Décision de finales d'échecs

Mathis Caristan & Alexandre Fernandez

UPMC

28 Mars 2017

Table des matières

- **Idée :** Paralléliser la boucle `for` dans `evaluate`
- Profondeur max : 2 $\longrightarrow$ ~30 noeuds
- Deux choix d'implémentations possibles

  a°) `parallel for` (plus simple)

  b°) `task` (plus efficace)

```
#pragma omp parallel firstprivate(T,result)
#pragma omp single
evaluate(T, result, 0);
```

```
if(prof < OMP_MAX_PROF) {
#pragma omp parallel for
    for (int i=0 ; i<n_moves ; i++) {
        fct_for(T, moves[i], &child[i], &child_result[i], prof, result);
    }
}
// Sinon appel a fct_for sans #pragma omp parallel for

void fct_for(tree_t *T, move_t m, tree_t *child, result_t *child_res,
        int prof, result_t *cur_res) {
    play_move(T, m, child);

    evaluate(child, child_res, prof);

    int child_score = -child_res->score;

#pragma omp critical(CHILD)
    { // BLOCK OMP : critical
        if (child_score > cur_res->score) {
            cur_res->score = child_score;
            cur_res->best_move = m;
            cur_res->pv_length = child_res->pv_length+1;
            for (int j=0 ; j<child_res->pv_length ; j++)
                cur_res->PV[j+1] = child_res->PV[j];
            cur_res->PV[0] = m;
        }
    } // BLOCK OMP : critical
}
```

- Bloc `critical` pour protéger les accès concurents à cur_res
- Factorisation du code dans la fonction `fct_for`

- `play_move` et `evaluate` sont lancés dans des tâches
- Barrière de synchornisation à la fin du `for`
- Découpage de la boucle en deux zones, dont une seule est critique

```
if(prof < OMP_MAX_PROF) {
    int child_score[n_moves];
    tree_t child[n_moves];
    result_t child_result[n_moves];
    for (int i=0 ; i<n_moves ; i++) {

#pragma omp task firstprivate(i) shared(child_result, child_score, child)
        {
            play_move(T, moves[i], &child[i]);

            evaluate(&child[i], &child_result[i], prof+1);

            child_score[i] = -child_result[i].score;
        }
    }
#pragma omp taskwait
    int max = 0;
    for(int i=0; i<n_moves; i++) {
        if(child_score[i] > child_score[max]) max = i;
    }
    if (child_score[max] > result->score) {
        result->score = child_score[max];
        result->best_move = moves[max];
        result->pv_length = child_result[max].pv_length + 1;
        for(int j = 0; j < child_result[max].pv_length; j++)
            result->PV[j+1] = child_result[max].PV[j];
        result->PV[0] = moves[max];
    }
}
```

- **Idée :** Atteindre une profondeur suffisante pour avoir une multiplicité de tâches intéressante.
- Pré-calcul (séquentiel) consistant à un parcours en largeur de l'arbre.
- Équilibrage de charge dynamique
- Réglages nécessaires selon les conditions d'utilisation

Simulation Monte-Carlo et données

- Trois niveaux dans la simulation : **Le niveau partonique, le niveau vrai (ou niveau particules, niveau hadronique) et le niveau reconstruit**.
- Les données ne donnent accès qu'au niveau reconstruit.
- La simulation est utilisée dans la mesure de section efficace inclusive pour évaluer l'efficacité de la sélection.

Sections efficaces

$$\sigma_{incl}(pp \to t\bar{t}) = \frac{N_{signal}^{hadronique}}{\epsilon_{t\bar{t}}^{incl} \cdot \mathcal{L} \cdot BR^{hadronique}}$$

Coupure	Variable
Préselection	$\lvert\eta\rvert < 2.4$ & $p_T > 25 GeV$
Nombre de jets	$n_{jet} \geqslant 6$
Isolation des jets	$\Delta R(jet_i, jet_j) > 0.6$
Impulsion transverse du 5ème jet	$p_T^{jet5} > 65 GeV$
Étiquetage des jets	$n_{bjet} = 2$
Isolation des jets étiquetés	$\Delta R(b_1, b_2) > 1$
Impulsion transverse du 6ème jet	$p_T^{jet6} > 45 GeV$
Distance boson W - jet de B	$\star$

L'étiquetage au niveau vrai correspond à une association avec un quark b au niveau partonique par proximité spatiale. Au niveau reconstruit, c'est un algorithme qui utilise les informations des détecteurs à pixels et du trajectographe.

- $\epsilon_{t\bar{t}}^{incl}$ déterminée grâce à la simulation. La section efficace inclusive dépend du modèle.
- **On cherche à *minimiser* cette dépendance dans la section efficace fiducielle.**
- Les coupures doivent être adaptées et optimisées pour être appliquées au niveau vrai.

Détermination des bonnes combinaisons de jets par une méthode de χ^2

La cinématique du processus permet de déterminer
la bonne combinaison de jet pour chaque événement.

$$\chi^2 = \frac{(m_{b_1 j_1 j_2} - m_t)^2}{\sigma_t^2} + \frac{(m_{b_2 j_3 j_4} - m_t)^2}{\sigma_t^2} + \frac{(m_{j_1 j_2} - m_W)^2}{\sigma_W^2} + \frac{(m_{j_3 j_4} - m_W)^2}{\sigma_W^2}$$

m_t = paramètre libre.
m_W, σ_W, σ_t, sont déterminés à partir de distributions de masse au niveau hadronique de la simulation.

Fiabilité et amélioration du χ^2

La méthode de χ^2 peut être améliorée en utilisant l'information sur l'étiquetage des jets B.

Écart entre le minimum, et le second minimum du χ^2.

Écart entre le minimum de χ^2, et le χ^2 de la bonne combinaison.

L'utilisation de l'étiquetage des jets donne de meilleurs résultats.

- $\star$ Les jets de B sont choisis parmi les jets étiquetés. (Au niveau vrai)
- $\star$ Les autres jets sont choisis parmi les jets non étiquetés.

La coupure sur $\Delta R(b, W)$

La coupure ne garde que les événements qui vérifient

$$|\Delta R(b, W) - \langle \Delta R(b, W) \rangle| < 2\sigma_{\Delta R(b,W)}$$
$$\Delta R(b, W) = \sqrt{(\eta_b - \eta_W)^2 + (\phi_b - \phi_W)^2}$$

$\Delta R(b, W)$ déterminé en associant les jets au niveau vrai (hadronique) aux partons.

$\Delta R(b, W)$ déterminé en utilisant la combinaison de jets obtenue grâce au χ^2.

Chevauchement des sélections aux niveaux vrai et reconstruit - Exemple

1 : Évenements présents uniquement dans la sélection au niveau vrai.

2 : Évenements présents dans les deux sélection (chevauchement).

3 : Évenements présents uniquement dans la sélection au niveau reconstruit.

Chevauchement des sélections aux niveaux vrai et reconstruit aux différentes étapes de la sélection

- Toutes les coupures ont été optimisées.

Chevauchement des sélections aux niveaux vrai et reconstruit en fonction des coupures

- Toutes les coupures ont été optimisées.
- L'étiquetage et la dernière coupure ont un effet important sur le chevauchement

Optimisation

Coupure	Étape précédente	Effet
Étiquetage	47.37	24.68
Distance b-W	23.47	12.39

- L'effet de l'étiquetage est grandement dû à un effet de détecteur.
- Plusieurs optimisations ont été testées pour la dernière coupure.

Les paramètres de la coupure ($\langle \Delta R(b, W) \rangle$ et σ) changent en fonction de l'impulsion transverse du top.

Factorisation

- Seule la coupure d'étiquetage est dominée effets de détecteurs.

Une possibilité pour la correction à la section efficace inclusive est de ne corriger que par l'efficacité des trois dernières coupures.

$$\epsilon_{t\bar{t}}^{truth} = \epsilon_{fidu} * \epsilon_{model} \qquad ; \qquad \epsilon_{fidu} = \epsilon_{blabel} * \epsilon_{isoB} * \epsilon_{\Delta R(b,W)}$$

$$\sigma_{fidu}(pp \rightarrow t\bar{t}) = \frac{N_{signal}^{had}}{\epsilon_{fidu} \cdot \mathcal{L} \cdot BR^{had}} = 2.13 \text{ pb}$$

Conclusion

Ce stage a été le deuxième que j'ai fait en laboratoire de recherche :

- Il a renforcé mon envie de faire de la recherche ainsi que ma passion pour la physique des particules.
- Ça a été l'occasion de me préparer à la thèse. J'ai également pu discuter avec les doctorants de mon bureau, ce qui a été particulièrement enrichissant.
- Le stage m'a permis de mettre en application mes connaissances de l'année, et renforcer des compétences techniques. Ça a également été l'opportunité d'acquérir un savoir-faire plus technique.