

Apprendimento a ECAI 98

Fabrizio Riguzzi

La sessione "Numerical Methods in Machine Learning" comprendeva gli articoli:

"Using a Data Metric for Preprocessing Advice for Data Mining Applications" di Robert Engels e Christiane Theusinger;

"Implicit Feature Selection with the Value Difference Metric" di Terry R. Payne e Peter Edwards;

"Collective Learning in Multi Agent Systems" (young researcher paper) di Stéphane Calderoni;

"Simple Synchrony Networks: Learning Generalizations across Syntactic Constituents" (young researcher paper) di Peter Lane.

La sessione "Inductive Logic Programming" comprendeva gli articoli

"An Integrated Framework of Learning Numerical Terms in FOL" di Marco Botta, Attilio Giordana e Roberto Piola;

"Learning Recursive Theories with ATRE" di Donato Malerba, Floriana Esposito e Francesca Alessandra Lisi;

"Version Space Learning with Instance-Based Boundary Sets" (vincitore del premio come uno dei tre migliori articoli) di Evgueni N. Smirnov e Peter J. Braspenning;

"Synthesizing Inductive Logic Programming and Genetic Programming" (young researcher paper) di Ryutaro Ichise;

"Integrating Abduction and Induction" (young researcher paper) di Fabrizio Riguzzi.

"Using a Data Metric for Preprocessing Advice for Data Mining Applications" di Robert Engels e Christiane Theusinger si occupa del pre-processing dei dati nella Knowledge Discovery in Databases. L'articolo propone un processo che parte da una descrizione informale delle specifiche utente di un problema di KDD tipico per ottenere i requisiti utente funzionali e non funzionali. Questi requisiti sono poi utilizzati in combinazione con metadati, ovvero dati che descrivono oggettivamente i dati insieme a informazioni statistiche calcolate a partire dai dati iniziali, allo scopo di presentare una soluzione all'utente. Tale soluzione di solito assume la forma di una istanziazione dei parametri di un processo KDD parametrico in uno strumento software commerciale. I metadati sono calcolati utilizzando un tool che prende il nome di DCT (Data Characterisation Tool) che è stato incluso nell'ambiente software Clementine. Vengono presentati alcuni esempi basati sui dataset di UCI.

L'articolo "Implicit Feature Selection with the Value Difference Metric" di Terry R. Payne e Peter Edwards riguarda il problema degli attributi irrilevanti nell'apprendimento mediante algoritmi di tipo nearest neighbour. Gli autori considerano una particolare metrica per la misura della distanza tra le istanze, la Value Difference Metric (VDM), e mostrano come tale metrica possa essere utilizzata per ridurre l'impatto degli attributi irrilevanti sulla accuratezza nella classificazione.

Tale metrica usa delle semplici tecniche probabilistiche per pesare gli attributi in maniera diversa nello spazio degli stati. Il calcolo di tali pesi risulta avere un costo computazionale inferiore a quello delle tecniche per il trattamento degli attributi irrilevanti basate sul pre-processing dei dati.

Sono stati realizzati esperimenti che confrontano alcune metriche tradizionali con metriche che utilizzano tali tecniche probabilistiche su una serie di dataset reali e artificiali.

L'articolo "Collective Learning in Multi Agent Systems" (young research paper) di Stéphane Calderoni descrive un approccio per l'apprendimento collettivo in una società di agenti autonomi. L'apprendimento avviene a due livelli: a livello individuale, ciascun agente impara dalla propria esperienza mediante un algoritmo di reinforcement learning, a livello collettivo, la società di agenti apprende combinando la conoscenza dei singoli agenti mediante un algoritmo genetico.

Ciascun agente mantiene una matrice di scelta che contiene i valori della funzione di rinforzo per ciascuna coppia condizione-azione. Tale matrice viene codificata in un cromosoma e viene inviata da ciascun agente agli agenti circostanti. Essi selezionano i migliori ed eseguono incroci e mutazioni con una certa frequenza, facendo così evolvere la conoscenza del sistema.

Il metodo proposto è stato verificato sperimentalmente utilizzando un sistema per la modellazione e simulazione di società di agenti artificiali scritto in Java. Il problema considerato per l'esperimento è quello del trasporto del cibo in una colonia di formiche: intorno al nido delle formiche sono disposte un certo numero di ammassi di cibo. Le formiche devono localizzare gli ammassi partendo dal nido e devono riportare il cibo verso il nido, lasciando feromoni per indicare il cammino alle formiche successive. L'esperimento ha mostrato che il numero di unità di cibo portate al nido cresce esponenzialmente con il trascorrere del tempo, evidenziando così gli effetti dell'apprendimento sulle prestazioni della colonia.

L'articolo "Simple Synchrony Networks: Learning Generalizations across Syntactic Constituents" (young research paper) di Peter Lane descrive un algoritmo di apprendimento per un nuovo tipo di architettura connessionista chiamato Simple Synchrony Networks (SSN) e riporta i risultati di esperimenti sull'apprendimento di grammatiche ricorsive. Le SSN combinano una tecnica per l'apprendimento di pattern su serie temporali, Simple Recurrent Network (SRN), con Temporal Synchrony Variable Binding (TSVB). TSVB è una tecnica che consente di apprendere separatamente informazioni riguardanti diverse entità del training set e di generalizzare tali informazioni sulle entità del test set. Negli esperimenti, il training viene fatto su frasi che contengono al massimo una proposizione al loro interno con alcune parole ristrette a certe classi di costituenti. Nel testing, la rete generalizza le informazioni apprese a frasi che contengono fino a tre proposizioni al loro interno, e con parole che possono apparire in ogni costituente. Questi risultati dimostrano che le SSN sono in grado di apprendere generalizzazioni di costituenti sintattici.

L'articolo "An Integrated Framework of Learning Numerical Terms in FOL" di Marco Botta, Attilio Giordana e Roberto Piola presenta un metodo per il raffinamento di letterali numerici in basi di conoscenza espresse in un linguaggio logico del primo ordine. Il metodo consiste nel rappresentare i letterali numerici mediante funzioni continue parametrizzate e utilizzare un algoritmo di "error gradient descent" per determinare i parametri. I letterali numerici ammessi hanno la forma $\text{Inside}(f(x_1, \dots, x_n), [a, b])$ dove $f(x_1, \dots, x_n)$ è una funzione reale e $[a, b]$ un intervallo chiuso. Tale letterale può essere interpretato come una funzione che assume il valore 1 nell'intervallo $[a, b]$ e 0 al di fuori. La forma rettangolare della funzione Inside viene approssimata da una funzione continua a campana al fine di utilizzare la Delta-Rule per compiere la discesa della superficie che rappresenta l'errore come viene fatto nelle reti neurali multistrato. La teoria da raffinare può essere codificata manualmente oppure acquisita automaticamente da un sistema di apprendimento simbolico di teorie del primo ordine in grado di trattare attributi numerici. L'articolo presenta una estesa sperimentazione del metodo proposto che comprende tre casi di studio. I risultati mostrano che il passo di raffinamento conduce sempre ad un significativo miglioramento dell'accuratezza di classificazione.

L'articolo "Learning Recursive Theories with ATRE" di Donato Malerba, Floriana Esposito e Francesca Alessandra Lisi presenta un nuovo approccio per l'apprendimento di teorie ricorsive. Al fine di imparare definizioni di predicati mutuamente ricorsivi, clausole per predicati diversi vengono apprese alternativamente utilizzando una strategia di ricerca "separate-and-parallel-conquer". Nella fase di separate l'algoritmo seleziona una serie di esempi positivi che costituiscono i semi e lancia una fase di conquer in parallelo per ciascun seme, mentre nella fase di conquer l'algoritmo cerca una clausola consistente che copra l'esempio positivo fornito come seme. La ricerca nella fase di conquer percorre lo spazio delle clausole da generale a specifico, sincronizzandosi ad ogni livello con le fasi di conquer parallele, in modo che si proceda al livello successivo solo quando tutte le fasi di conquer parallele hanno terminato un livello. In questo modo si assicura la generazione delle clausole consistenti più semplici. Apprendendo clausole per predicati diversi alternativamente, si incorre nel problema della non-monotonicità della proprietà di consistenza: aggiungendo una clausola consistente ad una teoria consistente si può ottenere una teoria inconsistente. Al fine di risolvere questo problema, la teoria corrente viene riformulata prima di aggiungere la clausola successiva: i predicati nelle teste delle clausole apprese vengono rinominati in modo tale da non entrare in conflitto con la nuova clausola. L'approccio proposto è realizzato in un nuovo sistema chiamato ATRE che utilizza una rappresentazione degli esempi basata su oggetti.

L'articolo "Version Space Learning with Instance-Based Boundary Sets" (vincitore del premio come uno dei tre migliori articoli) di Evgeni N. Smirnov e Peter J. Braspenning presenta una nuova tecnica di rappresentazione degli spazi delle versioni che rende l'apprendimento in questo contesto polinomiale sotto certe condizioni.

Un problema di apprendimento nello spazio delle versioni consiste nel cercare un insieme VS di descrizioni del concetto target che sono consistenti con le istanze di training, ovvero tali che ciascuna descrizione classifichi correttamente tutte le istanze positive e tutte quelle negative. Il problema viene risolto considerando le istanze una ad una e aggiornando l'insieme VS in modo tale che le descrizioni inconsistenti siano rimosse. L'apprendimento finisce quando VS ha un solo elemento che descrive intensionalmente il concetto. Gli algoritmi per l'apprendimento sono caratterizzati da sette operazioni fondamentali, di cui le più importanti sono l'operazione di $\text{Update}(i, VS)$ che aggiorna VS con l'istanza i e l'operazione di intersezione tra due spazi delle versioni.

Uno spazio delle versioni VS può venire rappresentato mediante i due insiemi di confine S e G che contengono, rispettivamente, le descrizioni del concetto target più specifiche e più generali contenute in VS, se il linguaggio dei concetti L_c soddisfa alcune condizioni dette di ammissibilità. Utilizzando questa rappresentazione per lo spazio delle versioni, le operazioni sopra citate non sono trattabili, ovvero hanno costo esponenziale in quanto la dimensione di S e G cresce esponenzialmente con il numero di esempi.

L'articolo propone una rappresentazione dello spazio delle versioni mediante insiemi di confine basati sulle istanze. Uno spazio delle versioni è rappresentato da una coppia ordinata di sequenze di insiemi di confine $\langle\langle S_1, \dots, S_p \rangle, \langle G_1, \dots, G_n \rangle\rangle$ dove S_p è l'insieme delle descrizioni più specifiche dello spazio delle versioni ottenuto considerando l'istanza positiva ip e tutte le istanze negative, mentre G_n è l'insieme delle descrizioni più generali dello spazio delle versioni ottenuto considerando l'istanza negativa in e tutte le istanze positive.

Le operazioni di base sopra citate diventano in questo caso polinomiali su una serie di grandezze che non dipendono dal numero di esempi ma solo dal linguaggio: la più grande dimensione degli insiemi S e G di uno spazio delle versioni riferito ad una istanza e la complessità temporale della costruzione di uno di tali spazi delle versioni.

"Synthesizing Inductive Logic Programming and Genetic Programming" (young research paper) di Ryutaro Ichise descrive il sistema SYNGIP che apprende teorie del primo ordine mediante una strategia di ricerca di tipo genetico. Tale strategia di ricerca offre il vantaggio di essere meno sensibile ai minimi locali e di gestire spazi di ricerca molto ampi. Per applicare tecniche di Genetic Programming, è necessario definire un individuo. L'approccio proposto considera due definizioni, una in cui l'individuo corrisponde a una clausola di Horn e un gene corrisponde ad un letterale e l'altra in cui l'individuo corrisponde ad un programma e un gene ad una clausola. Le tecniche di incrocio e mutazione modificano le clausole in modo tale da rispettare le dichiarazioni di tipo e modo dei vari predicati. Viene utilizzata una funzione di fitness che tiene conto sia dell'accuratezza della clausola che della sua semplicità. Mentre i sistemi di ILP possono gestire solo esempi che hanno una classificazione discreta, utilizzando una strategia di ricerca genetica è possibile gestire esempi aventi una classificazione continua.

"Integrating Abduction and Induction" (young research paper) di Fabrizio Riguzzi presenta un approccio per l'integrazione di abduzione e induzione che consiste nell'estendere un sistema di ILP con capacità di ragionamento abduttivo.

L'algoritmo è ottenuto estendendo l'algoritmo top-down di base. L'abduzione è utilizzata per coprire gli esempi: nel caso un esempio positivo non sia derivabile a partire dalle informazioni conosciute, l'abduzione è utilizzata per compiere delle assunzioni che consentano, una volta aggiunte alla teoria, di coprire l'esempio. Allo stesso modo, assunzioni di assenza di atomi sono compiute per assicurare che gli esempi negativi non siano coperti. Le assunzioni generate possono riguardare predicate background oppure target: nel caso in cui riguardino predicati target, sono considerate come nuovi esempi e aggiunti al training set.

Tale algoritmo fornisce uno schema che può essere specializzato per risolvere i seguenti problemi: apprendimento da conoscenza incompleta, apprendimento di teorie abduttive, apprendimento di eccezioni, apprendimento di predicati multipli e apprendimento di programmi logici con negazione.