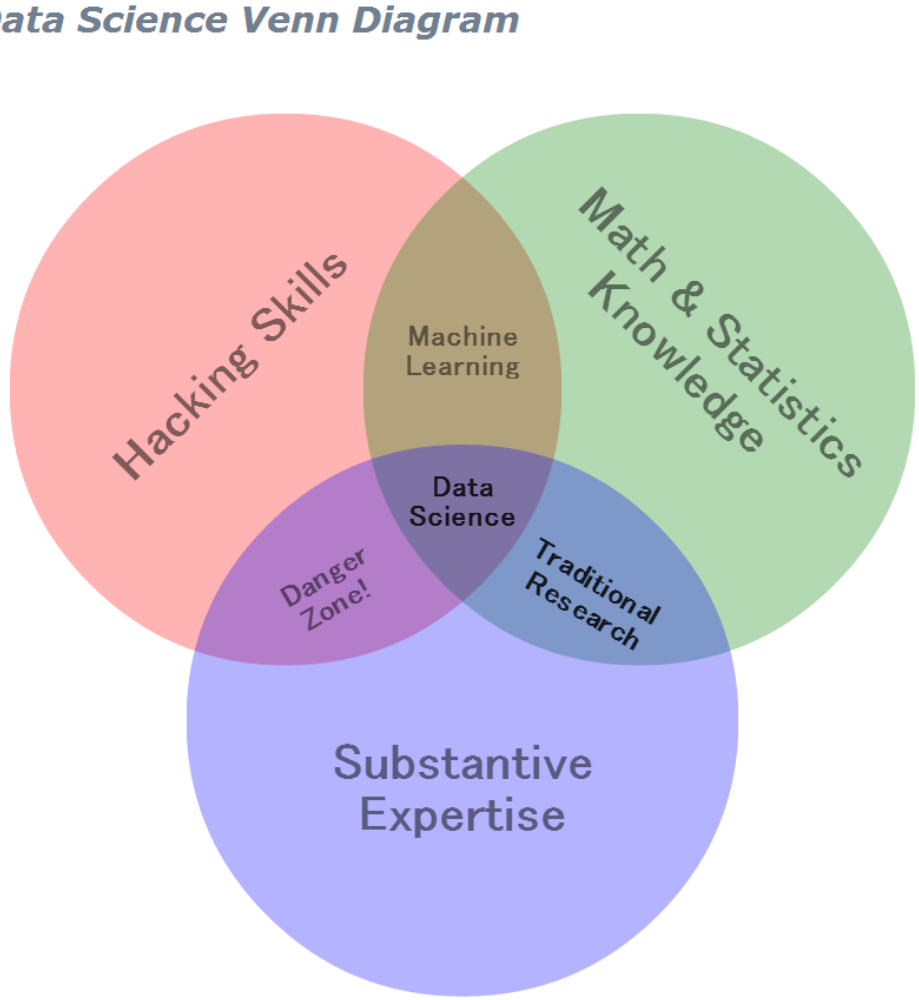
機械学習入門

経済学部 BX584

機械学習の概観

https://www.nttpc.co.jp/cgi-bin/gpu/simulation/dgx/index.cgi

機械学習の位置づけ
個別分野のデータを分析するための道具



http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

機械学習の概観

- 機械学習の3種類
  - 教師あり学習
  - 教師なし学習（この授業では扱わない）
    - 私の専門は、これ。
  - 強化学習（この授業では扱わない）

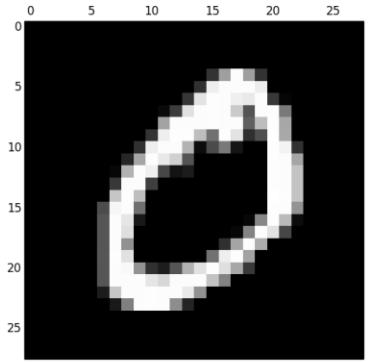
教師あり学習

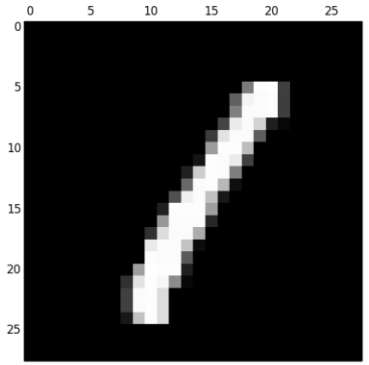
- 入力データから望ましい出力データを予測する
  - 入力データから望ましい出力データを算出する関数を探す
  - 良い関数＝望ましい出力データに近い出力データを与える関数
- 教師あり学習とは・・・
  - （入力データ，対応する望ましい出力データ）のペアがたくさんある
    - ないときは・・・個々の入力データに望ましい出力データを人間が付与する
  - このようなデータを使って、良い関数を機械に見つけさせる
    - 得られた関数は、よく「予測モデル」と呼ばれる

教師あり学習の2種類

- 分類 (classification)
  - 出力が離散値（「ラベル」「クラスラベル」とも呼ぶ)
- 回帰 (regression)
  - 出力が連続値

例）計算機で手書き数字画像を分類





目標の値

"0"

"1"

分類（classification）

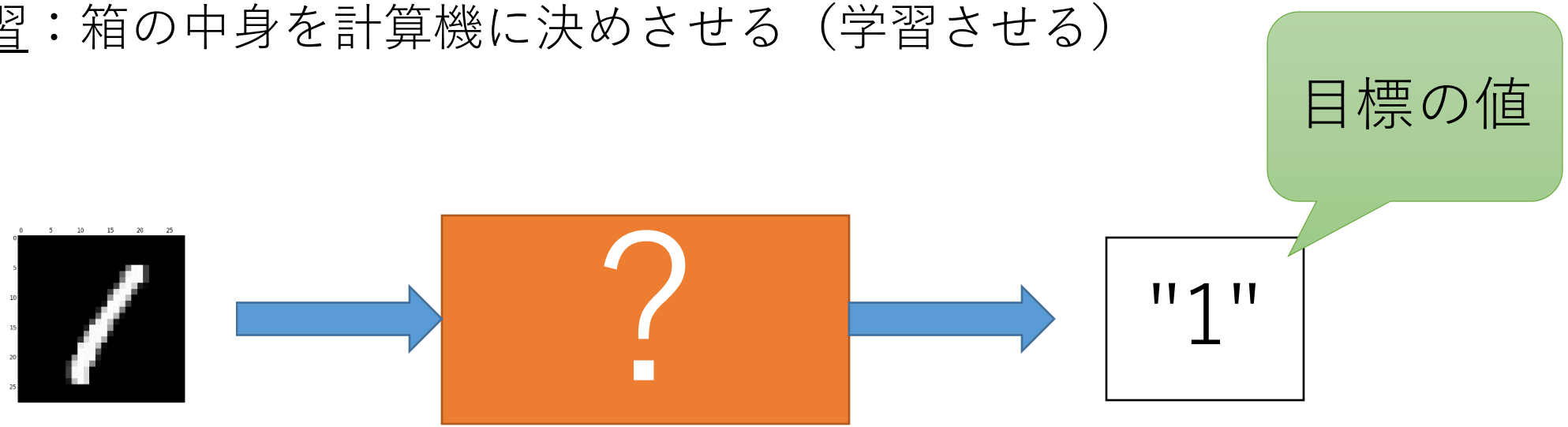
- 入力：分類したいデータ
- 出力：どのグループへ分類すべきかを示す値（ラベル）
- 与えられた入力を変換し、所望の出力を得る計算方法（関数）が欲しい！
  - この変換の計算は、計算機で実行するので、いくら複雑でも構わない。
  - 所望の出力が得られる計算方法をどうやって見つけるかが、問題。

回帰（regression）

- 入力：回帰したいデータ
- 出力："どこ"へ回帰すべきかを示す値（連続値）
- 与えられた入力を変換し、所望の出力を得る計算方法（関数）が欲しい！
  - この変換の計算は、計算機で実行するので、いくら複雑でも構わない。
  - 所望の出力が得られる計算方法をどうやって見つけるかが、問題。

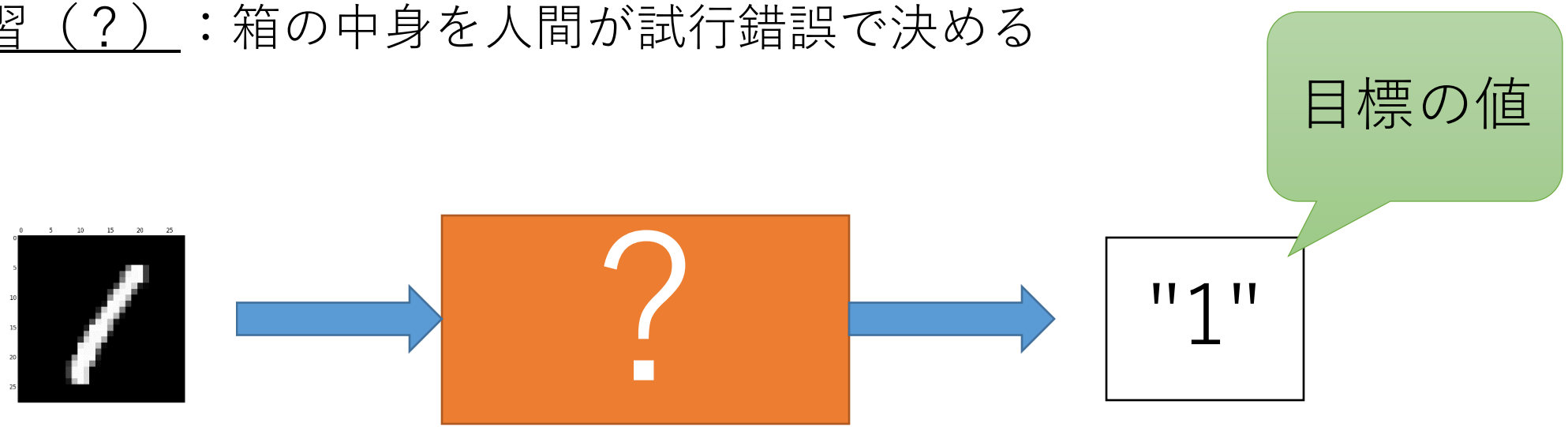
機械学習（計算機に学習させる）とは・・・

- 適当な値を入れると、その値に応じて何かの値が出てくる箱がある
- 学習：出てくる値が目標の値になるように、箱の中身を決める
- 機械学習：箱の中身を計算機に決めさせる（学習させる）



機械学習ではない学習とは・・・

- 適当な値を入れると、その値に応じて何かの値が出てくる箱がある
- 学習：出てくる値が目標の値になるように、箱の中身を決める
- 人間学習（？）：箱の中身を人間が試行錯誤で決める



機械学習とは

良い関数を計算機に見つけてもらうこと。

関数＝適当な値を入れると、その値に応じて何かの値が出てくる箱
良い関数＝どんな値を入れても、出てくる値が望ましい値

計算機に見つけてもらうのであって、
人間が試行錯誤して見つけるのではない。

もう少しテクニカルに言うと・・・

関数の、無数にあるパラメータ設定の中から
良い設定を計算機で見つけること。

良い設定＝どんな入力に対しても（見たことがない入力に対しても）
望みどおりの出力が得られる

関数を選ぶ範囲

- 関数を選ぶ範囲は、あらかじめ決めておく
- どう決める？
  - 入力データのフォーマットを決める
  - 出力データのフォーマットを決める
  - 計算式の「かたち」を決める
    - 計算式はパラメータを含む（このパラメータを計算機で決めるのが機械学習）

例）一次式 $y = ax + b$ （xが入力、yが出力）

機械学習を実践する

機械学習実践の流れ

1. データの準備・前処理
2. 手法の選択
3. 実際に計算機に学習させる
4. 得られた予測モデルの評価
5. 新しいデータに対して予測をおこなう

１．データの準備・前処理

- データがあってはじめて機械学習を使える
  - データは電子化されているか？
    - 紙に印刷されたデータはダメ
  - 機械学習ライブラリで使えるフォーマットか？
    - PDFやWordはダメ
      - 使えるフォーマットへ変換できるか？
    - Excelは良い（pandasで扱える）
    - CSVは良い（テキストファイルなのでPythonでじかに読める）

教師あり学習のためのデータの準備

- 教師あり学習を使うためにはラベル付きデータが必要
  - ラベル（望ましい出力）を人間が付与する
- ラベル付きデータは多ければ多いほど良い
  - 人間による作業のコストを頭に入れておく
- 既存のラベル付きデータを使う可能性もある
  - 自分がしたいと思っている予測に合っているデータを探す
    - そういうデータがあるかどうか・・・

データの前処理

- データの内容は（ある程度）人間の目でチェックする
  - 明らかにおかしいデータは事前にはじく

    例：CSVの表形式のデータで部分的に列がズレていた

    例：欠損値があった（欠損値はどう扱う？）
  - 実際には機械学習のアルゴリズムを動かしてから気づくことも多い

    例：日本語のテキストデータに英語データが混ざっていた
- フォーマットはライブラリを駆使して変換
  - Pythonはフォーマット変換のライブラリが充実している

２．機械学習の手法の選択

- 分類（連続値入力から離散値出力を予測）だけでも様々
  - 決定木（この講座では扱わない）
  - ナイーヴベイズ（この講座では扱わない）
  - k-最近傍法
  - パーセプトロン
  - SVM
  - ロジスティック回帰
  - 多層パーセプトロン

    ・・・

提案：SVMから始める

- 分類問題ならとりあえずSVMから
  - 枯れた手法
  - ライブラリも充実
    - Pythonならscikit-learnの実装を使えばよい
    - 最近の実装は計算速度も良い
  - より複雑な手法はあとから試す
    - 複雑な手法を使ってもSVMより悪いなら意味がない

この授業での順番

- k-最近傍法
  - ノンパラメトリックな手法
- 一変数入力の線型回帰（単回帰）
  - 一変数の線形回帰は（NN含む）ほぼすべての教師あり学習の基本
- 多変数入力の線型回帰（重回帰）
- ロジスティック回帰
  - 「回帰」という名前だが二値分類の手法
- SVM
  - おおよその原理だけ説明
    - 実は奥が深い手法だが詳細は省略

３．実際に計算機に学習させる

- 損失関数を決めておく
  - 望ましい出力からのズレを求める関数
- 学習のプロセス
  - 与えられた入力に対して予測モデルの出力を計算
  - その出力値が望ましい出力からどのくらいズレているかを計算
    - ここで損失関数を使う。
  - そのズレが小さくなる方向へ予測モデルの中身を変化させる
    - 損失関数の勾配を利用して予測モデルのパラメータを動かす。

教師あり学習に登場するふたつの関数

- 入力と出力の対応を表わす関数＝予測モデル
  - 与えられた入力について望ましい出力を求めるための関数
  - 教師あり学習ではこの関数のパラメータを計算機に調整させる
- 予測モデルの出力の良さ（悪さ）を表す関数＝損失関数
  - 予測モデルの出力と望ましい出力の「距離」を求める関数
    - 損失関数の値は小さければ小さいほど良い
  - 損失関数が最小化されるように予測モデルのパラメータを動かす
    - 勾配（各パラメータによる微分値を集めたもの）を利用する

４．評価

- 評価用のデータは別に用意しておく
  - 訓練データ：予測モデルのパラメータを決めるために使うデータ
  - 評価用データに訓練データと同じデータが混ざっていてはいけない
    - 検証データ
- 訓練データでは良い性能が出たが評価用データではダメだった
  - この現象を「オーバーフィッティング（過学習）」と呼ぶ
    - 教師あり学習では、このオーバーフィッティングを避けなければいけない
  - 訓練データは、本来はたくさんあるデータの一部、とみなすべき
    - 訓練データ以外のデータで望ましい出力が得られないと意味がない

5．新しいデータで予測

- 実際にモデルを運用する
  - しかし…運用前に運用時の性能を知りたい
- 機械学習の実験をするときはデータを3つに分ける
  - 訓練データ
    - 予測モデルに学習させるためのデータ
  - 検証データ
    - 予測モデルのハイパーパラメータをチューニングするためのデータ
  - テストデータ
    - 実際の運用（＝予測モデルはもう変更しない）を模擬的におこなうためのデータ
      - 完全に未知のデータとして扱う

機械学習実践の流れ

1. データの準備・前処理
2. 手法の選択
3. 実際に計算機に学習させる
4. 得られた予測モデルの評価
5. 新しいデータに対して予測をおこなう

課題20250602

NumPy を使って以下の計算をするPythonプログラムを書きなさい。

- 縦4、横4のサイズのNumPyの配列を作る。ただし、配列の中は1から20の範囲の整数の乱数で埋める。
- その配列の中にあるすべての偶数を、-1という値で置き換える。
- そのあとで、その配列のすべての要素の和を計算する。
- 元の配列、偶数を-1で置き換えた後の配列、そして、すべての要素の和を、順に表示すること。

- ヒント： 1.では numpy.random.randint を使う。

（参考資料： https://www.sejuku.net/blog/67872Links to an external site. ）