UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

DIPARTIMENTO DI INFORMATICA

Dottorato di ricerca in Informatica e Matematica
XXXII ciclo

Progetto di ricerca

Dottorando: Dott. *Sergio Angelastro*

Tutor: Prof. *Stefano Ferilli*

Firma del dottorando _____

Firma del tutor _____

1. Titolo della ricerca:

Incremental Mining of Declarative Complex Process Models.

2. Area nella quale si inquadra la ricerca:

Apprendimento automatico e Data Mining.

3. Motivazioni e obiettivi della ricerca

La gestione di modelli di processo ha acquisito rapidamente interesse e importanza nell'industria e nella ricerca. A tal proposito è nata l'area del business process management [9] nella quale sono state sviluppate tecniche e metodologie di analisi dei processi e grazie alle quali diversi settori, come quelli industriali, hanno tratto grandi vantaggi in termini di comprensione e supporto alla progettazione e gestione dei processi.
Inoltre, produrre modelli manualmente è un compito inerentemente costoso, complesso e soggetto ad errori [1] e dunque sorge il problema di adattare e raffinare modelli esistenti, laddove siano disponibili, in ambienti dinamici [4]. Da qui nasce la forte motivazione nello studio dei processi e delle tecniche per modellarli e apprenderli (almeno parzialmente) automaticamente.
Con l'avvento di nuovi domini applicativi, come ambienti intelligenti (AmI), sono nati nuovi problemi e sfide. In AmI si sviluppano soluzioni per l'assistenza alle persone per migliorare la qualità della loro vita. Gli ambienti dovrebbero essere predisposti per reperire conoscenza su preferenze e bisogni dei soggetti coinvolti [2,3], anticipando in maniera proattiva le loro azioni. Una soluzione è quella di utilizzare modelli di alto livello che astraggano pattern di attività e processi comportamentali, garantendone una migliore comprensione, come i modelli di processo, già proposti in letteratura.
Gli studi condotti nel process mining hanno prodotto tecniche e metodi che si prestano bene alla risoluzione delle problematiche emergenti dai vari domini. Infatti il process mining ha come obbiettivi principali quelli di:

1. supportare la comprensione del processo (*come* e *perché*?);
2. supervisionare o simulare esecuzioni di processo;
3. scoprire e/o raffinare modelli esistenti.

Stabiliti questi presupposti, si orienta la propria ricerca in quest'area, nella quale alcune sfide aperte a problemi noti richiedono ancora soluzioni più appropriate. Una fra tante è quello per la complessità nei processi, per cui diverse metodologie sono state impiegate, a volte con scarsi risultati. Nella mia ricerca propongo l'adozione di approcci dichiarativi, già capaci di gestire modelli e domini complessi [8], per sopperire al problema della complessità e comprensibilità dei modelli. Essi consentono di definire più agevolmente, attraverso vincoli logici, sia i modelli che le condizioni da soddisfare nel completamento dei processi. Inoltre, nell'ottica di applicare le tecnologie in contesti reali, i cui processi esibiscono comportamenti altamente dinamici nel tempo, sorge l'esigenza di sviluppare soluzioni che apprendano incrementalmente i modelli dei processi, garantendo un utilizzo real-time.
La mia ricerca si antepone sui problemi della complessità e della predizione, quest'ultimo ancora abbastanza immaturo in letteratura. Per la complessità ci si affida a paradigmi procedurali ed a strategie di tipo *divide-et-impera*, per apprendere e gestire modelli di workflow semplificati. Attraverso tale semplificazione si potrà disporre di modelli modularizzati, i quali aiuteranno a comprendere meglio la composizione e/o la scomposizione di quei processi troppo complessi. Al fine di garantire modelli predittivi, da applicare laddove il modello di workflow non è noto a priori (ad esempio gli ambienti intelligenti) si vogliono adottare delle metodologie per il task di predizione, focalizzando l'interesse su *activity e process prediction*. Ci si prepone come obbiettivo quello di ricavare i dataset relativi ai contesti di interesse.

4. Stato dell'arte

L'idea principale del process mining è quella di scoprire processi reali ed inferire automaticamente modelli formali che ne riflettano e descrivano il loro comportamento. La scoperta avviene mediante analisi ed estrazione di conoscenza da log di eventi, resi disponibili da diversi sistemi informativi presenti oggigiorno.
I task principali sono i seguenti:

1. **Process Discovery**, inteso come la scoperta (automatica) di processi da log di eventi, estraendo un modello sufficientemente generale (conforme agli esempi osservai), senza che sia necessario dare qualsiasi informazioni a priori.
2. **Conformance Checking**, intesa come il controllo della conformità del modello costruito, rispetto al log di eventi (realtà e astrazione devono essere conformi l'una con l'altra). Fornisce informazioni sull'allineamento dei processi col modello producendo deviazioni o discrepanze, utili a raffinamenti futuri;
3. **Enhancement**: raffinamento o estensione di un modello costruito a priori attraverso nuove istanze di processo.

I **log di eventi** sono il punto di partenza per registrare sequenzialmente gli eventi che caratterizzano un processo, così come esso viene eseguito, riferiti ad attività istantanee e identificabili separatamente, in relazione a particolari casi. I **casi** (istanze) di processo sono raccolti nei log e rappresentati come **trace**, il quale è una mappatura delle reali esecuzioni con le sequenze di **eventi**. Ogni evento è un'unità di informazione riferita a uno e un solo caso di processo in cui si registrano il frammento di lavoro, definito **task** (un task in esecuzione è un'attività), l'istante temporale e il numero di volte in cui occorre e dati sul contesto [7]. Nonostante un trace sia praticamente una sequenza di eventi esso può prevedere flussi di lavoro paralleli o iterativi. Un **workflow model** è la specifica formale di come un insieme di azioni può essere composto per produrre processi validi [12]. Esso cattura 4 tipologie di flusso:

1. **Sequenzialità**, se le dipendenze tra task sono sequenze;
2. **Iteratività**, se le dipendenze formano dei cicli;
3. **Opzionalità**, se la dipendenza per uno o più task è una selezione/opzione;
4. **Concorrenza/Parallelismo**, se le dipendenze esprimono punti di giunzione (join) o punti di divisione (fork) tra task.

Le tecniche del process mining mirano ad inferire modelli che siano:

1. **completi**, che generino tutte le possibili sequenze di eventi osservate;
2. **non ridondanti**, che generino il minor numero di sequenze di eventi mai osservate;
3. **minimali**, che siano possibilmente i più semplici e compatti.

Uno schema è accurato se è completo e non ridondante tipicamente in contrasto con la minimalità. Infatti la semplicità/compattezza di un modello comporta underfitting su casi di processo mai osservati prima (ridondanza). L'obbiettivo è quello di costruire un modello basato su un buon compromesso tra queste proprietà.
Nel process mining classico l'impostazione è quella basata su "solo positivi", differente da quella tipica del machine learning nella quale per un particolare concetto vi è un trainer che fornisce esempi e controesempi. Nei log non è registrato, o meglio, indicato, quello che non dovrebbe esser fatto poiché esso prevede tutti e i soli comportamenti osservati, assunti come le reali, e si presume, valide esecuzioni.
Il process mining si pone su più prospettive [12], ognuna delle quali con propri obiettivi da raggiungere:

1. **process perspective**, rispetto al flusso di controllo (How?) si ricerca una buona caratterizzazione del processo, in funzione di tutte le possibili alternative osservate;
2. **organizational perspective**, rispetto all'organizzazione (Who?) quali risorse sono coinvolte nell'esecuzione del processo= L'obbiettivo è classificare le persone in termini di ruoli e unità di organizzazione definendo le relazioni tra di essi (reti sociali);
3. **case perspective**, (What?) è possibile scoprire colli di bottiglia in processi reali, come quelli industriali, oppure predire le attività successive durante l'esecuzione di qualsiasi istanza di processo (nota e non).

Le varie ricerche in quest'area hanno contribuito allo sviluppo di algoritmi di mining maturi e hanno altresì indirizzato a molte sfide importanti [13,16]. Alcune di quelle aperte, proposte nell'ambito del process mining, riguardano le seguenti problematiche:

- **noise**: i log di eventi potrebbero contenere dati non corretti o non completi, generando problemi in fase di estrazione. Essendo tali dati poco frequenti, potrebbero rappresentare esecuzioni non valide o controesempi (non previsti nel process mining) e quindi generare rumore. Gli algoritmi di mining necessitano per cui di essere robusti rispetto al rumore;
- **duplicate tasks**: tale problema occorre in quei modelli di processo (ad esempio in una rete di Petri) in cui due o più nodi si riferiscono allo stesso task. Se un log prevede lo stesso task più volte, nel learning si deve affrontare il problema di distinguerli;
- **Non-free choice constructs**: accade molto spesso che due o più task condividano lo stesso preset (nelle Reti di Petri) e che la scelta di uno di essi dipenda fortemente da azioni effettuate in precedenza. L'estrazione di costrutti di non-free choice è un problema non banale;
- **Mining loops**: i task in un processo possono occorrere diverse volte durante l'esecuzione (duplicate tasks) e generare loop o cicli. Non banale è il problema di scoprire cicli, laddove ci siano, dettati da salti all'indietro in qualsiasi punto del processo;
- **Different perspectives**: gli eventi del processo possono essere integrati con informazioni aggiuntive al fine di aumentare il potere espressivo di un modello estratto;
- **Delta analisys**: riguarda il confronto di modelli di processo e modelli di riferimento ideali per verificare la conformità (similarità o le diversità) dei comportamenti;
- **Concurrent processes**: esistenza di flussi paralleli all'interno di un processo, attività che concorrono parallelemante;
- **Process re-discovery**: riguarda la selezione di algoritmi di mining che possano riscoprire una classe di modelli di processo da un log completo.

Generalmente un processo può essere modellato come un grafo, dove i nodi rappresentano le attività/tasi o stati, mentre gli archi connettono i nodi al fine di rappresentare il flusso di controllo potenziale. In letteratura sono stati proposti differenti modelli per la rappresentazione dei processi, ognuno dei quali con pro e contro, che verranno sintetizzati di seguito.
Una notazione basilare è rappresentata dai *transition system* (TS) [17], formalmente sono una tripla di insiemi $<S,A,T>$, S stati, A attività e T è il sotto insieme del prodotto cartesiano tra S x A x S. La Ha il vantaggio di essere semplice e lo svantaggio di non essere in grado di gestire adeguatamente la concorrenza, poiché richiede un numero esponenziale di *transition* per ogni esecuzione.
Un formalismo più efficiente è quello delle *Petri Net*, i cui *place* rappresentano condizioni e dipendenze, mentre *transition* le attività, le quali sono attivate mediante il meccanismo *token-based.* Esso ha dato le basi per lo sviluppo delle *workflow net* (WF-net) [18], specializzazione delle *Petri Net*, con le quali si modella la dimensione del flusso di controllo di un processo, specificando il comportamento dinamico di un singolo caso in isolamento. Uno degli algoritmi di mining noti basato su questo formalismo è l'α-algorithms [18]. Le WF-net non permettono *deadlocks* o *live locks* (garantendo la terminazione), non ci sono token pendenti. Sono delle buone candidate per la

specifica, validazione e esplorazione del processo. Tuttavia è in grado di gestire flussi concorrenziali solamente tra coppie di task.
Altri approcci adottano le FSM (Finite State Machine) [13], nelle quali un trace di eventi è intesa come una frase appartenente a qualche linguaggio sconosciuto e tramite il discovery si produce una grammatica nella forma di una macchina a stati finiti (modello del linguaggio). Le FSM catturano 3 strutture basilari del processo (sequenza, selezione e iterazione), tuttavia non considerano il comportamento concorrente.
Le tecniche basate su Hidden Markov Models (HMMs) [15] (generalizzazione degli automi a stati finiti) offrono vantaggi grazie alla loro natura probabilistica, dove le transizioni tra stati e la generazione dell'output hanno una distribuzione di probabilità. Purtroppo uno svantaggio molto rilevante di queste tecniche, ancora, è quello di non essere adatte per modellare i comportamenti concorrenti.
Un approccio distante da quelli su cui si basano le notazioni precedenti è quello del Process Mining Dichiarativo, derivante dalla logica, grazie al quale si descrivono formalmente i modelli specificando vincoli da soddisfare durante l'esecuzione delle attività. Si dimostra capace nella gestione di processi e domini che esibiscono comportamenti molto complessi [8]. In [23] si è sviluppato un algoritmo basato sul dichiarativo chiamato Declarative Process Mining Learner, di contro le tecniche basate sul dichiarativo, come questa, richiedono un setting con esempi positivi e negativi, non proprio del process mining. Una versione incrementale dello stesso approccio è rappresentata da Process Miner [24], il quale è vantaggioso sia dal punto di vista dell'accuratezza che del runtime.
Un framework basato su approcci dichiarativi è WoMan (**Wo**rkflow **Man**agement) trattato in [19] che apprende *from scratch* e *incrementalmente* modelli di processo. Opera nell'Inductive Logic Programming (ILP) facendo uso della First-Order Logic (FOL) come formalismo di rappresentazione, da cui trae vantaggio, e si avvale di tecniche di machine learning (ML) (come InTheLex [20]) per costruire teorie su pre/post condizioni sui task (e non solo) potenziando i workflow nella verifica della loro applicabilità [21]. I *log* accettati da WoMan sono riportati in un formalismo proprio [19] in cui ogni trace è costituito da un timestamp, un evento che indica l'inizio o la fine di un'attività o processo, dal modello di processo a cui si riferisce e dal caso in cui esso occorre, dall'attività con relativo numero progressivo di occorrenza dall'eventuale risorsa (agente) che la esegue. Rispetto ad altri approcci consente l'esplicita espressione del parallelismo tra attività ed evita il bisogno di supporlo statisticamente, potenzialmente errato. Il modello generato è suddiviso in *task* (attività con peso/probabilità) e *transition* (dipendenze tra insiemi di task con peso/probabilità), la sua esplorazione (utile nei task di supervisione o simulazione) avviene mediante un approccio ispirato al meccanismo token-based [22]. Il framework affronta il problema del *noise* in maniera implicita e naturale, col suo modulo di learning, aggiornando le probabilità di *task* e *transition* e ignorando (durante l'esplorazione) quei casi rumorosi, laddove sia necessario. Per la supervisione il framework è in grado di confrontare modelli e log grazie a diverse feature che determinano l'allineamento modello-istanza.
Gli algoritmi di soft computing come le reti neurali imparano a riconoscere pattern di processo nei dati attraverso meccanismi di feedback, mentre gli algoritmi genetici [14] rappresentano i pattern del processo come stringhe di cromosomi per apprendere Reti di Petri. Essi sono altresì in grado affrontare alcune delle sfide sussistenti in letteratura come attività nascoste o costrutti di non-free choice.
Altre tecniche cercano di scovare relazioni tra attività, come il parallelismo, con approcci probabilistici, facendo uso di matrici di causalità tra attività o come in [18] di tabelle di dipendenza/frequenza, supponendo statisticamente l'esistenza di un flusso parallelo. Ad esempio, siano *a* e *b* attività, se risulta frequente nelle istanze di processo che *"a segue b"* e *"b segue a"*, è ammissibile pensare a un parallelismo.
In [7] si è sviluppata una tecnica probabilistica che rileva comportamenti concorrenti nei trace dei log ed inferiscono un modello che li descriva. Tuttavia tale approccio non rispetta una delle

proprietà del process mining, infatti i modelli sono incompleti e non sono in grado di replicare con precisione i comportamenti osservati. In [5] si sono sviluppate 4 metriche (*entropia, causality, periodicity e event type counts*) per la scoperta del comportamento concorrente, combinate fra di loro, che sebbene risultino efficaci tendono ad aumentare il tempo di esecuzione degli algoritmi in maniera proibitiva.
Come riportato in [13], la maggior parte degli studi riguardano il *discovery*, dove si mira a scoprire modelli che meglio descrivano insiemi di istanze, il problema maggiormente affrontato in letteratura è quello pertinente con il *noise*, i cui approcci risolutivi sono nella fattispecie basati su algoritmi non noti. A seguire, le sfide sui cui molti studi sono orientati sono i *mining loops*, affrontati maggiormente sia con algoritmi genetici che con algoritmi non noti; rilevanti sono le tecniche adoperate per tali scopi che ricadono nell'area del data mining e del soft computing. Molto importante è il problema dei *concurrent processes*, che come si è visto molte tecniche non riescono ad affrontare con efficienza e le cui soluzioni non fanno uso di tecniche note o standard.
Sebbene le tecniche adottate per affrontare i vari problemi sono molteplici, non è vero il fatto che ne esista almeno una che possa essere indirizzata verso tutti.

5. Approccio al problema

Come già accennato precedentemente si vuole far uso di approcci dichiarativi per far fronte alla complessità esibita dai processi in particolari domini. Nello specifico i problemi che si vogliono affrontare riguardano:

1. **complex process models** – la complessità dei processi è un problema noto in letteratura ed è stata affrontata gestendo i task duplicati, i comportamenti concorrenti e i loop;
2. **prediction** - riguarda la capacità di un modello di processo nell'essere predittivo;

Si propongono a tal proposito delle soluzioni generali basandosi su una metodologia, sviluppata nel laboratorio LACAM del dipartimento di informatica, che rientra nella sfera del *declarative process mining*, e a supporto della quale è stato prodotto un framework chiamato **WoMan** [19]. Le soluzioni adottate in questa metodologia si sono dimostrate già capaci di gestire la complessità introdotta da concorrenza, loop e task duplicati. Inoltre il framework adotta un proprio approccio per la supervisione durante la quale confronta l'esecuzione di un processo con il relativo modello, generando delle possibili alternative o stati parziali plausibili (ognuna con un proprio grado di allineamento), che è possibile percorrere nel grafo delle dipendenze, su cui il modello è stato costruito. Si vogliono integrare nuove soluzioni che supportino quelle esistenti per garantire la massima efficienza nell'affrontare i problemi sopracitati e che apportino progressi nello stato dell'arte:

1. per il task di **prediction**, facendo fede all'attuale approccio per la supervisione si intendono esplorare ed integrare delle tecniche di *link prediction*. Per sopperire al problema dell' *activity prediction* si intende produrre delle classifiche di link potenziali, tra stato corrente e azione candidata, ordinate per score, con le quali rappresentare la plausibili predizioni della prossima attività da portare a termine. Inoltre, sfruttando sempre l'approccio sopracitato, si vogliono esplorare ed integrare tecniche *distance-based* o *similarity-based*, al fine di classificare istanze di processo non note in uno spazio di modelli di workflow candidati;
1. per i **complex process models**, si vuole sviluppare un modo alternativo e più generale per gestire la complessità dei modelli, partendo dal riconoscimento di *pattern* che rappresentino *sub-workflow* indipendenti, etichettabili come veri e propri processi. La metodologia deve essere in grado di capire se un modello di processo sia complesso, in altri termini se esso possa essere composto/scomposto con/in ulteriori sotto-processi. Si parte dall'estrazione di *sub-workflow* indipendenti esplorando e integrando tecniche di *frequent subgraph mining*, al fine di modularizzare gli schemi dei processi attraverso strategie *divide-et-impera*. Disponendo di tali modelli, semplificati, si vogliono ampliare le soluzioni per la

supervisione e per la predizione al fine di perpetrare l'esplorazione tra i diversi moduli con cui un sotto-processo possa essere conforme.

6. Risultati attesi

La prospettiva è quella di produrre una metodologia da integrare in uno strumento di apprendimento, gestione ed analisi di processi, che sia in grado di affrontare e risolvere le problematiche proposte, portando ad un avanzamento nello stato dell'arte. Risulta ragionevole l'impiego di tali soluzioni in quei domini in cui il process mining si è già dimostrato utile, ma anche in altri non ancora esplorati, al fine di garantire un reale miglioramento. In particolar modo:

- in **contesti aziendali**, i quali sono pervasi da procedure caratterizzate da interazioni complesse tra task differenti e interconnessi. Il process mining in quest'ambito propone una gestione e una definizione formale di tali interazioni, le quali sono fondamentali per il raggiungimento di dati obbiettivi. Disponendo di modelli di processo adatti sarà possibile determinare il successo economico e la produzione delle aziende. Infatti, mentre molte informazioni sui processi derivano da chi li mette in pratica (dipendenti, operai ecc.), i modelli formali di attività sono delineati da manager e supervisori, un gap che spesso causa un disallineamento tra reale e modellato. In aggiunta, produrli manualmente è un compito complesso, costoso e soggetto ad errori. Per questo è possibile adottare tecniche di process mining per apprenderli e raffinarli automaticamente affinché rispettino coerentemente la realtà. Tali strumenti sono di interesse per ottenere modelli astratti da confrontare con la pratica per verificare la correttezza o conformità della realtà in base alle loro politiche;
- in ***Ambient Intelligence*** (AmI), in particolare negli Smart Home Environments (SHE) nei quali è stato dimostrato che i dati raccolti dai vari sensori possono essere collezionati (in log) per scoprire pattern corrispondenti alle attività dei soggetti coinvolti. I modelli appresi possono essere utilizzati per riconoscere tali pattern e rispondere in maniera context-aware ai bisogni degli utenti. Infatti le routine giornaliere possono essere viste come processi da apprendere e modellare mediante i task del process mining. I modelli che si costruiscono possono essere continuamente confrontati con le sotto-sequenze di eventi osservati nei log, al fine di verificare la coerenza tra comportamento osservato e appreso. Se i nuovi eventi sono consistenti con il modello è possibile proporre le prossime azioni più propense ad essere eseguite (*activity prediction*). Grazie alle capacità incrementali è possibile rilevare costantemente ogni aspetto variabile del comportamento umano, aggiornando il relativo modello, se necessario. Attraverso la scoperta di subroutine o sotto-processi a cui una certa sotto-sequenza di eventi può appartenere, si può comprendere, modellare e organizzare le attività di un utente in goal e sotto-goal da predire o classificare;
- applicando opportune astrazioni e analogie a situazioni non propriamente correlate ai processi è possibile applicare quanto fatto a **domini "non convenzionali"**. Infatti, alcuni risultati sperimentali preliminari hanno dimostrato che apprendere modelli di processo ed applicarvi task predittivi o classici, in campi quali linguaggio naturale, match di scacchi etc., può rivelarsi proficuo proseguire nell'esplorazione di strade alternative, basate approcci *process-based*, per risolvere le problematiche relative ad altri contesti.

7. Fasi del progetto

Primo Anno – Al fine di conseguire gli obbiettivi si necessita di una buona base teorica reperibile in letteratura e l'approfondimento di eventuali tecnologie coinvolte, di sistemi esistenti, nonché l'acquisizione delle competenze necessarie ad utilizzarle. Grazie a tale base sarà possibile ricercare, studiare e approfondire metodi per l'integrazione di tecniche e metodi risolutivi ai problemi emergenti nell'ambito del *process mining.* In particolare per la scoperta e la gestione di modelli di

processo complessi si adotteranno tecniche di *graph mining*; per una migliore comprensione e per affrontare la complessità dei processi si adotteranno, inoltre, tecniche di ILP; per supportare i vari task predittivi si analizzeranno varie tecniche di *machine learning* come la *link prediction* o metodi *similarity-based*.

Secondo Anno - A seguito dello studio condotto durante il primo anno, si potrà lavorare sulla progettazione di metodi che utilizzino modelli predittivi arricchiti di preziose teorie sui workflow di processo e che riescano a scovare e costruire automaticamente schemi secondo l'approccio *divide-et-impera*, componendo *subroutine* o sotto-processi. Inoltre, sarà possibile realizzare, implementare ed integrare dei moduli software che prevedano l'uso di esempi di processo adatti al sistema, per valutare le prestazioni con le nuove realizzazioni. Sarà a questo punto necessario individuare i dataset con cui valutare le prestazioni delle nuove realizzazioni.

Terzo Anno – Alla luce dei risultati ottenuti durante il lavoro del secondo anno, sarà necessaria una loro riorganizzazione, per pianificare una sperimentazione più ampia del framework risultante, in modo da avere risultati più solidi su cui basare la stesura del lavoro di tesi. Si intende far in modo che esso rappresenti un connubio dei risultati ottenuti attraverso le varie sperimentazioni, che avranno messo in luce i punti critici su cui si saranno basati tutti i successivi studi e realizzazioni.

8. Valutazione dei risultati

Per la valutazione della qualità o bontà del modello appreso saranno applicate le metriche già definite in letteratura [10,11] e riportate di seguito:

- **Precision.** Un modello preciso non dovrebbe prevedere comportamenti mai osservati, infatti soffrirebbe di *underfitting*. Può essere intesa come la frazione di comportamenti permessi dal modello e che non sono stati mai visti nel training set; meno comportamenti mai visti sono permessi, più preciso sarà il modello;
- **Simplicity**. Un modello che al contempo sia il più semplice e che riesca a replicare quanti più comportamenti osservati è il migliore; va in contrasto con la *precision*. Può essere intesa come la quantità di *task* e *transition* apprese per coprire i casi osservati in un particolare contesto. Essa cresce in proporzione della complessità del dominio;
- **Fitness.** Indica quanto il modello è in grado di replicare il comportamento osservato nei *log*; essa va in contrasto con la *simplicity* e può essere intesa come la porzione di trace in un log che il modello è in grado di replicare;
- **Generalization.** Riguarda l'abilità del modello di riprodurre comportamenti futuri, mai visti prima; è una misura per la confidenza della *precision* e può essere intesa come la frazione di comportamenti previsti nei log che non sono permessi dal modello; più comportamenti sono permessi più generale sarà.

Si vuole, inoltre, valutare la capacità di un sistema di workflow mining e management nella gestione della complessità mediante le seguenti metriche:

1. **Learning time**: tempo necessario ad apprendere un singolo esempio; il sistema dovrebbe essere in grado di mantenere costanti e rapidi i tempi di apprendimento in relazione alla disponibilità di nuovi esempi e alla crescita del modello costruito;
2. **Scalability**: tempo necessario a compiere un singolo task per ogni esempio; il sistema dovrebbe essere in grado di scalare man mano che il modello appreso cresca in rapporto al numero di esempi utilizzati per il training.

Per valutare le performance di un modello, quando esso è impiegato per la *process prediction*, o delle teorie sulle condizioni, ci si avvale delle classiche metriche usate in machine learning (TP = *True Positive*, TN = *True Negative*, FP = *False Positive*, FN = *False Negative*):

1. **Accuracy**: $Acc = (TP + TN) / (TP + TN + FP + FN)$;
2. **Precision**: $\pi = TP / (TP + FP)$;
3. **Recall**: $\rho = TP / (TP + FN)$;

4. **F-Measure**: $F = (2*\pi*\rho) / (\pi+\rho)$;
5. **Area Under the ROC Curve:** sull'asse delle ascisse c'è il False Positive Rate (FPR) mentre su quello delle ordinate il True Positive Rate (TPR). L'area al di sotto della curva ROC permette di confrontare più classificatori;
    a. **FPR** = FP / (FP+TN);
    b. **TPR** = TP/ (TP+FN).

Per quanto riguarda il task della *activity prediction* ci si avvale di misure appositamente definite in [22]:

1. **Recall**: Numero di eventi predetti / (Numero di eventi predetti + Numero di evento non predetti);
2. **Prediction**: Percentuale di predizioni eseguite sugli eventi di un trace;
3. **Ranking**: Posizionamento medio del task predetto nella classifica dei task candidati;
4. **Quality**: Mix dei parametri precedenti appartenente all'interno [0,1].

9. Eventuali referenti esterni al Dipartimento

Durante il periodo della scuola di dottorato si vogliono instaurare rapporti di collaborazione con ricercatori di istituti o Università italiane o estere. Gli eventuali riferimenti esterni sono:

- **Federico Chesani**, del Dipartimento di Informatica dell'Università di Bologna, il quale si è occupato, tra gli altri, di approcci dichiarativi, strategie abduttive o induttive nel *process mining* e *business processes*;
- **Marco Montali**, Faculty of Computer Science, Free University of Bozen-Bolzano, che si occupa di knowledge representation and automated reasoning per la *formal specification, verification, synthesis, planning, monitoring, mining and intelligent management of dynamic systems,* focalizzando l'attenzione particolarmente su *business processes* per catturare i comportamenti di organizzazioni complesse;
- **Gianluigi Greco**, Professore Associato di Computer Science, all'Università della Calabria, il quale si occupa di process mining. Nello specifico, di modellazione di processi ontology-driven e della scoperta di tassonomie su modelli di processo comportamentali;
- **Luigi pontieri**, Ricercatore dell'ADA Lab dell'ICAR, che si occupa di scoprire conoscenza da log di processi e in particolar modo all'estrazione di modelli di workflow, dedicando una discreta attenzione alla predizione;
- **Wil van der Aalst**, Professore Ordinario di Information Systems at the Tecnische Universiteit Eindhoven (TU/e), le cui ricerche rivolgono l'attenzione al *process mining, business process managment, models of concurrency, simulation for analysis, workflow patterns and workflow models systems.*

10. Riferimenti bibliografici

1. HERBST, Joachim; KARAGIANNIS, D. An inductive approach to the acquisition and adaptation of workflow models. In: Proceedings of the IJCAI. 1999. p. 52-57.
2. RASHIDI, Parisa; COOK, Diane J. Adapting to resident preferences in smart environments. In: AAAI Workshop on Preference Handling. 2008. p. 78-84.
3. RASHIDI, Parisa; COOK, Diane J. Keeping the resident in the loop: Adapting the smart home to the user. IEEE Transactions on systems, man, and cybernetics-part A: systems and humans, 2009, 39.5: 949-959.

4. ELLIS, Clarence; KEDDARA, Karim; ROZENBERG, Grzegorz. Dynamic change within workflow systems. In: Proceedings of conference on Organizational computing systems. ACM, 1995. p. 10-21.
5. WEIJTERS, Anton JMM; VAN DER AALST, Wil MP. Rediscovering workflow models from event-based data using little thumb. Integrated Computer-Aided Engineering, 2003, 10.2: 151-162.
6. VAN KASTEREN, Tim; KROSE, Ben. Bayesian activity recognition in residence for elders. 2007.
7. COOK, Jonathan E.; WOLF, Alexander L. Discovering models of software processes from event-based data. ACM Transactions on Software Engineering and Methodology (TOSEM), 1998, 7.3: 215-249.
8. PESIC, Maja; VAN DER AALST, Wil MP. A declarative approach for flexible business processes management. In: International Conference on Business Process Management. Springer Berlin Heidelberg, 2006. p. 169-180.
9. VAN DER AALST, Wil, et al. Process mining manifesto. In: International Conference on Business Process Management. Springer Berlin Heidelberg, 2011. p. 169-194.
10. VAN DER AALST, Wil MP. Mediating between modeled and observed behavior: The quest for the “right” process: keynote. In: Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on. IEEE, 2013. p. 1-12.
11. BUIJS, Joos CAM; VAN DONGEN, Boudewijn F.; VAN DER AALST, Wil MP. On the role of fitness, precision, generalization and simplicity in process discovery. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer Berlin Heidelberg, 2012. p. 305-322.
12. VAN DER AALST, Wil MP. The application of Petri nets to workflow management. Journal of circuits, systems, and computers, 1998, 8.01: 21-66.
13. TIWARI, Ashutosh; TURNER, Chris J.; MAJEED, Basim. A review of business process mining: state-of-the-art and future trends. Business Process Management Journal, 2008, 14.1: 5-22.
14. DE MEDEIROS, A. KA; WEIJTERS, A. JMM; VAN DER AALST, W. MP. Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery, 2007, 14.2: 245-304.
15. DA SILVA, Gil Aires; FERREIRA, Diogo R. Applying hidden Markov models to process mining. Sistemas e Tecnologias de Informação. AISTI/FEUP/UPF, 2009.
16. VAN AALST, W. M. P.; WEIJTERS, A. J. M. M. Process mining: a research agenda, Computers in Industry. vol, 2004, 53: 231-244.
17. VAN DER AALST, Wil MP, et al. Process mining: a two-step approach to balance between underfitting and overfitting. Software & Systems Modeling, 2010, 9.1: 87.
18. VAN DER AALST, Wil; WEIJTERS, Ton; MARUSTER, Laura. Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering, 2004, 16.9: 1128-1142.
19. FERILLI, Stefano. WoMan: logic-based workflow learning and management. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014, 44.6: 744-756.
20. FERILLI, S. A Framework for Incremental Synthesis of Logic Theories: An Application to Document Processing. 2000. PhD Thesis. Ph. D. thesis, Dipartimento di Informatica, Universita di Bari, Bari, Italy.
21. FERILLI, Stefano. Handling Complex Process Models Conditions Using First-Order Horn Clauses. In: International Symposium on Rules and Rule Markup Languages for the Semantic Web. Springer International Publishing, 2016. p. 37-52.
22. FERILLI, Stefano, et al. Predicting Process Behavior in WoMan. In: AI* IA 2016 Advances in Artificial Intelligence. Springer International Publishing, 2016. p. 308-320.

23. LAMMA, Evelina, et al. Applying inductive logic programming to process mining. In: International Conference on Inductive Logic Programming. Springer Berlin Heidelberg, 2007. p. 132-146.
24. JENSEN, Kurt, et al. (ed.). Transactions on Petri Nets and Other Models of Concurrency VII. Springer, 2013.