 2011.
- N. Cesa-Bianchi, M. Re, G. Valentini, *Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference*, Machine Learning, vol.88(1), pp. 209-241, anno 2012.
- M. Notaro, M. Schubach, P. N. Robinson e Giorgio Valentini, *Prediction of Human Phenotype Ontology terms by means of hierarchical ensemble methods*, BMC Bioinformatics, vol 18, numero 1, pagine 449, 12 Ottobre, anno 2017.
- Burdakov O., Sysoev O., Grimvall A., Hussian M. *An O(n2) Algorithm for Isotonic Regression.*, Large-Scale Nonlinear Optimization. Nonconvex Optimization and Its Applications, vol 83. Springer, anno 2006.
- Szklarczyk D, Franceschini A, Wyder S, *STRING v10: protein–protein interaction networks*, integrated over the tree of life, Nucleic Acids Research, anno 2015.