UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

Dipartimento DI INFORMATICA

Dottorato di ricerca in Informatica

XXVII ciclo

Progetto di ricerca

Dottorando: Dott. Fulvio *Rotella*

Tutor: Prof. Stefano *Ferilli*

Coordinatore

Prof. Donato Malerba

Firma del dottorando _____

Firma del tutor _____

1 Titolo della ricerca:

Apprendimento probabilistico incrementale del prim'ordine in domini complessi.

2 Area nella quale si inquadra la ricerca:

L'area di ricerca in cui si colloca il presente progetto è quella dell'Apprendimento Automatico.

3 Obiettivi della ricerca

L'obiettivo della ricerca è centrato sullo studio e l'evoluzione di metodi di apprendimento automatico incrementali del prim'ordine attraverso l'integrazione di tecniche statistiche. Tale obiettivo mira ad integrare due approcci tipicamente umani, ovvero da un lato l'apprendimento induttivo a partire da esempi e dall'altro la capacità di utilizzare statistiche sui dati per garantire una sorta di affidabilità (o credenza) alle teorie apprese. In tal senso questo progetto vuole far cooperare l'approccio incrementale con l'approccio statistico in modo da guidare il primo nella formulazione di una teoria le cui componenti siano vere con una certa probabilità. L'applicazione dell'approccio risultante dalla cooperazione tra tali tecniche sarà applicato in domini reali quali l'elaborazione di dati sociali e biologici o l'elaborazione di documenti, gestendone anche aspetti complessi come la presenza di big data, l'incrementalità e la sequenzialità/evoluzione. Al fine di ottenere teorie logico-probabilistiche in scenari complessi si intende sviluppare metodi per l'apprendimento di parametri e struttura estendendo gli operatori di inferenza tradizionali in modo da renderli capaci di trattare l'incertezza.

4 Motivazioni della ricerca

L'apprendimento, la rappresentazione e il ragionamento numerico/statistico, da un lato, e relazionale, dall'altro, sono stati studiati di solito separatamente nell'area dell'Intelligenza Artificiale. Storicamente per un task di apprendimento si costruiscono feature di basso livello e si tenta di risolvere il problema adottando metodi di natura statistica. Ma questi approcci tradizionali non sempre sono in grado di catturare tutta la complessa rete di relazioni, spesso nascosta, tra eventi, oggetti o una loro composizione. Spesso, d'altra parte, si è interessati proprio a produrre e manipolare rappresentazioni complesse dei dati. Questa prima sfida è stata affrontata ricorrendo a linguaggi di rappresentazione del prim'ordine. In quest'area rientra l'apprendimento induttivo in logica del prim'ordine (I.L.P. , Inductive Logic Programming) [MR94], il cui sviluppo negli ultimi 40 anni è stato determinante per trattare tutti quei problemi la cui natura era nativamente relazionale. Ma per affrontare problemi nel mondo reale serve anche poter trattare l'incertezza che scaturisce da dati mancanti e dal rumore, ed eventualmente ottenere inferenze approssimate più che nette risposte si/no. L'incertezza nei dati relazionali complica la rappresentazione, in quanto essa può colpire gli attributi di un oggetto, il tipo (o più in generale l'identitità) di un oggetto e le relazioni in cui è coinvolto. Nel connubio tra l'apprendimento induttivo del primo ordine e quello statistico il primo fornisce il linguaggio di rappresentazione e gli strumenti di ragionamento, mentre il secondo garantisce robustezza. Questa nuova disciplina conosciuta come Probabilistic Inductive Logic Programming (PILP) [RK04] o Statistical Relational Learning (SRL) [Get02], sebbene mostri un'evidente rilevanza in domini complessi quali Social o Biological Data, estende i problemi già presenti nell'apprendimento automatico ereditandone di nuovi dai modelli grafici probabilistici (PGM) [Pea97] (apprendimento di parametri, apprendimento del modello e inferenza). Questi problemi sono stati affrontati in letteratura ricorrendo a numerose strategie, applicando sia nuove tecniche di ragionamento che algoritmi di ottimizzazione, ma tutt'ora non esistono metodi che, dati degli esempi, riescano ad apprendere in modo efficiente e/o efficace un modello (o una teoria) logico-probabilistico con i parametri opportuni. Inoltre è da notare che gli approcci di apprendimento statistico-relazionale mirano ad apprendere una teoria dopo aver raccolto quanti più esempi possibile. Tale approccio batch è poco adatto a gestire alcuni contesti reali, nei quali non è detto che si sappia quando si è raccolto il numero ottimale di esempi utili per apprendere il concetto target. Quindi la sfida consisterà nel trattare enormi quantità di dati sfruttando rappresentazioni relazionali, ma dalle

quali si cercherà di apprendere in modo incrementale la struttura e i parametri. Inoltre, dal momento che tipicamente i domini reali, come le reti sociali, si evolvono nel tempo, potrebbe essere necessario anche adottare metodologie per la ricerca di schemi tipici di cambiamenti strutturali in reti dinamiche.

5 Stato dell'arte

Molti approcci di apprendimento automatico in logica del prim'ordine partono da tecniche di ILP consolidate e le estendono con una semantica probabilistica. Dall'altra parte ci sono approcci che partono dai modelli grafici probabilistici e li estendono con rappresentazioni relazionali.

Tra gli approcci al problema in letteratura ricordiamo i Bayesian Logic Programs (BLPs) [KR07]. Un BLP consiste di due componenti: una logica rappresentata dall'insieme delle clausole Bayesiane, e una quantitativa definita dall'insieme dalle distribuzioni di probabilità sulle clausole. Nei BLP un programma logico C diventa un set di clausole della forma $h \leftarrow b_i$ dove h è un automo e i b_i rappresentano il corpo della clausola. Per ciascuna clausola in C, la probabilità $P(b_i|h)$ è la distribuzione di probabilità condizionale tale che per una sostituzione random θ per cui se $h\theta$ è ground ed è *vero* la query b_i ha successo (altrimenti fallisce). Si assume che la probabilità a priori di h $P(h)$ è la probabilità che per una sostituzione random θ, h sia vero. Un BLP insieme ad una background theory induce una Bayesian Network. Comunque le clausole di un programma logico bayesiano sono clausole regolari, infatti su di esse possono essere applicati operatori di raffinamento di ILP. Un algoritmo di apprendimento di BLP è rappresentato da Scooby [KR01], il quale tenta un approccio greedy hill-climbing. Esso prende in input un BLP iniziale e calcola i parametri massimizzando la likelihood. Poi applica operatori di raffinamento generalizzando e specializzando H per calcolare tutti i vicini nello spazio delle ipotesi. L'apprendimento quindi procede in due step che vengono iterati fino al raggiungimento della verosimiglianza (*likelihood*) desiderata, e quindi nel primo si apprende la struttura (o il modello) a partire dagli esempi utilizzando CLAUDIEN[RD97], poi si apprendono le probabilità. In caso di dati completi si tenta un approccio *Maximum Likelihood Estimation* altrimenti si ricorre ad *Expectation Maximization* o ad approcci gradient-based (meglio conosciuti come *hill-climbing*).

Un approccio più diretto per l'integrazione della probabilità nella programmazione logica è stato fornita da ProbLog [RKT07]. Essenzialmente questo linguaggio rappresenta Prolog dove tutte le clausole sono etichettate con una probabilità che indica il loro valore di verità. La nascita di ProbLog è stata motivata dalla necessità di apprendere reti biologiche in cui gli archi sono etichettati con probabilità. Un programma Problog specifica una distribuzione di probabilità su tutti i possibili possibili sottoprogrammi non-probabilistici. Dal momento che per apprendere un programa Prolog è possibile usare varie tecniche di ILP, il lavoro in questo framework si è concentrato sull'apprendimento dei parametri associati alla teoria logica. La probabilità di successo di una query q corrisponde alla probabilità che quella query ha una dimostrazione, data la distribuzione sui programmi logici. Sebbene ProbLog realizzi l'inferenza esatta, questa strategia è inattuabile anche per piccoli programmi logici. A tal fine sono stati sviluppati metodi di inferenza approssimata [KCR+08] che producono un gran numero di sottoprogrammi e li usano per stimare la probabilità. Un metodo di approssimazione è definuto *Bounded Approximation* in quanto si utilizzano formule DNF (Disjunctive Normal Form) per ottenere sia un upper che un lower bound alla probabilità di una query. L'algoritmo usa un albero SLD incompleto, ovvero un SLD-tree dove i branch sono estesi solo fino ad una data probabilità t, in modo da ottenere DNF formule per i due bound. La formula lower bound rappresenta tutte le dimostrazioni con una probabilità superiore la soglia corrente; la formula upper bound include tutte le derivazioni che sono state fermate a causa del raggiungimento della soglia. L'algoritmo procede in modo *iterative-deepening*, iniziando con una soglia di probabilità più alta e successivamente moltiplicando quella soglia per un fattore finchè la differenza tra i bound correnti diventa sufficientemente piccola. Un altro metodo è il K-Best, il quale usa un numero fisso di dimostrazioni in modo da approssimare la probabilità e permettere un miglior controllo sulla complessità globale che è un punto cruciale se ci sono da valutare moltissime query (es. nel caso dell'apprendimento dei parametri). Perciò la k-probability $P_k(q|T)$ approssima la probabilità di successo delle k-migliori spiegazioni invece di tutte le dimostrazioni. Altri possibili metodi di approssimizzazione sono quelli di Monte Carlo. In questi algoritmi vengono ripetutamente campionati programmi logici dal programma ProbLog e viene controllata la consistenza di alcune query di interesse. La frazione degli esempi dove la query è dimostrata è presa come stimatore della

probabilità della query, e dopo m campioni, viene calcolato il 95% di confidenza. Sebbene questi intervalli non corrispondono esattamente ai bound usati negli altri algoritmi di approsimazione, viene utilizzato un criterio di stop, che consiste nel considerare l'ampiezza dell'intervallo di confidenza fino a λ.

Un approccio differente alla programmazione non in clausole di Horn, è dato dalla programmazione logica disgiuntiva. La programmazione logica disgiuntiva consente di formalizzare, in modo semplice e naturale, problemi decisionali complessi attraverso la scrittura di programmi logici disgiuntivi (una collezione di regole logiche in cui è consentito l'uso sia della disgiunzione nella testa (antecedente) sia della negazione nel corpo (conseguente)). Gli LPAD (*Logic Programs with Annotated Disjunction*,[VVBA04] sono basati sulla programmazione logica disgiuntiva. Un LPAD consiste di un set di regole della forma $(h_1 : a_1)V(h_n : a_n) \leftarrow b_1, b_m.$ dove h sono atomi, b sono letterali e a sono numeri reali tra 0 e 1 la cui somma è 1. Gli LPAD possono essere istanziati scegliendo una testa per ogni clausola grazie ad una funzione di selezione. La probabilità di una selezione è data dal prodotto delle probabilità delle scelte individuali. Una volta istanziato un LPAD, diviene un programma logico tradizionale. Per quanto riguarda gli algoritmi di apprendimento della struttura e dei parametri, recentemente è stato presentato un approccio a tale task [Rig04]. L'algoritmo parte da alcuni gli esempi definiti come coppie $< I, Pr(I) >$ dove I sono le interpretazioni e $Pr(I)$ sono le probabilità associate, e da uno spazio di possibili LPADs definito dal linguaggio. Nel caso in cui $Pr(I)$ non sono disponibili, si può partire da un multiset di interpretazioni e quindi calcolate le probabilità usando le occorrenze. L'obiettivo è trovare uno o più LPAD che assegni ad ogni interpretazione la probabilità associata limitando la ricerca all'interno del language bias definito. In un altro lavoro [RD10] si applica l'approccio *Information Bottleneck* (IB), alla sottoclasse dei linguaggi SRL, che sono riducibili alle reti Bayesiane. Tale approccio afferma che quando le reti risultanti coinvolgono variabili nascoste, apprendere questi linguaggi richiede l'uso di tecniche di apprendimento da dati incompleti, come l'algoritmo EM. Il metodo IB [TPB99] è una tecnica che cerca il miglior tradeoff tra accuratezza e complessità quando si effettua il clustering di una variabile random X, data una distribuzione di probabilità congiunta tra X e la variabile osservata Y. Recentemente questo approccio è stato capace di evitare alcuni massimi locali nei quali EM viene intrappolato quando apprende con variabibili nascoste.

Nell'ambito dell'apprendimento statistico relazionale degne di nota sono le Markov Logic Network [RD06], le quali sono una combinazione tra reti Markoviane e logica del prim'ordine. Consideriamo una base di conoscenza del prim'ordine, essa è un insieme di hard constraints, (vincoli rigidi) sui possibili mondi, infatti se un mondo vìola anche solo una formula, esso ha probabilità zero. Diversamente la Markov Logic è basata sull'idea che quei vincoli devono essere rilassati: se un mondo vìola una formula, esso è meno probabile ma non impossibile. Ogni formula viene etichettata con un peso che riflette quanto forte è il vincolo: più grande è il peso, più grande è la differenza in log-probability tra un mondo che soddisfa la formule e un altro che non lo fa. Questo peso non rappresenta probabilità a differenza dei framework precedenti. Un insieme di formule in Markov Logic sono una Markov Logic Network (MLN). Quindi una Markov Logic Network N è un insieme di coppie $(F_i; w_i)$, dove F_i è una formula e w_i è un numero reale. Si può pensare una MLN come un template per generare Markov Network, infatti in mondi differenti, (differenti insieme di costanti), essa produrrà reti di grandezza differente, ma tutte con determinate regolarità nella struttura e nei parametri (per esempio tutti i grounding della stessa formula avranno lo stesso peso). Quindi atomi nel template produrranno nodi nella rete, formule nel template genereranno cricche nella rete, e c'è un arco tra due nodi se i corrispondenti atomi ground appaiono entrambi in almeno un grounding di una formula. Una MLN senza variabili è una Markov Network ordinaria. A differenza delle basi conoscenza del prim'ordine, una MLN può produrre risultati utili anche se ci sono delle contraddizioni: infatti una MLN può essere ottenuta fondendo diverse basi di conoscenza anche se parzialmente incompatibili. Bisogna ricordare che se i pesi crescono, la MLN tende ad assomigliare ad una pura base di conoscenza logica. Per quanto riguarda l'apprendimento di parametri e struttura sono stati prodotti numerosissimi lavori sia generativi che discriminativi. Gli approcci generativi ottimizzano le congiunte di tutte le variabili mentre quelli discriminativi propongono di massimizzare la conditional likelihood del predicato query. L'approccio discrimativo in [BFE08] sceglie le strutture che massimizzano la likelihood condizionale e l'insieme dei parametri per mezzo della metodologia maximum likelihood. Inizialmente l'approccio per l'apprendimento di una MLN seguiva quello di una BLP, ovvero in due fasi: prima con CLAUDIEN

si apprendeva la struttura e poi si introducevano le probabilità. Successivamente sono stati proposti metodi di apprendimento in un singolo step ottimizzando la pseudo-likelikelihood[KD05] in quanto i sistemi ILP sono progettati per apprendere teorie del prim'ordine che hanno una certa accuratezza e frequenza nei dati, e non nel massimizzare la likelihood sui dati che poi influenza la qualità delle predizioni di MLN.

6 Approccio al problema

L'approccio alla risoluzione dei problemi riguarderà l'estensione dei metodi di apprendimento incrementali del prim'ordine con componenti probabilistiche con l'obiettivo di permettere l'apprendimento di parametri e struttura.

I metodi di apprendimento induttivo in logica verranno esplorati al fine di apprendere la struttura ottimale sia con approccio batch che incrementale. Inoltre si ricorerrà ai linguaggi grafici per codificare la distribuzione di probabilità associata al modello appreso. L'adozione di questi ultimi sarà utile in quanto permetterà apprendimento di modelli che meglio si adattano agli esempi di addestramento. Dal momento che si desidera apprendere sia il modello che i parametri, ci saranno da fronteggiare due problemi: l'apprendimento della struttura di un programma logico, che in domini finiti è NP-difficile, e l'inferenza esatta, che per alcuni modelli grafici è computazionalmente intrattabile.

La ricerca sarà centrata su due punti focali: da un lato, la definizione di algoritmi di apprendimento automatico che limitano lo spazio di ricerca della struttura, dall'altro, metodi di inferenza approssimata che permettano di garantire la miglior aderenza tra la teoria logico-probabilistica e i dati di input.

Quando sarà possibile evitare l'inferenza approssimata, si cercherà di sviluppare metodi di *lifted probabilistic inference* [Poo03] in modo da effettuare inferenza esatta non incorrendo nell'esplicita enumerazione di tutti gli stati ma solo manipolando rappresentazioni degli stati nel prim'ordine.

7 Ricadute applicative

- L'interesse per la Social Network Analysis è motivato da numerosi possibili scenari, riguardanti l'estrazione di informazioni a differenti livelli di granularità focalizzandosi sull'intera rete o sui singoli oggetti. Per esempio, si può essere interessanti ad l'estrarre informazioni su una persona che non è esplicitamente descritta, ma emerge da considerazioni generali derivate dai suoi vicini diretti o indiretti nella rete; oppure si può essere interessati a scoprire gruppi emergenti di elementi che hanno comportamenti o gusti simili. La caratteristica peculiare che distingue i social network è l'esistenza di ricche raccolte di oggetti collegati in reti di relazioni complesse. La logica del prim'ordine è un ambiente che fornisce il potere espressivo per trattare queste relazioni, mentre le tecniche statistiche sono sufficientemente robuste per trattare grandi quantità di dati. I Semantic Network Service non sono limitati a relazioni di amicizia o professionali, ma esistono anche:
  - citation network, riguardanti la memorizzazione di articoli scientifici, e la relazione tra questi articoli e i loro argomenti attraverso autori e co-autori o relazioni mutue di referenza (es DBLP)
  - social shopping website, concentrati sull'e-commerce (es. Amazon) e sulla condivisione di opinioni su prodotti (es. Doyoo).
  - social media website, che forniscono suggerimenti su musica e film basandosi sulle preferenze dell'utente e su comportamenti tipici (Last.fm).

  Alcuni interessanti scenari applicativi che possono avere una rilevanza industriale sono:
  - *link prediction*, consiste nell'identificare se tra due attori può esserci una connessione o se in futuro potrebbe instaurarsi; in tal senso predire un link potrebbe interessare ad esempio in un contesto di e-commerce in quanto si potrebbero proporre oggetti simili di interesse non basandosi solamente su informazioni attributo-valore.

  - *community detection*, cerca di identificare comunità (gruppi di attori) sulla base della struttura della rete; questo scenario può essere utile sia nel contesto di e-commerce per la proposta di nuovi oggetti sul mercato che per l'analisi delle tendenze di gruppi, che per semplificare la fitta struttura della rete e manipolarla a diversi livelli di astrazione.
  - *outlier detection / object classification*, si cerca di predire correttamente la tipologia di attore e identificare individui con uno strano comportamente rispetto agli altri nella rete; potrebbe essere interessante ad esempio per scopi di sicurezza pubblica preventiva (ed. pedofilia, attentati).
  - *information diffusion* cioè lo studio di come l'informazione si propaga nelle reti sociali e capire quanto una tendenza emergente si diffonde in modo da studiare in modo profondo la rete e gli elementi; applicazioni spaziano dal commercio alla sociologia.
  - *viral marketing* si focalizza su come i suggerimenti o le opinioni su determinati prodotti fornite da alcuni elementi della rete, influenzino l'adozione di tali prodotti da parte degli elementi vicini nella rete.

- La bioinformatica è un dominio applicativo in cui l'informazione si può rappresentare naturalmente in termini di relazioni tra oggetti eterogenei. E' chiaro come gli sviluppi in tale settore possano influenzare l'industria chimico-farmaceutica. Fra i problemi/scenari rilevanti in tale ambito:
  - "How do proteins fold?". Uno dei task classici di biologia molecolare è il *Protein fold recognition* ovvero il problema di determinare se una data sequenza proteica si ripiega in una struttura precedentemente osservata. Una complicazione di incertezza consiste nel fatto che non è sempre vero che la struttura è stata precedentemente osservata.
  - *Genetic regulatory networks*, che consistono in un insieme di geni, proteine e piccole molecole e le loro interazioni. Molte di queste reti sono grandi e complesse e modellarne il comportamento è un task che coinvolge dimensioni temporali oltre che relazioni tra le componenti. L'obiettivo in questo caso può essere predire il comportamento della rete come un sistema dinamico.

Riferimenti bibliografici

[BFE08] Marenglen Biba, Stefano Ferilli, and Floriana Esposito. Discriminative structure learning of markov logic networks. In *ILP*, pages 59–76, 2008.

[Get02] Lise Carol Getoor. *Learning statistical models from relational data.* PhD thesis, Stanford, CA, USA, 2002. AAI3038093.

[KCR+08] Angelika Kimmig, Vítor Santos Costa, Ricardo Rocha, Bart Demoen, and Luc De Raedt. On the efficient execution of problog programs. In Maria Garcia de la Banda and Enrico Pontelli, editors, *ICLP*, volume 5366 of *Lecture Notes in Computer Science*, pages 175–189. Springer, 2008.

[KD05] Stanley Kok and Pedro Domingos. Learning the structure of markov logic networks. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 441–448, New York, NY, USA, 2005. ACM.

[KR01] Kristian Kersting and Luc De Raedt. Towards combining inductive logic programming with bayesian networks. In *ILP*, pages 118–131, 2001.

[KR07] Kristian Kersting and Luc De Raedt. *Bayesian Logic Programming: Theory and Tool*, chapter 10. MIT Press, 2007.

[MR94] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *JOURNAL OF LOGIC PROGRAMMING*, 19(20):629–679, 1994.

[Pea97] Judea Pearl. Graphical models for probabilistic and causal reasoning. In *The Computer Science and Engineering Handbook*, pages 697–714. 1997.

[Poo03] David Poole. First-order probabilistic inference. In *IJCAI*, pages 985–991, 2003.

[RD97] Luc De Raedt and Luc Dehaspe. Clausal discovery. *Machine Learning*, 26(2):99–146, 1997.

[RD06] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[RD10] Fabrizio Riguzzi and Nicola Di Mauro. Applying the information bottleneck approach to srl: Learning lpad parameters. In *The 20th International Conference on Inductive Logic Programming (ILP10)*, 2010. (<u>see the accompanying technical report Application of the Information Bottleneck to LPAD Learning</u>).

[Rig04] Fabrizio Riguzzi. Learning logic programs with annotated disjunctions. In *ILP*, pages 270–287, 2004.

[RK04] Luc De Raedt and Kristian Kersting. Probabilistic inductive logic programming. In *ALT*, pages 19–36, 2004.

[RKT07] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In Manuela M. Veloso, editor, *IJCAI*, pages 2462–2467, 2007.

[TPB99] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. pages 368–377, 1999.

[VVBA04] Joost Vennekens, Sofie Verbaeten, Maurice Bruynooghe, and Celestijnenlaan A. Logic programs with annotated disjunctions. In *In Proc. Int'l Conf. on Logic Programming*, pages 431–445. Springer, 2004.

8 Fasi del progetto

Si prevede una suddivisione in tre fasi corrispondenti agli anni di dottorato.
Prima fase:

- studio dello stato dell'arte e definizione di prototipi volti alla risoluzione del problema. Le attività riguarderanno:
  - studio dei metodi di apprendimento automatico del prim'ordine già esistenti in letteratura;
  - studio dei modelli grafici probabilistici per la definizione di algoritmi di apprendimento su grafi;
  - partecipazione a conferenze e scuole estive inerenti all'ambito di ricerca.

Seconda fase:

- definizione di algoritmi di apprendimento probabilistico relazionali per l'apprendimento di modelli e parametri. In particolare, si applicheranno:
  - operatori di raffinamento tipici di ILP estesi con probabilità;
  - algoritmi di approssimazione per l'apprendimento di programmi logici-probabilistici;
- sperimentazioni delle tecniche su dati sociali;
- pubblicazione dei risultati in conferenze nazionali e internazioni su temi affini;
- stage presso un'università straniera.

Terza fase:

- analisi sperimentale dei metodi proposti;
- stesura della tesi.

9 Valutazione dei risultati

Tabella 1: Notazioni

Risultato del test	Positivi reali	Negativi reali	Totali di riga
Positivi	TP	FP	TP + FP (numero di istanze con test positivo)
Negativi	FN	TN	FN + TN (numero di istanze con test negativo)
Totali di colonna	TP+FN (numero di istanze con una determinata condizione)	FP+TN (numero di istanze senza una determinata condizione)	TP+TN+FP+FN (numero totale di istanze)

Alcune metriche potranno essere le seguenti:

- *Area under the ROC curve*, tale metrica permette di stimare la probabilità di assegnare un unità statistica al suo reale gruppo di appartenenza e quindi valutare la bontà del metodo usato per la classificazione. La curva di ROC è definita sulle ascisse dal False Positive Rate $FP/(FP+TN)$ (oppure 1-Specificity) mentre sulle ordinate dal True Positive Rate $TP/(TP+FN)$ o (Sensitivity).
- *Precision* è definita come il rapporto $TP/(TP+FP)$ ed indica il numero di veri positivi (il numero di oggetti etichettati correttamente come appartenenti alla classe) diviso il numero totale di elementi etichettati come appartenenti alla classe (la somma di veri positivi e falsi positivi, che sono oggetti etichettati erroneamente come appartenenti alla classe)
- *Recall* è definita come il rapporto $TP/(TP+FN)$ ovvero il numero di veri positivi diviso il numero totale di elementi che attualmente appartengono alla classe (per esempio la somma di veri positivi e falsi negativi, che sono oggetti che non sono stati etichettati come appartenenti alla classe ma dovrebbero esserlo).
- *Accuracy* è definita come il rapporto $TP+TN/(TP+TN+FP+FN)$ ovvero il numero dei risultati corretti (veri positivi e negativi) diviso il numero totale di elementi.

Si sceglierà in base agli obiettivi e al dominio applicativo, i dati e le metriche opportune per poter effettuare test sperimentali per validare le strategie proposte.

10 Eventuali referenti esterni al Dipartimento