Scikit-Learn

1.4.4 Nearest Neighbors Algorithms

茨城大学大学院理工学研究科情報工学専攻

菊池裕紀

1.4.4.1 Brute Force

- 全ての距離を計算して比較する（総当り）
  - ➢最も単純な手法
  - ➢データ数が少ない場合は効果的
  - ➢データ数が増えると実行が不可能になってしまう
- algorithm=“brute”で提供されている

1.4.4.2 K-D Tree

- データ構造の一種
  - ➢最近傍探索などを効率よく行うためのデータ構造
  - ➢k次元の空間を平衡2分木を用いて、各次元の平面で2分割しておき、これを二分探索する
  - ➢次元が大きくなると効果がなくなってしまう（次元の呪い）
- algorithm=“kd_tree”で提供されている

1.4.4.3 Ball Tree

- 次元の呪いに対応するようにK-D Treeを拡張したもの
  - 高次元でも効果的
  - 重心C、半径rで定義された球上にデータを配置する

- algorithm=“ball_tree”で提供されている

1.4.4.4 Choice of Nearest Neighbors Algorithm

- 最適なアルゴリズムの選択はとても複雑であり、要素の数による
  - ➢サンプル数が少なければ“Brute Force”が有効

    30以下のデータ数ならばBrute Forceで問題ない
  - ➢データの次元数やスパース性の影響

    Brute Forceは影響を受けないが、K-D TreeやBall Treeは影響をうける
  - ➢近接点の数

    Brute Forceはあまり関係ないが、K-D TreeやBall Treeは影響をうける
  - ➢query pointの数