La surveillance efficace de bout-à-bout pour la gestion des pannes dans les systèmes distribues

June 24, 2014

Abstract

Dans cette thèse, nous présentons notre travail sur la gestion des pannes dans les systèmes distribués, avec comme motivation principale le suivi de fautes et de changements brusques dans de grands systèmes informatiques comme la grille et le cloud. Au lieu de construire une connaissance complète a priori du logiciel et des infrastructures matérielles comme dans les méthodes traditionnelles de détection ou de diagnostic, nous proposons d'utiliser des techniques spécifiques pour effectuer une surveillance de bout en bout dans des systèmes de grande envergure, en laissant les détails inaccessibles des composants impliqués dans une boîte noire. Pour la surveillance de pannes d'un système distribué, nous modélisons tout d'abord cette application basée sur des sondes comme une tâche de prédiction statique de collaboration (CP), et démontrons expérimentalement l'efficacité des méthodes de CP en utilisant une méthode de la max margin matrice factorisation. Nous introduisons en outre l'apprentissage actif dans le cadre de CP et exposons son avantage essentiel dans le traitement de données très déséquilibrées, ce qui est particulièrement utile pour identifier la class de classe de défaut de la minorité. Nous étendons ensuite la surveillance statique de défection au cas séquentiel en proposant la méthode de factorisation séquentielle de matrice (SMF). La SMF prend une séquence de matrices partiellement observées en entrée, et produit des prédictions comportant des informations à la fois sur les fenêtres temporelles actuelle et passé. L'apprentissage actif est également utilisé pour la SMF, de sorte que les données très déséquilibrées peuvent être traitées correctement. En plus des méthodes séquentielles, une action de lissage pris sur la séquence d'estimation s'est avérée être une astuce pratique utile pour améliorer la performance de la prédiction séquentielle. Du fait que l'hypothèse de stationnarité utilisée dans le surveillance statique et séquentielle devient irréaliste en présence de changements brusques, nous proposons un framework en ligne semi-supervisé de détection de changement (SSOCD) qui permette de détecter des changements intentionnels dans les données de séries temporelles. De cette manière, le modèle statique du système peut être recalculé une fois un changement brusque est détecté. Dans SSOCD, un procédé hors ligne non supervisé est proposé pour analyser un échantillon des

séries de données. Les points de changement ainsi détectés sont utilisés pour entraîner un modèle en ligne supervisé, qui fournit une décision en ligne concernant la détection de changement à parti de la séquence de données en entrée. Les méthodes de détection de changements de l'état de l'art sont utilisées pour démontrer l'utilité de ce framework. Tous les travaux présentés sont vérifiés sur des ensembles de données du monde réel. Plus précisément, les expériences de surveillance de panne sont effectuées sur un ensemble de données recueillies auprès de l'infrastructure de grille Biomed faisant partie de l'European Grid Initiative et le framework de détection de changement brusque est vérifié sur un ensemble de données concernant le changement de performance d'un site en ligne ayant un fort trafic.

1 Introduction

Les systèmes ne sont pas fiables aujourd'hui: le système distribué à grand échelle du monde réel est bien établi en infrastructures modernes technologies de l'information, et le caractéristique de telle système est décrit comme la suite "la défaillance d'un ordinateur que vous ne savez pas peut laisse votre propre ordinateur inutilisable".

Dans cette thèse, nous soulignons notre travail sur une aspect de la surveillance des fautes: la découverte des fautes. Plus concrètement, deux modalités du comportement de fautes sont considérés dans notre situation: d'une part, la fonction de la service est en binaire. C'est-à-dire, si une service est disponible ou non disponible. D'autre part, la performance de la service est considérée. Quand une performance quantitative est descente, on dit il y a une faute. Dans le premier cas, les composants d'un grand système distribué sont modélisées comme les entités de la ligne et colonne dans une matrice d'état. Et leur états d'interconnexion sont entrées dans cette matrice. La tâche principale est donc de prédire l'ensemble de la matrice d'état avec un ensemble d'entrée partiellement observé en entrée. Dans le cas de suivi de la performance, l'objectif est de détecter les changements de la performance des services d'une manière en ligne. La performance de différents types décrivant le comportement en temps réel d'un service sont recueillies et analysées en ligne, et nous prenons les décisions pour savoir si il y a un changement et quand le changement a eu lieu à partir de ce flux de données de performance.

L'objectif final de la découverte du défaut est d'améliorer la disponibilité et la fiabilité du systèm. Cela permet aux utilisateurs ou le niveau plus élevé du système de surveillance d'obtenir les informations précises et utiles sur les défauts existants ou possibles. L'approche la plus direct est alors de détection et / ou de diagnostic, où un modèle interne détaillée du système est exploité pour identifier les composants défectueux ou au moins les possibilités de failles. Les motifs élémentaires des défauts peuvent être diagnostiqués grâce à diverses techniques comme l'inférence statistique,

l'analyse de la causalité basée sur le fichier log ou de rediffusion déterministe. Ce diagnostic de défaut peut être considéré comme le processus de reconnaissance de l'explication la plus probable pour les symptômes basés sur certains modèles de causalité et d'effet parmi les propositions d'intérêt dans le domaine du problème.

Bien que ces approches de maximiser le profit des données de surveillance, ils sont confrontés à quelques limites pratiques potentiellement importants. La première est simplement l'évolutivité. De plus, le modèle décent du système disponible peut souvent être irréaliste. En conséquence, ce travail pose le problème de découverte de défaut dans un mode boîte noire: nous avons uniquement les donnés de l'entrée et la sortie.

Dans ce cadre, l'inférence doit s'adresser à deux difficultés spécifiques afin être réaliste. Tout d'abord, il faut prendre en compte les distributions fortement déséquilibrées, comme des défauts sont moins fréquents que le comportement nominal; cela appartient à l'aspect spatial de l'inférence. Deuxièmement, dans le domaine temporel, nous avons une hypothèse que les mesures ne pourraient pas être mise à jour complètement, car l'environnement est très dynamiques.

La même stratégie a été appliquée avec de succès dans différents contextes de traiter les deux distributions déséquilibrées et information bruyant: Apprentissage actif sélectionne des échantillons plus d'information afin d'améliorer au mieux la précision de la prédiction. D'autre part, et toujours en considérant la réalisme, l'apprentissage actif présente les inconvénients de ralentir le processus de découverte de défaut et de le rendre plus compliqué. Un objectif en parallèle de ce travail est donc d'évaluer la contribution de l'ingrédient d'apprentissage actif dans le domaine de faute d'inférence que nous proposons.

La motivation de cette thèse est la gestion de faute dans de grands systèmes comme la grille et le cloud. La grille est considéré en quelque sorte à l'ancienne, donc quelques mots sur leur pertinence pourraient être nécessaires. Les technologies spécifiques qui ont été utilisés pour construire des grilles dans les années 2000 ont bien sûr été remplacés par ceux liés aux cloud. Cependant, le paradigme essentiel de la grille est: des ressources matérielles et en toute sécurité assez fédérateur, logiciels et données de plusieurs fournisseurs indépendants. Ainsi les grilles illustrent à la fois les problèmes physiques de systèmes à grande échelle à travers le monde, et les autres questions et les principaux associés à un système multi-opéré et multi-propriété.

Basé sur les motivations ci-dessus, cette thèse s'attache au système de surveillance de fautes et le changement de performance dans un système à grand échelle avec l'approche de l'apprentissage automatique. Les contributions principales sont:

1. Pour la surveillance de fautes de composants interconnectées dans un

système distribué, nous le considérons comme une application de la prédiction collaborative. Cette stratégie profite de la méthode de la factorisation de la matrice en maximisant le marge qui montre l'efficacité. En plus, nous introduisons l'apprentissage actif dans la méthode de la prédiction collaborative et cela signifie une avantage critique dans la surveillance de faute. [2, 1].

2. La deuxième contribution concerne l'extension de la surveillance de faute statique dans le cas séquentiel. La méthode de la factorisation de matrice séquentiel (SMF) que nous proposons, prend une séquence partiel de matrice observée comme l'entrée, et une prédiction informative générée en temps courant et passé. L'extension de SMF avec l'apprentissage actif (SMFA) est proposée dans le cas séquentiel, qui est adapté pour les données déséquilibres. En plus, un lissage sur la séquence d'estimation de chaque algorithme a montré un apporte de meilleures performances de la prédiction.
3. La troisième contribution concerne la détection de changement dans les données d'une séquence de temps. Un formalisme de la détection de changement semi-supervisé est proposé. Les étiquettes préférées sont appris hors ligne par un algorithme de segmentation, et le modèle de la détection est appris en ligne d'une manière en ligne. Ensuite, ce modèle est utilisé pour la détection de changement en ligne.
4. La dernière contribution n'est pas le moins importante. Nous appliquons les méthodes proposées aux données dans le monde réel. Plus précisément, pour la surveillance de fautes, la méthode collaborative est testé avec les données de infrastructure de Biomed grille dans le cadre de European Grid Initiative, i.e., la prédiction de la disponibilité fonctionnelle entre les éléments de calcul (CE) et des éléments de mémorisation (SE). Pour la performance de la surveillance, les données sont recueillis dans un site avec une grande quantité de l'utilisateurs de trafic, i.e., la découverture des modèles de changement de la performance d'un site.

2 Surveillance de fautes collaborative

Cette partie du manuscrit présente l'étape de la construction de connaissances, une boucle de la surveillance et la formulation de l'inférence de fautes. Cet ensemble de tâches est considéré commme un problème de la Prédiction Collaborative (PC). Notre but est de surligner la direction d'appliquer le formalisme PC dans ce nouveau domaine.

2.1 Inférence de faute par la Prédiction Collaborative

Le problématique de la sélection de serveur est lié à l'inférence de faute de bout-à-bout: nous nous intéressons à la performance de la capacité pour

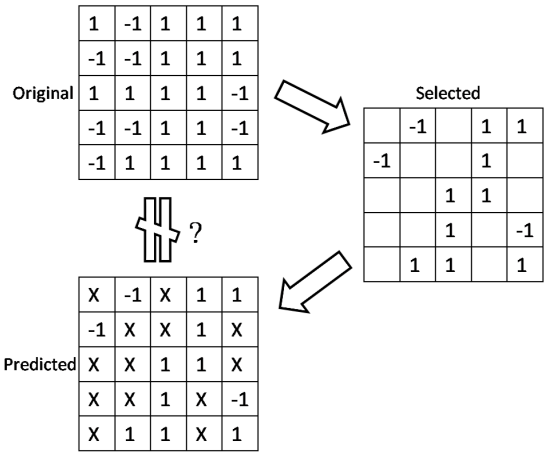

Figure 1: Illustration of matrix recovery

fournir une demande. Ensuite, l'inférence de faute de bout-à-bout est formalisée comme un problème de la complétion de matrice également, où chaque entrée de la matrice représente l'état de bout en bout fonctionnalité de service entre le fournisseur de service correspondant et demandeur. Dans la tâche de la complétion de matrice, la infrastructure globale est traitée comme une boîte noire. Nous mettons en exergue les deux point (l'entrée et la sortie) sur l'inférence de faute sans la connaissance à priori.

Dans la plupart de cette section, une hypothèse de base est *statisme*. C'est-à-dire, la partie physique qui générer des fautes ne change pas au fil du temps. Et puis, une tâche d'inférence est formalisée comme un problème de *complétion de matrice*: Si X est une matrice (creuse) observée, le but d'un problème PC est de retrouver une matrice complète et à valeurs réelles Y, qui est une approximation de X de la même taille, i.e., ce qui permet de minimiser la mesure de la contradiction entre Y et X. Lorsque Y est égal à X sur les entrées connues, le problème est appelé *complétion exact*, et sinon *approximation*.

Pour l'inférence de fautes, X est une sonde choisie et Y est la matrice prédite. Une illustration de la matrice *choisi* et *prédit* est donnée dans la Fig. 1.

La matrice inconnue Y sera supposé d'être déterministe; puis, les données disponibles pour la prédiction peuvent être considérés comme un échantillon des éléments de cette matrice inconnue déterministe complète. La distribution d'échantillonnage est défini par Le processus de sélection de sonde.

2.2 But et approches

2.2.1 Objectives

Suite à la description ci-dessus, le but est de minimiser le nombre de sondes, pour une prédiction en qualité. Il y a deux problématique distinct pour la minimisation de nombre de sondes: la sélection intelligente de sonde, et une algorithme efficace pour la complétion de la matrice. Nous considérons trois approches pour la sélection de sondes.

Algorithm 1: Generic active probing algorithm

input : Initial partially observed binary(-1/+1) matrix M_0, threshold λ, max # of new samples N, active-sampling heuristic h

output : Full binary-valued matrix M^{T_i} predicting unobserved entries of M_0

initialize: Initialize the vars

$S(T_0) = S(M_0)$ /*currently observed entries set*/ ;

$i = 0$ /*current iteration times*/ ;

$n = 0$ /*current number of new samples*/ ;

while $(n < N)$ **do**

  $M^{T_i} = StandardMC(S(T_i))$ /*Prediction based on observed entries via standard matrix completion (MC) procedure*/ ;

  $S'(T_i) = ActiveSampling(M^{T_i}, h, \lambda)$ /*Actively choose the next set of new samples and query their labels*/ ;

  $S(T_{i+1}) = S(T_i) \cup S'(T_i)$;

  $n = n + \#S'(T_i)$;

  $i = i + 1$;

- **Uniforme-Statique.** Les sondes sont choisis uniformément au hasard parmi tous les (CE, SE) paires. Dans ce contexte, le choix de la sonde et la prédiction sont complètement indépendantes: l'étape de prédiction n'a aucune influence sur le choix des sondes. Ce ne serait pas réaliste dans les systèmes de recommandation (les utilisateurs ne sélectionnent pas uniformément les produits qu'elles notent parmi tous les projets), mais peut être pleinement mis en œuvre dans la sélection de la sonde. En outre, pour la tâche de prédiction ultérieure, échantillonnage uniforme fournit des limites théoriques sur l'erreur de généralisation MMMF.
- **Sondage actif.** L'approche d'apprentissage actif général est instanciée par la sondage actif: l'apprentissage est construit progressivement en interrogeant la source d'information pour exemples étiquetés. Ce processus est illustré dans Algorithm 1: une matrice prédite est donnée

par la complétion de matrice standard (MC) basé sur d'échantillons pré-sélectionnés (étape 5). Ensuite des heuristiques sont utilisées pour filtrer la prochaine sous-ensemble d'échantillons, qui sont étiquetés en exécutant effectivement les sondes et l'observation de leur résultat (étape 6). A la fin de l'itération de ce programme, une prédiction finale est calculée. Dans notre contexte, la méthode CP pour MC a une impact sur la sélection de sondes. En plus, le min-marge heuristique [5] est utilisé pour sélectionner des sondes supplémentaires.

- **Récompense différent.** Dans les deux méthodes précédentes, la même pénalité est associée à tous les types de mauvaises prédictions. On pourrait faire valoir que un faux négatif (prédire le succès tandis que le résultat réel est un échec) est plus nocif que d'un faux positif (prédire l'échec tandis que le résultat réel est un succès), parce-que la nature fédéré des ressources de calcul offre de multiples options pour les utilisateurs. Coûts asymétriques (dans les deux sens) se posent dans de nombreux autres contextes, par exemple, des tests médicaux [3], et peut être intégrée dans l'étape d'apprentissage de base, comme indiqué dans la section suivante.

2.2.2 Max margin matrice factorisation

L'approche max margin matrice factorisation (MMMF) [4] exploite la même approche que dans la récupération exacte pour récupération approximative. Au lieu de prendre une approximation de faible rang (e.g., SVD), MMMF minimise l'écart de la norme de trace de la matrice estimée Y entre l'estimation et l'observation sous la contrainte de non (dur-marge), ou faible (soft-marge). Cette formule est convexe, alors une solution optimal globale est garantie. Soit S l'ensemble de entré connu dans X, deux fonctions objectives sont considérées.

- Dur-marge: minimise $\|Y\|_\Sigma$ sous les contraintes

$$Y_{ij}X_{ij} \geq 1 \text{ for all } ij \in S;$$

- Faible-marge: minimise

$$||Y||_\Sigma + C \sum_{ij \in S} \max(0, 1 - Y_{ij}X_{ij}). \tag{1}$$

Comme la sortie de la procédure de minimisation est une matrice à valeurs réelles, un seuil de décision (par exemple, +1 pour les valeurs positives, -1 pour les valeurs négatives) est sélectionné pour la matrice binaire prédite finale.

3 Surveillance de faute séquentielle

L'objectif global de contrôle séquentiel de fautes est exactement la même que dans la section 2 : La prédiction de comportement de fautes de l'information de la sonde limitée. La différence est de prendre en compte le fait que le système évolue dynamiquement à différentes échelles de temps.

3.1 Motivation

Dans la Section 2, la méthode statique est décrit pour manipuler la surveillance de fautes dans un système distribué. L'entrée de cette méthode est un seul matrice statique sans information passée.

Il y a deux limites de cette méthode. D'abord, c'est plus approprié pour obtenir un instantané du système à temps court; par contre, suite à la variabilité des états du système, il est limite pour une tâche à long terme. En plus, fautes transitoires sont systématiquement observées: transitoires sont les fautes qui varient à haute fréquence et devraient être considérés comme du bruit; les praticiens n'ont pas d'explication claire, et ils pourraient aussi bien être des failles dans le logiciel de surveillance lui-même. Bien sûr, le problème est de les dissocier de fautes réels, mais de courte durée.

Un autre motivation est d'explorer la méthode sans l'apprentissage actif. Dans la section précédente, l'apprentissage actif est une prédiction efficace dans le cas "curated". D'autre part, l'apprentissage actif n'est pas pratique: il est nécessaire d'utiliser une boucle de rétroaction et des logiciels plus compliqué que le réglage passive. A l'échelle de la grille, une source de complexité inutile doit être éliminé. Ainsi, nous explorons l'hypothèse que le passé fournit suffisamment d'informations pour être équivalent à celui obtenu en interrogeant de façon sélective le présent.

3.2 Description de problème

Tout d'abord, nous présentons les formulations et les notions utilisées dans cette section.

- $X_t \in B^{M \times W}$ désigne la matrice partiellement observée au moment t.
- $\hat{Y}_t \in R^{M \times W}$ est le résultat de l'algorithme de prédiction.
- $Y_t \in B^{M \times W}$ est la version binaire de $\hat{Y}_t$ prenant un seuil binaire ρ.

3.3 Factorisation de matrice séquentielle

3.3.1 Algorithme SMF

Il y a deux sorte d'informations dans la surveillance séquentielle: l'information spatiale et l'information temporelle. L'information spatiale peut être exploitée par la méthode de la prédiction collaborative comme MMMF, En

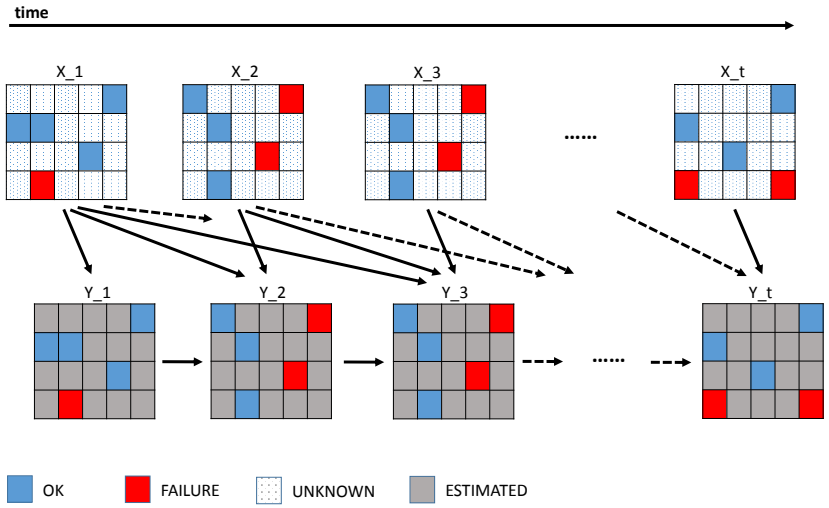

Figure 2: An illustrative example of sequential fault prediction problem

plus, l'information temporelle concerne le caractéristique d'évolution dans le temps. Cela permet d'ajouter des opportunités supplémentaires afin d'améliorer la performance de prédiction.

Dans cette section, nous proposons un algorithme, la factorisation de matrice séquentielle (SMF), afin d'utiliser l'information spatiale et temporelle. Cela permet d'exploiter les états en terme long et court. Dans la partie suivante, S_u, S_c et S_r sont l'ensemble d'indices dans la matrice X, pour représente l'ensemble le plus incertain, le plus confident, et les échantillon au hasard. Nous rappelons la fonction objective de MMMF est:

$$\arg\min_{Y} ||Y_t||_\Sigma + C\mathcal{L}_h(Y_t(S_r), X_t(S_r)), \tag{2}$$

C représente la coefficient de la régularisation, S_r est un ensemble d'échantillon au hasard dans X_t, et $\mathcal{L}_h(Y_t(S_r), X_t(S_r))$ est hinge loss entre Y et X défini comme:

$$\mathcal{L}_h(Y_t(S_r), X_t(S_r)) = \sum_{ia \in S_r} max(0, 1 - Y_{ia}X_{ia}). \tag{3}$$

La fonction objective (Enq. 2) est composée en deux termes. Le premier est le norme de la trace de la matrice estimée Y_t et la deuxième terme est l'écart entre l'estimation et l'observation. Ensuite, nous allons développer

la fonction objective de SMF en ajoutant l'information *la plus incertaine* et *la plus confident* à l'équation 2 progressivement.

Dans un premier temps, nous considérons l'information le plus incertain. Dans le cas séquentiel, l'ensemble de la prédiction le plus incertain S_u (entrées avec petite marge à l'hyper-plan de la classification) est capable d'$\hat{}$etre dérivé de Y_{t-1} et leur étiquettes au moment t peut être interrogée à partir du système. Donc la vérité de ces prévisions plus incertaines de Y_{t-1} sont disponibles dans l'échantillon X_t. On la note $X_t(S_u)$.

La deuxième information est les prédictions *les plus confiantes* cachés dans l'estimation de l'histoire. Au lieu d'échantillonner leurs vrai étiquettes au moment t pour S_c, les prédictions précédentes peut être utilisées pour la prochaine exécution. Plus précisément, dans SMF nous choisissons les prédiction les plus confiants de Y_{t-1} et supposons que leurs états demeurent inchangées au temps t avec un niveau de cohérence γ. γ est calculé en termes de différence entre Y_{t-1} et X_t, i.e., la différence entre l'estimation précédente et l'observation au courant. Les critères de la classification typique comme la sensitivité (TPR) ou FSCORE peut être utilisé pour la mesure de cohérence. L'ensemble observé dans X est l'union de la prédiction les plus incertaines S_u et l'ensemble d'échantillon au hasard S_r, i.e., $X(S_u \cup S_r)$. Le niveau de la cohérence γ est évalué entre $X_t(S_u \cup S_r)$ et $Y_{t-1}(S_u \cup S_r)$ comme la suite:

$$\gamma = TPR(Y_{t-1}(S_u \cup S_r), X_t(S_u \cup S_r)), \tag{4}$$

$TPR(A, B)$ est une fonction pour calculer la sensitivité (taux de vrais positifs) de A en fonction de l'ensemble de l'ensemble vérité B. Dans la prédiction, γ est utilisé comme un ratio de coût adaptatif qui ajuste le poids (pénalité) de l'information heuristique dans la fonction objective (similaire au coefficient C dans Eqn. 2).

En plus de l'ensemble le plus incertain S_u et l'ensemble le plus confiant S_c, nous introduisons également un ensemble aléatoire S_r dans la fonction objective de SMF. Il est pour éviter la sur-montage dans les informations d'historique, où le changement brusque entre la dernière estimation et l'observation en cours peut se produire. Pour résumer, la SMF est décrite comme la suivante:

$$\arg\min_{Y} ||Y_t||_\Sigma + C\mathcal{L}_h(Y_t(S) - X_t(S)) + C\gamma\mathcal{L}_h(Y_t(S_c) - Y_{t-1}(S_c)), \tag{5}$$

$S = S_u \cup S_r$ est l'ensemble d'échantillon au moment t par interrogation d'étiquettes et S_c est la prédiction le plus confiant hérité de $t-1$. La différence entre Eqn. 2 and 5 est présentée par la sélection de S_u et la présence de S_c. Eqn. 5 est convexe comme Eqn. 2, et la méthode dans Section 2 sert à retrouver la minimum globale directement. Le pseudo code de SMF est présenté dans Algorithm 2.

Algorithm 2: Sequential Matrix Factorization (SMF)

Input: $\mathbf{\hat{Y}_{t-1}}$, last prediction;
$\mathbf{N_u}$, number of most uncertain samples from Y_{t-1};
$\mathbf{N_c}$, number of most confident samples from Y_{t-1};
$\mathbf{N_r}$, number of random samples;
$\mathbf{C}$, slack penalty.

Output: Full real-valued matrix $\mathbf{\hat{Y}_t}$

Initialize: $Init\ h_1, h_2, h_3$, /*Initialize the most uncertain, most confident and random sampling heuristic, respectively*/;

1 $S_u \leftarrow Sample(h_1, N_u, \hat{Y}_{t-1})$ /*select N_u most uncertain sample indexes from $\hat{Y}_{t-1}$*/;
2 $S_r \leftarrow Sample(h_2, N_r)$, /*select N_r random sample indexes*/;
3 $S \leftarrow S_u \cup S_r$;
4 $X_t(S) \leftarrow QueryLabels(S)$, /*query the true label for entries in S*/ ;
5 $\gamma \leftarrow TPR(X_t(S), Y_{t-1}(S))$ /*given $X_t(S)$ (true labels for entries in S), compute the sensitivity of $Y_{t-1}(S)$*/;
6 $S_c \leftarrow Sample(h_3, N_c, Y_{t-1})$, /*select N_c most confident samples from Y_{t-1}*/;
7 $\hat{Y}_t \leftarrow \arg\min_Y ||\hat{Y}_t||_\Sigma + C\mathcal{L}_h(\hat{Y}_t(S) - X_t(S)) + C\gamma\mathcal{L}_h(\hat{Y}_t(S_c) - \hat{Y}_{t-1}(S_c))$ /*find an estimation that minimizes the objective function*/;
8 **return** $\hat{Y}_t$

3.3.2 Factorisation de matrice séquentielle avec l'échantillonnage actif

Une autre approche intuitive est de sélectionner l'échantillon activement et itérative-ment dans X_t afin d'améliorer la performance de la prédiction. L'information heuristique comme la prédiction la plus incertaine et la plus confiante est dans Y_t^1 ou les estimation suivante $Y_t^i, i = 2, 3, ...$ Cela permet de bénéficier d'estimation en exploitant proprement.

Dans cette section, nous proposons la factorisation de matrice séquentielle avec l'apprentissage active basée sur SMF itérative. Tous les étapes sont illustrées dans Algorithm 3. Pour la simplicité, la matrice estimée au moment t de $i$ème itération est décrite comme Y_t^i. Au début, nous appliquons SMF pour une estimation initiale Y_t^0 de Y_{t-1} (ligne 4). Ensuite, une estimation itérative est employé à la séquence de prédiction $Y_t^i, i = 1, 2, ...$ jusqu'a le maximum de l'échantillon (ligne 5 à 9). Dans la sélection d'échantillon actif, les prédictions les plus incertaines et les plus confiantes sont choisies de l'estimation précédente par l'algorithme SMF.

Algorithm 3: Sequential Matrix Factorization with Active sampling (SMFA)

Input: N, max # of new samples;
$\hat{Y}_{t-1}$, last prediction;
P_0, initial sample rate for the 1st prediction;
P_a, active sample rate at each iteration;
ρ, ratio of random samples and most uncertain samples for P_a;
C, slack penalty.

Output: Full real-valued matrix $\hat{Y}_t$

initialize: $Init(N_c)$ /*Initialize the number of most confident samples to select in each iteration*/;

$i = 0$ /*current iteration index*/ ;

$n = N \times P_0$ /*current number of new samples*/ ;

$[N_u, N_r] \leftarrow getSampleSize(n, \rho)$ /*Get random and most uncertain sample size for the initial prediction*/;

$\hat{Y}_t^i \leftarrow SMF(\hat{Y}_{t-1}, N_u, N_c, N_r, C)$;

while $(n < N)$ **do**

  $[N_u, N_r] \leftarrow getSampleSize(N \times P_a, \rho)$ /*Get random and most uncertain sample size according to ρ and P_a*/;

  $\hat{Y}_t^{i+1} \leftarrow SMF(\hat{Y}_t^i, N_u, N_c, N_r, C)$;

  $n = n + N_u + N_c + N_r$;

  $i = i + 1$;

$\hat{Y}_t = \hat{Y}_t^i$;

return $\hat{Y}_t$

4 Détection de changement séquentielle

4.1 Introduction

Les sections précédentes font deux different hypothèses sur le comportement de donnée temporel. Dans Section 2, l'information temporelle est inutile (hypothèse *statique*), parce-que tous les données dans une période donnée ont été regroupées en une seule matrice observée. Dans Section 3, l'hypothèse implicite était stationnaire. C'est-à-dire l'information passée est pertinente pour la future. Ensuite, tous les informations passées sont intégrées dans le modèle de comportement (possible faible). Une autre approche est de considérer que des ruptures se produisent, et que le modèle devrait être reconstruit à ces points de changement. Parce que la non-stationnarité est de plus en plus reconnu comme un phénomène omniprésent, la détection de point de changement a pris des recherches approfondies dans divers domaines et les domaines d'application.

En conséquence, nous considérons le problème de *la détection de point de changements*. Cette section ne propose pas d'ajouter un nouvel algorithme pour l'énorme corps de méthodes de détection de changement existants, mais pour définir et évaluer un nouveau cadre, le *la détection des changements semi-supervisé en ligne* (SSOCD). SSOCD intègre de la détection des changements de point en ligne et hors ligne et les exploite efficacement.

4.2 Formalisme de la détection de changement semi-supervisée en ligne, SSOCD

Notre contribution dans cette section est la proposition d'un formalisme de la détection de changement en ligne de la manière semi-supervisé, SSOCD. Par rapport à notre connaissance, c'est la première essaie pour combler la segmentation non supervisée hors ligne et les méthode de la détection de changement supervisée en ligne, tels que des changements peuvent être détectés en ligne. Dans Fig. 3, le formalisme SSOCD est divisé en deux étape: l'étape hors ligne et l'étape en ligne. Dans l'étape hors ligne, la segmentation non supervisée est utilisée pour étiqueter les points de changement dans la séquence d'apprentissage. Ensuite, ces données étiquetés sont appris pour construire le modèle de la détection de changement d'une manière supervisée. Avec cette modèle, les données sont précédés séquentiels et les changements cachés dans le flux de données qui sont capturés par les méthodes habituellement hors ligne sont détectés d'une manière en ligne. Algorithm 4 présente le pseudo code de ce formalisme.

4.3 Résumé

Dans cette section, nous nous intéressons le problème de la détection des changements en ligne. Nous proposons notre méthode de combler des méthodes

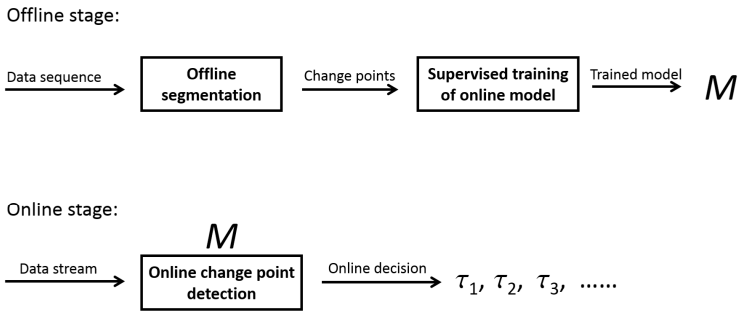

Figure 3: Supervised change point detection framework

Algorithm 4: Semi-Supervised Online Change Detection, SSOCD

Input: X_{train}, train set; ;
X_{test}, sequentially arriving test data; ;
ρ, significance threshold. ;

Output: M_{online}, online change detection model learned from offline methods;
τ, detected change points;

Offline Stage:;
$\tau_{offline} \leftarrow$ stand_offline_segmentation(X_{train}) //segment the training data offline;
$\tau^{'}_{offline} \leftarrow$ significance_test(X_{train}, $\tau_{offline}$) //test the significance of offline result;
$M_{online} \leftarrow$ supervised_train(X_{train}, $\tau^{'}_{offline}$, ρ) //train the online CPD model;
Online Stage:;
$\tau \leftarrow$ stand_online_CPD(M_{online}, X_{test}) //detect change point online;
return τ;

hors ligne et en ligne dans le cadre de SSOCD, dans lequel les approches hors ligne non supervisé sont utilisés pour étiqueter les points de changement afin d'apprendre un modèle bayésien d'une manière supervisé en ligne. Ce modèle est alors engagé dans la détection des changements en ligne.

5 Conclusions

Fiabilité des grands réseaux et des nuages à grande échelle est toujours la préoccupation principale à la fois de l'administration du système et de l'utilisateur. Au lieu de construire une connaissance a priori complète du logiciel et des infrastructures matérielles dans les plupart de méthode pour la détection et la diagnostique, nous proposons d'utiliser les techniques appropriées pour effectuer une surveillance de fautes de bout-en-bout pour les systèmes à grande échelle, en laissant les détails de composants impliqués dans une boîte noire. Une série d'hypothèses concernant le comportement temporel des données sont: 1)*statique*, 2)*stationnaire*, 3)*non stationnaire*, chaque hypothèse correspond aux travaux dans la section 2, 3 et 4, respectivement. Ensuite, nous résumons les problèmes principaux abordés dans les sections précédentes dans l'ordre.

- **Statique** Section 2 considère le comportement temporel des données sous la forme d'un facteur non pertinent pendant une période donnée, et effectue la tâche de prédiction sur une matrice d'observation repliée. Plus précisément, la prédiction de collaboration est utilisé comme une stratégie prometteuse évolutive et pour extraire des informations cachées dans les données de surveillance avec une intrusion limitée au système de cible. Efficacité d'une combinaison de prévision de collaboration et d'apprentissage actif a été démontrée sur un vaste ensemble de données recueillies à partir d'une grille de production (EGI). Fondamentalement, deux questions clés dans le problème de prédiction de défaut sont minutieusement explorées avec une stratégie active basée sur min marge prédiction heuristique: le déséquilibre des exemples positifs et négatifs (dans les systèmes réels défauts sont toujours le groupe minoritaire), et les fautes transitoires. Le modèle de prévision interne d'Active palpage est mis à jour avec les acquisitions d'adaptation de nouvelles connaissances, de telle sorte que l'information cachée des données de surveillance est découverte de manière itérative et progressive.
- **Stationnaire** Section 3 suppose un modèle stationnaire implicite sur le comportement temporel des données, à savoir l'information du passé peut bénéficier la prédiction pour l'avenir. Plus précisément, les corrélations entre les observations successives de données consécutifs sont explorées en utilisant la matrice séquentielle factorisation de la méthode (SMF). En outre, l'apprentissage actif est également utilisé en combinaison

du SMF (SMFA) dans le but d'alléger le problème de déséquilibre de données et le problème de fautes transitoires. Par la comparaison de SMF/SMFA avec plusieurs méthodes de base comme l'EWMA, DEP, FMA et TENSOR sur un vaste de données séquentielle, nous avons exposé les avantages de SMF/SMFA.

- **Non-stationnaire** Dans le but de détecter des changements brusques dans une séquence d'observation de la surveillance de bout-en-bout, Section 4 donne une hypothèse d'un modèle non-stationnaire sur le comportement temporel des données. Un formalisme semi-supervisé de la détection des changements en ligne, c'est à dire la SSOCD, est proposé pour combler les approches non supervisés et supervisés. Plus précisément, dans SSOCD les approches de segmentation non supervisé hors ligne sont utilisés pour la génération d'étiquettes de point de changement pour construire un modèle bayésien en ligne, et le modèle formé est alors engagé dans la détection des changements en ligne. De cette manière, deux types de changement dans la détection supervisée en ligne sont: le manque d'étiquette de points de changement et les fautes de la détection de changement.

References

[1] Dawei Feng, Cécile Germain, and Tristan Glatard. Efficient distributed monitoring with active collaborative prediction. *Future Generation Computer Systems*, 29(8):2272–2283, 2013.

[2] Dawei Feng, Cecile Germain-Renaud, and Tristan Glatard. Distributed monitoring with collaborative prediction. In *12th Int. Symp. On Cluster, Cloud and Grid Computing*, pages 376–383, 2012.

[3] Katharina Morik, Peter Brockhausen, and Thorsten Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *16th Int. Conf. on Machine Learning*, pages 268–277, 1999.

[4] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336, 2005.

[5] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2, March 2002.