統計処理及び機械学習に基づくデータマイニング入門 第1回

宮本　隆志

ナビプラス株式会社

February 27, 2015

この勉強会について

- データマイニングの入門講座です。
  - 想定する聴衆は、これからデータを解析してみようという初学者を想定しています。
  - 専門家や既に実務経験の豊富な方々には物足りない内容かと思います。
- データマイニング ＝
  - 大量のデータを
  - 統計学や機械学習などの手法を用いて探索・分析して
  - 意味あるパターンやルールを発見する

  と考えます。この勉強会では手法の話をします。
- ツールは無料のオープンソースのものを使用します
  - メインに Python を使用します。Anaconda-2.1.0 を用いて説明します。
- 参考書はイベント Web ページには記載しましたが、あまり準拠しません。
  - あまり準拠すると著作権的に問題があるので。

自己紹介

名前 宮本　隆志 (@tmiya_)

所属/仕事 ナビプラス株式会社 / データ解析周りの R&D の仕事

ナビプラス マーケティングソリューションツールの開発・提供

- サイト内検索エンジン・レコメンドエンジン、レヴュー投稿エンジンが中心
- 次世代インターネットサービスの研究・開発
- 上記に付随する広告商品の販売

前職 ネット広告の入札サーバを開発する会社で似たような仕事

興味 機械学習 / 関数型言語 / 定理証明系

- Coq という定理証明系の勉強会を毎月開催しています

勉強会の進め方：予定

- 講義 (30 分 + 30 分) + 実習 (40 分) の形式。
- 前半は統計処理とか機械学習の手法の講義を中心。
  - 過去に開催された「プログラマの為の数学勉強会」の内容に関しては、繰り返し説明は控えるつもり。
  - アルゴリズムの詳細については、説明全てを繰り返すのは難しいので「パターン認識・機械学習勉強会」資料を参照で。
- 後半は Python を用いて簡単なデータ処理のハンズオンを予定。
  - 興味のない方は講義について質問したり退出したり Python 以外で解析するとかでお願いします。
  - Python 環境は Anaconda-2.1.0 を推奨しますが、各自に任せます。
  - 未導入の方は次のページで説明します。
- 退出前にアンケートの記入をお願い致します。講義の改善と実習用サンプルデータに使用させて頂きます。
- 講義形式の勉強会とは別に、読書会（教科書とか論文とか）とかやりたいので興味のある方、後で懇親会とかで相談しましょう。

Anaconda 導入

まだ python 環境を導入していない人は、講義をしてる間に導入して下さい。

- 無線 LAN の SSID, password はホワイトボードを見て下さい。
- 電源容量の関係で、延長コードによる過度のタコ足配線はご遠慮ください。
- Google 検索「python anaconda」→ダウンロードページへ
- Python version は version 2.7 を使用するつもりです。
- Windows/Mac/Linux 自分のマシンに合わせて。
- PATH を通したりとかは、ダウンロードページの指示に従って下さい。
- 不明な点は休み時間かハンズオンの時間にお願いします。

参考文献

データマイニング全般：各手法に関しては都度紹介します。

- ”Data Mining Techniques, 3rd edition” (Linoff and Berry)
- **「データマイニング手法」 上の和訳**
- **「購買心理を読み解く統計学」 (豊田秀樹)**

統計学：各自お好きな教科書で。

- **東京大学教養学部統計学教室編　基礎統計学 I　III 「統計学入門」「自然科学の統計学」「人文・社会科学の統計学」**

機械学習：各自お好きな教科書で。

- **「パターン認識と機械学習（上下)」(ビショップ)**
- **「データマイニングの基礎」(元田、津本、山口、沼尾)**

Python **での機械学習：各自好きな教科書で**

- ”Learning scikit-learn: Machine Learning in Python” (Garreta and Moncecchi)

データマイニングの作業

- データマイニング作業：下記の３ステップ＋１作業
  - データの準備：データの取得、前処理、集計
  - モデルの適用：モデル選択、パラメータ決定
  - モデルの評価・検証
  - 上記の各局面での視覚化
- 代表的なモデル：その他も出来る範囲内で紹介する予定です
  - 回帰分析
  - クラスタリング
  - アソシエーション分析

データマイニングの作業

先ほど「モデルの評価・検証」を行うと述べました。

- 検証の必要性
  - そもそもデータに適合したモデルでなければ役に立たない。
  - 過学習：データへの適合度だけ考慮すると、モデルは不必要に複雑化する。
  - オッカムの剃刀：より少ない要因やパラメータで、現象を説明したい/理解したい。
- モデル選択方法：あるパラメータ/項をモデルに含めるか否か？
  - 交差検証 (cross validation)：解析に使用しない検証用データを残しておいて、そのデータを用いてモデルの良し悪しを判断 (ホールドアウト検証)。他に K 分割検証、一個抜き交差検証、など。
  - 情報量基準：対数尤度にパラメータ数によるペナルティを考慮した数値。情報量基準が最良となるモデルを選択する。
  - 正則化：モデル自身にパラメータ数を抑制する項を追加する。(L1 正則化：多くのパラメータが 0 になる)
  - 有意性検定：帰無仮説 $H_i : \theta_i = 0$ が棄却されるかで判断。(検定の話は第２回で)

線形回帰モデル

今回は一番単純なモデルの話として、最小二乗法を取り上げます。
入力変数 x の関数からなる基底 $\phi_0(x), \cdots, \phi_M(x)$ を考えます。(但し $\phi_0(x) = 1$)
関数 $y(x)$ が重み $\mathrm{w} = (w_0, \cdots, w_M)^T$ を用いて基底 $\{\phi_m(x)\}$ の線形結合

$$y(x, \mathrm{w}) = \sum_{m=0}^{M} w_m \phi_m(x)$$

と書ける時、線形回帰モデルと呼びます。
例えば $\phi_m(x) = x^m$ とすると

$$y(x, \mathrm{w}) = w_0 + w_1 x + \cdots + w_M x^M$$

と M 次多項式でモデリングすることになります。

正規分布

観測された値 $\mathrm{t} = (t_1, \cdots, t_N)^T$ は真の値 $y_n = y(x_n, \mathrm{w})$ と誤差 ϵ_n の和となります。誤差は正規分布 $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$ と考えることにします。一変数の正規分布は

$$\mathcal{N}(\mu, \sigma^2)(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}.$$

x, w, σ^2 が与えられた時の t の分布 (=尤度 $\mathcal{L}$) は

$$\mathcal{L} = p(\mathrm{t}|\mathrm{x}, \mathrm{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathrm{w}^T\phi(x_n), \beta^{-1})$$

と書けます。

最尤推定

最尤推定では尤度 $\mathcal{L}$ を最大化することでパラメータ $\mathbf{w}, \sigma^2$ を求めます。対数尤度 $\ln \mathcal{L}$ は

$$\ln \mathcal{L} = -\frac{N}{2}\ln(2\pi) - N\ln\sigma - \frac{1}{\sigma^2}E(\mathbf{w})$$

但し、二乗和誤差関数 $E(\mathbf{w})$ は

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{t}_n - \mathbf{w}^T\phi(x_n)\right\}^2 \\ &= \frac{1}{2}(\Phi\mathbf{w} - \mathbf{t})^T(\Phi\mathbf{w} - \mathbf{t}). \end{aligned}$$

です。 ただし、計画行列 Φ は

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_M(x_1) \\ \vdots & \ddots & \vdots \\ \phi_0(x_N) & \cdots & \phi_M(x_N) \end{pmatrix}.$$

正規方程式

$$\frac{\partial \ln \mathcal{L}}{\partial \mathbf{w}} = -\frac{1}{\sigma^2}\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}, \quad \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{2}(\Phi\mathbf{w} - \mathbf{t})^T(\Phi\mathbf{w} - \mathbf{t})$$

より、$\mathbf{w}$ を求める為には二乗和誤差関数 $E(\mathbf{w})$ を最小化すれば良いことが判ります。（最小二乗法）
二乗和誤差関数 $E(\mathbf{w})$ を $\mathbf{w}$ で微分して

$$\begin{aligned}
\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{2}\frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T\Phi^T\Phi\mathbf{w} - \mathbf{w}^T\Phi^T\mathbf{t} - \mathbf{t}^T\Phi\mathbf{w} + \mathbf{t}^T\mathbf{t}) \\
&= \frac{1}{2}\{\Phi^T\Phi + (\Phi^T\Phi)^T\}\mathbf{w} - \frac{1}{2}\Phi^T\mathbf{t} - \frac{1}{2}(\mathbf{t}^T\Phi)^T \\
&= (\Phi^T\Phi)\mathbf{w} - \Phi^T\mathbf{t} = 0.
\end{aligned}$$

これを解いて正規方程式

$$\mathbf{w} = \Phi^+\mathbf{t}, \quad \Phi^+ = \left(\Phi^T\Phi\right)^{-1}\Phi^T \text{ (擬似逆行列)}$$

を得ます。(詳細は例えば PRML 3. 1. 1 参照)

正則化最小二乗法

最小化する誤差関数 $E(\mathrm{w})$ に、正則化項

$$E_1(\mathrm{w}) = \lambda \sum_i |w_i|, \quad E_2(\mathrm{w}) = \lambda \sum_i w_i^2$$

などを足したものを考えます。これはパラメータ w に対するペナルティとして働き、モデルの複雑さを減らします。
正則化項 E_1 を加えた場合 (Lasso)、λ を増やすと $w_i = 0$ となる項が増えます。(See PRML 3. 1. 4.)
正則化項 E_2 を加えた場合 (Ridge)、

$$\mathbf{w} = \left(\boldsymbol{\lambda}\mathbf{I} + \boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\mathbf{t}$$

となります。対角項 λI を加えることで逆行列の計算が安定するようになります。(See PRML 3. 1. 2.)
ElasticNet は Lasso + Ridge を混合比 `l1_ratio` で組み合わせたものです。変数の間に相関がある場合、Lasso より安定した解を返します。

情報量基準

パラメータの数を増やせば増やすほど、モデルは測定データとの適合度は高くなります。しかし、過剰にデータに適合させてしまうため、未知の入力に対する予測性能、という意味では悪化します。

- 例えば１変数多項式で近似する場合、M 個のデータ点全てを通る $M-1$ 次多項式が作れます。しかし、与えたデータ点以外の場所での一致は悪くなります。

つまり「モデルの複雑さ」と「データの適合度」のバランスをとることが必要です。
情報量基準はこの問題への解となります。情報量基準の中で、AIC (赤池情報量基準) という量は

$$\mathrm{AIC} = -2\ln\mathcal{L} + 2K,\ \ \mathcal{L} = \text{尤度}, K = \text{パラメータ数}$$

で定義されます。 同様の量として BIC (ベイズ情報量基準) などの量があります。

Python に関して (1)

なぜ○○じゃなく Python なの？

- 他に OSS の選択肢はある。
- GUI ベースだと文章で説明し辛い。
- 前の勉強会も Python を使ってたので。
- 私がこの機会に勉強したかった。

なぜ IPython？

- インタラクティブシェル
- グラフ表示 (matplotlib)
- markdown と LaTeX 数式表示
- ノートブックとして公開

なぜ Anaconda？

- 数値計算とかに便利なライブラリが一式つまった Python のディストリビューションで便利
- 講義する場合、環境が揃ってた方が説明しやすい。

Python に関して (2)

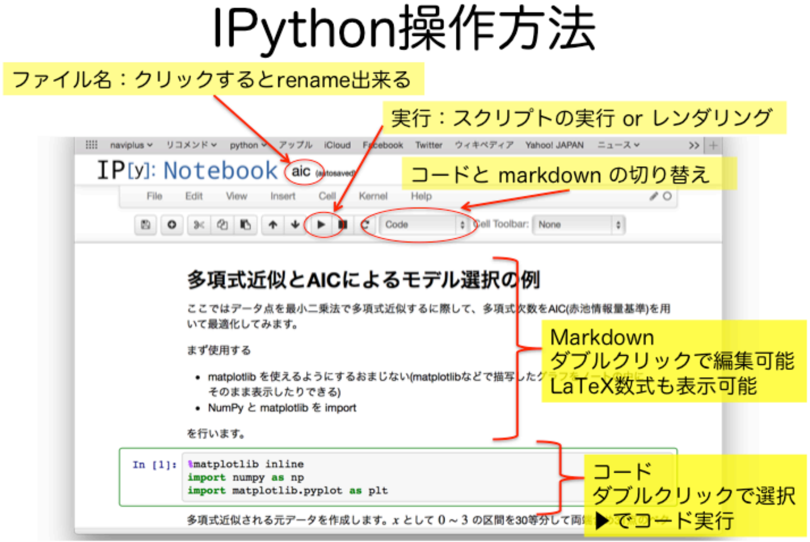
主要なライブラリ（今回使用分）

- matplotlib　様々な形式でのプロットを行う
- numpy　多次元配列とその計算、線形代数、FFT、乱数など
- scipy　科学技術計算用ライブラリ
- IPython　インタラクティブ環境
- pandas　データ形式と各種解析
- scikit-learn　機械学習用ライブラリ、テスト用データセット
- statsmodels　統計処理ライブラリ

IPython

起動方法：`$ ipython notebook`

⇒ Home **画面：ノートブック選択** or **新規作成** ⇒



ハンズオン (1)

ハンズオンファイルの取得とノートブックの開き方

- **適当な** directory **で**
  ```
  $ git clone https://github.com/takashi-miyamoto-naviplus/spml4dm.git
  ```
  を行ってください。
- git **を導入してない場合は、**nbviewer **経由でダウンロード（右上のボタンでローカルに保存します）**
  - http://nbviewer.ipython.org/github/takashi-miyamoto-naviplus/spml4dm/blob/master/1/aic.ipynb
  - http://nbviewer.ipython.org/github/takashi-miyamoto-naviplus/spml4dm/blob/master/1/boston.ipynb
- **無線** LAN **が使えない人は、**USB **メモリでファイルを渡します。**
- **ダウンロードしたファイルのあるディレクトリで**
  ```
  $ ipython notebook
  ```
  を実行

ハンズオン (2)

多項式近似と AIC によるモデル選択の例

- **この演習では、線形回帰の練習として** sin **関数に誤差を乗せたデータ点に対して、多項式近似を行い、**AIC **を使ってモデル (多項式次数) を決定することを学びます。**
- **また** NumPy **ライブラリを用いた線形回帰の行い方、**matplotlib **を用いたグラフのプロット方法について練習します。**
- **ブラウザ上で** `spml4dm/1/aic.ipynb` **を開いてください。**
  - git **を使わない場合は、**`http://nbviewer.ipython.org/github/takashi-miyamoto-naviplus/spml4dm/blob/master/1/aic.ipynb` **で直接開くこともできます。**
  - **右上のダウンロードボタンでローカル保存してください。**

ハンズオン (3)

scikit-learn を使った線形回帰

- この演習では、boston データセットを用いて線形回帰の練習をします。このデータセットには、ボストンの建物の価格と建物毎の条件 (部屋数や面積、近くの犯罪率、小学校の生徒数に対する教師の数など) が書かれています。線形回帰を行い、各種条件から建物の価格を求めます。
- 線形回帰に際して、交差検定付きの Ridge, Lasso などの正則化を行い、過学習を避ける工夫をします。
- また scikit-learn を用いて各種線形回帰を行う、pandas, matplotlib を用いてデータの集計と可視化を行う練習をします。
- `spml4dm/1/boston.ipynb` を開いてください。
  - git を使わない場合は同様に nbviewer 経由でアクセスして、保存してください。

ハンズオン用 URL

git で取得できない場合は nbviewer 経由でアクセスできます

git 取得方法 `$ git clone https://github.com/takashi-miyamoto-naviplus/spml4dm.git`

aic.ipynb 取得 `http://nbviewer.ipython.org/github/takashi-miyamoto-naviplus/spml4dm/blob/master/1/aic.ipynb`

boston.ipynb 取得 `http://nbviewer.ipython.org/github/takashi-miyamoto-naviplus/spml4dm/blob/master/1/boston.ipynb`

帰る前にアンケートの記入をお願いします。

次回：検定の話

次回は、χ^2 検定を中心に、検定の話をしようと思います。

今回使用した公式集

ベクトル $\mathbf{x}$ での微分：ベクトル $\mathbf{a}$ 、正方行列 $\mathbf{A}$ が定数の時に

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{a}^T \mathbf{x} = \frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{a} = \mathbf{a},$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{x} = (\mathbf{A} + \mathbf{A}^T) \mathbf{x}.$$

となります。
オンラインで読める数学公式集としては「線形代数公式集 (仮)」 (http://sci-tech.ksc.kwansei.ac.jp/~inagai/write/stat_linear.pdf) **とかお勧めです。**
書籍の場合、 PRML **上巻付録とか「統計のための行列代数（上下)」とかが良いです。**