

Utilisation d'agents LCS

M. Gaciarz, P. Mathieu

Résumé

Très brève documentation portant sur l'utilisation des agents Lcs et des indicateurs nécessaires à leur fonctionnement.

1 Agents Lcs

Voir le fichier TestLcs.java, très documenté, explique l'utilisation des agents Lcs.

2 Indicateurs

Nous détaillons ici comment créer des indicateurs de marché. Ceux-ci peuvent être créés à partir d'un signal (pour les plus simples) ou en mémorisant une information à chaque instant (comme le prix courant, la profondeur du marché, etc) et étudiant son évolution (hausse par rapport à la valeur précédente, par rapport à la moyenne, ...).

2.1 Créer un indicateur à partir d'un Signal

Il suffit pour cela d'utiliser SignalDecorator, comme l'illustre l'exemple ci-dessous, qui crée un indicateur retournant aléatoirement vrai ou faux à partir du Signal RandomDirection.

```
Descriptor d=new SignalDecorator(new RandomDirection(0.0));
```

2.2 Créer un indicateur en mémorisant des informations sur le marché

L'article "apprendre des ordres" présente divers indicateurs basés sur les prix ou les ordres. Il est facile de les implémenter en utilisant 2 outils : les indicateurs génériques et les modules de mémoire.

2.2.1 Indicateurs génériques

Le tableau ci-dessous présente différents indicateurs basés sur les prix.

Il y a seulement 3 modèles d'indicateurs dans ce tableau :

- ceux qui comparent le prix à une valeur précédente (indicateur 1), qu'on appelle indicateurs A
- ceux qui comparent le prix à une moyenne de prix précédents (indicateurs 2 à 4), qu'on appelle indicateurs B

no	Indicateur de marché
1	$p_t > p_{t-1}$
2	$p_t > 1/5 \times \sum_{i=t-1}^{t-5} p_i$
3	$p_t > 1/10 \times \sum_{i=t-1}^{t-10} p_i$
4	$p_t > 1/100 \times \sum_{i=t-1}^{t-100} p_i$
5	$p_t > 1/2[Minp_i + Maxp_i]_{i \in [t-1, t-10]}$
6	$p_t > 1/2[Minp_i + Maxp_i]_{i \in [t-1, t-100]}$

TABLE 1 – Indicateurs à analyse technique communs aux deux approches (PriceLCS et OrderLCS)

- ceux qui comparent le prix courant à la moyenne des deux prix extrêmes sur une certaine durée (indicateurs 5 et 6), qu'on appelle indicateurs C

Avec un LCS capable de mémoriser les prix, il suffit de préciser le type d'indicateur, et la date du passé qui sert de point de comparaison. Par exemple, pour implémenter l'indicateur 2, on utilise un indicateur B ayant pour référence une date de 5 tics dans le passé. On l'implémente donc de la manière suivante :

```
MemoryModule priceModule=new PriceMemory();
Descriptor indic2=new DescriptorB(5, priceModule);
```

La première ligne signifie que l'indicateur raisonne sur l'historique des prix. Mais il est également possible d'utiliser d'autres informations que les prix, notamment des informations venant des ordres, grâce à la notion de modules de mémoire.

2.2.2 Modules de mémoire

Nous venons d'expliquer le fonctionnement d'un indicateur s'appuyant sur l'historique, mais il est possible de mémoriser et réutiliser n'importe quel type d'informations. Un MemoryModule permet de mesurer à chaque instant une information sur le marché, comme la profondeur de celui-ci, et de l'ajouter à un historique. Cet historique peut ensuite être utilisé par un indicateur générique pour décider d'une direction.

no	Indicateurs de marché
10	$r_t > 1/k_{10} \times \sum_{i=t-1}^{t-k_{10}} r_i$

TABLE 2 – Indicateur de marché basés sur l'évolution du ratio $r = \frac{\text{meilleur}P_{Vente}}{\text{meilleur}P_{Achat}}$

Il est très simple d'implémenter l'indicateur 10 présenté ci-dessus, avec par exemple $k_{10} = 10$:

```
MemoryModule bidAskModule=new SpreadMemory();
Descriptor indic2=new DescriptorB(10, bidAskModule);
```

3 modules différents sont implémentés dans ATOM :

- PriceMemory mémorise tout simplement le prix du titre à chaque instant
- SpreadMemory mémorise le ratio $r = \frac{\text{meilleur}P_{Vente}}{\text{meilleur}P_{Achat}}$
- DepthMemory mémorise le ratio $q = \frac{\text{offre}}{\text{demande}}$, avec :

$$\text{offre} = \sum_{\text{ordre} \in \text{Ventes}} \frac{\text{Quantite}(\text{ordre})}{\text{Prix}(\text{ordre}) - \text{bidAskMid}}$$

$$\text{demande} = \sum_{\text{ordre} \in \text{Achat}} \frac{\text{Quantite}(\text{ordre})}{\text{bidAskMid} - \text{Prix}(\text{ordre})}$$

$$\text{bidAskMid} = \frac{\text{meilleur}P_{Vente} + \text{meilleur}P_{Achat}}{2}$$

3 Conclusion

Grâce aux agents Lcs, il est facile de peupler les simulations réalisées sur ATOM avec des agents à apprentissage utilisant tous types d'informations.

Références