Qhapaq di molto(QDM)の詳細技術文書

パッショーネ将棋チーム：Ryoto Sawada

1. 開発動機

賞金を手に入れて参加費の元を取りたかった。

1. Qhapaq\_di\_moltoの構成

探索部：githubに公開されているやねうら王を利用(\*1)。技術的独自性は特になし。

評価関数：githubに公開されているNNUE型評価関数(\*1。tnkの実装をやねうら王にマージされたもの)を利用。少ない棋譜で過学習を起こさないように学習、解析を工夫している（後述）

並列計算：Hefeweizenが実装したMultiponderを拡張したPreponder付きMultiponderを独自に実装(\*2)

定跡：序盤で端歩をつくことをベースにしたものを自前で作成

1. 技術的な独自性

３−１．評価値に応じた動的な学習率の調整

図は将棋棋士、及び将棋ソフトの悪手率（手を指した前と後の評価値の差分から与えられる、どのぐらい悪い手を指してしまったかの指標）を手を指す前の局面の評価値別にプロットしたものである。対局者によって山の高さには差があるものの、ソフト、人間双方とも、評価値1000前後で最も悪い手を指しやすいことが解った。評価関数の表現力に限界があると仮定した場合、より重要な局面を正しく評価できるようにする必要がある。そこで、QDMの学習では教師データの評価値に応じて学習率を動的に変えることで、重要な局面を効率的に学習させることを考えた。具体的には、学習率に d^2/dv (1/(1+exp(-v/600))) で定義されるボーナスを与えた。



また、今回の評価関数の教師データにはレート測定用の自己対局（４コア８スレッド１手１秒、定跡にはtanuki-が公開している互角局面を利用）、及び、運営しているレーティングサイトに投稿された棋譜を用いた。教師データの総数は約１０００万局を用いた。

３−２．PreponderつきMultiponder

並列計算の手法としてはPreponderつきMultiponderを用いた。従来のponderは相手の手番中に相手が一番指しそうな手（候補手）についてのみ探索を行う。即ち、候補手が当たれば相手の思考時間を自分の持ち時間に実効的に加えることができるが、候補手が外れれば相手の思考時間中の読みは意味を失ってしまう。これに対し、Multiponderでは複数のノードを用意しそれぞれに別の候補手を読ませることでponderの一致率を上昇させている。PreponderはMultiponderの発展形であり、自分の手番中に、２手先の展開について予め他のノードに計算を行わせることで、自分の思考時間もponderに用いようとするものである。

３−３．合法手列挙におけるソートの高速化

詳しくは <https://github.com/TokusiN/SuperSort/> を参照

1. 実験結果

強化学習について；

Qhapaq Research Labで公開されている、orqha1018、及び、QDMの評価関数は前述の方法を用いて学習されている。学習元であったillqha（めきっと氏が開発。orqhaの学習元）やt.n.kの評価関数よりも強い評価関数が得られている。

ただし、ボーナス値の付け方の最適化は行えていない。レート測定用の自己対局という深く正確な教師データを用いて、学習率を下げて微調整をしたことがレート上昇の本質的な要因である可能性も否定できない。

Preponderについて：

下図はQhapaqの自己対局で、通常のponderを用いたもの vs 3ノードを用意したPreponderを用いたものの自己対局でPreponder側のponder hitの内訳を分類したものである。似通った探索、評価部であれば、実効的な持ち時間を通常のPonderに比べ増やすことができている。



1. 追試が可能であるかについて

学習については教師局面のシャッフルなどに乱数が伴うため、同じデータを作りなおすことは難しい。Preponderについてはソースコードを公開しているので、それを用いれば、上記実験結果を再現できる。

1. 開発過程

2018/10月頃まで　orqhaを作る

2019/2月頃　ライブラリルールの変化に伴い、tnkからの再学習が必要になる

2019/4月頃　評価関数の学習の疲れからかPreponderを実装を開始する

2019/5/5　ソートが相当早くなる