PEIOPT.py ソースコード

bound.csv Xの範囲



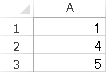
１列目：x1のパラメータ

２列目：x2のパラメータ

１行目：下限値

２行目：上限値

yinf.csv Yの制約条件



１列目：y1のパラメータ

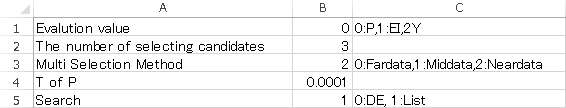
２列目：y2のパラメータ

１行目：0:２行目が下限値、３行目が上限値　1:最大化　-1:最小化

２行目：下限値（１行目が0の場合のみ有効）

３行目：上限値（１行目が0の場合のみ有効）

option.csv その他のオプション



　１行目：評価値（０：確率、１：期待値、２：予測値そのもの）

　２行目：出力候補数

　※学習データを用いてGPを構築

最も予測値の良い候補を選択し、dummyのy値として3で設定した値を代入

GPを再構築

この処理を繰り返すことでN候補作成し、出力

　３行目：dummyの値（０：最大化なら最小値、最小化なら最大値　１：中央値　２：最大化なら最大値、最小化なら最小値）

　４行目：例えば最大化を行う場合、(現在の最大値＋設定値)～∞の範囲で積分し確率を求める。

＝＝＝なぜ必要か？＝＝＝

最大化を行う際には、現在の最大値を超える確率が最大となるXが選択される。GPの予測値の最大値にデータが存在するケース等では、該当データにおける確率は0.5となる。その他のXに関しては、GPの予測値が最大値以下となる為、確率も0.5以下になる。この場合、新しい候補が選択されなくなってしまう（下図左）。そこで、現在の最大値に設定値を加算した値を超える確率を求めることで、現在の最大値近傍以外のXが候補として出力されるようにしている（下図右）。積分範囲を決定する際に、この値が大きいほど広い範囲が探索できる。設定値の目安はyの値をどれだけ改善したいかで決定。設定値の単位はyの単位と同一。

50%（最大）

50%以下

ほぼ0%

ほぼ0%

ほぼ0%

15%（最大）

　５行目：０：Differential Evolutionを用いて最適な候補を探す　１：Xeval.csvの中から最適な候補を探す。

X.csv トレーニングデータ

y.csv トレーニングデータ

Xeval.csv テストデータ

result.csv 出力ファイル