機械学習

線形回帰モデル

回帰問題
・回帰で扱うデータ
　入力ベクトルの各要素…説明変数（または特徴量）
　出力（スカラー値）　…目的変数

回帰問題を解くための機械学習モデル
・線形回帰モデル
　教師あり学習（正解付きデータから学習）
　入力とm次元パラメータの線形結合を出力
　$w = (w_1 、w_2 、\dots w_m)^T \in \mathbb{R}^m$
　$\hat{y} = w^T x + b = \sum_{j=1}^{m} w_j x_j + b$

線形結合
・入力ベクトルとパラメータの内積をとり切片を足し合わせたもの

線形回帰モデルのパラメータ
・モデルのパラメータ
　特徴量が予測値に与える影響を決定する重みの集合
　重みが大きければ予測に大きな影響を与え、重みが０なら予測への影響も０

線形単回帰モデル（ｍ＝１）
・データへの仮定
　データは、回帰直線に誤差が加わった形で観測されていると仮定する
　$y = (w_0 + w_1 x_1 + \varepsilon)$
　切片w_0、回帰係数w_1を学習で決める
　未知パラメータは最小二乗法により推定→誤差は正規分布を仮定しなくてもよい。
　最尤法を使う場合は正規分布を仮定するとより詳細な分析が可能となる。

線形重回帰モデル（ｍ＝多次元）
・データへの仮定
　データは、回帰平面に誤差が加わった形で観測されていると仮定する。
　$y = (w_0 + w_1 x_1 + w_2 x_2 + \varepsilon)$
　切片w_0、回帰係数w_1、w_2を学習で決める

データの分割
・データを学習用と検証用に分割
　→学習用を８０％、検証用を２０％など。

パラメータの推定
・平均二乗誤差（Ｍｅａｎ　Ｓｑｕａｒｅ　Ｅｒｒｏｒ）
　データとモデル出力の二乗誤差
・最小二乗法
　学習データの平均二乗誤差を最小とするパラメータを探索
　→勾配が０となる点を求める

非線形回帰モデル
・現実問題、線形なデータは少ない
　→非線形回帰モデル
・非線形回帰モデル
　多項式…次数の高い多項式を考えると非線形のデータをとらえることができる
　ガウス型基底

正則化法
・未学習
　学習データに対して十分小さい誤差を得られない場合
・過学習
　小さい誤差を得られたが、テスト集合との誤差が大きい場合
　→正則化で回避する
・汎化性能

新たな入力に対する予測性能
汎化誤差が小さいものが良い性能を持ったモデルといえる。
・正則化法
モデルの複雑さに伴ってその値が大きくなるペナルティ項（あるいは正則化項）を課した関数の最小化を考える。
正則化パラメータは、モデルの曲線の滑らかさを調節し、推定量の安定に寄与する。
・ペナルティ項
ＲＩＤＧＥ推定量：Ｌ２ノルムを利用。パラメータを０に近づける（縮小推定）
ＬＡＳＳＯ推定量：Ｌ１ノルムを利用。いくつかのパラメータを０と推定（スパース推定）

モデル選択
・ホールドアウト法
手元のデータを２つに分割。
一方を学習に使用、もう一方をテストに使用し予測精度や誤り率を推定。
・クロスバリデーション
手元の各クラスのデータをm個に分割。
ｍ－１個のデータを使用して識別器を学習、１つのグループのデータでテストを実行。
これをm回繰り返し、それらの誤り率（平均二乗誤差）の平均を予測値とする。
→平均を取るので１回のみよりも結果が安定する。

ロジスティック回帰
・ロジスティック回帰
入力からクラスに分類する問題。
入力はm次元のベクトル、出力は０または１を想定。
・ロジスティック線形回帰モデル
分類問題を解くための機械学習モデル。
入力からそのラベルを予測するシステムを構築。
入力とパラメータの線形結合をシグモイド関数に入力する。出力はｙ＝１となる確率となる。
・シグモイド関数

$$\sigma(x)=\frac{1}{1+\exp(-ax)}$$

a を増加させるとｘ＝０付近での曲線の勾配が増加する。
シグモイド関数の微分はシグモイド関数自身で表現することが可能。

$$\frac{\partial\sigma(x)}{\partial x}=a\sigma(x)(1-\sigma(x))$$

最尤推定
・尤度
あるデータを得たときに、分布のパラメータが特定の値であることがどれほどありえそうか（もっともらしいか）を表現。
尤度（データを固定してパラメータが変化）⇔確率（パラメータを固定してデータが変化）
・尤度関数
確率変数はベルヌーイ試行に従う。

$$P(Y=1\mid x)=p$$
$$P(Y=0\mid x)=1-P(Y=1\mid x)=1-p$$

としたとき、Y＝ｔとなる確率は以下のように表される。

$$P(Y=t\mid x)=P(Y=1\mid x)^{t}P(Y=0\mid x)^{1-t}$$
$$=p^{t}(1-p)^{1-t}$$

・同時確率
観測されたデータ（学習データ）を発生させるもっともらしい確率分布を求める。

$$P(Y=y_1\mid x_1)P(Y=y_2\mid x_2)\cdots P(Y=y_n\mid x_n)x)$$
$$=p_1^{y1}(1-p)^{1-y1}p_2^{y2}(1-p)^{1-y2}\cdots p_n^{yn}(1-p)^{1-yn}$$
$$=\prod_{i=1}^{n}p_n^{yi}(1-p)^{1-yi}$$
$$=L(w_0、w_1、\cdots w_m)\cdots(※)$$

（※）ロジスティック回帰モデルより

$$P(Y=1\mid x_1)=p_1=\sigma(w_0+w_1x_{11}+\cdots+w_mx_{1m})$$
$$P(Y=2\mid x_2)=p_2=\sigma(w_0+w_1x_{21}+\cdots+w_mx_{2m})$$
$$\vdots$$
$$P(Y=n\mid x_n)=p_n=\sigma(w_0+w_1x_{n1}+\cdots+w_mx_{nm})$$

・対数尤度関数
計算が面倒なので対数で書く。
平均二乗誤差は「最小化」、尤度関数は「最大化」では話がややこしい
→対数尤度関数にマイナスをかけて「最小化」で統一

$$E(w_0、w_1、\cdots w_m)$$
$$= -logL(w_0、w_1、\cdots w_m)$$
$$= \sum_{i=1}^{n} \{ t_n logpi + (1 - t_n) log(1 - pi)$$

勾配降下法
・勾配降下法の必要性
高次元となると、損失関数を微分して0になる値を解析的に求めることが難しい
→近似的に解を求める必要
・勾配降下法

$$w^{k+1} = w^k - \eta \frac{\partial E(w、b)}{\partial w}$$
$$b^{k+1} = b^k - \eta \frac{\partial E(w、b)}{\partial b}$$

η：学習率（ハイパーパラメータ）
対数尤度関数を最適化するため、係数（w）とバイアス（b）に関して微分する。

$$\frac{\partial E(w、b)}{\partial w} = \sum_{i=1}^{n} \frac{\partial E_i}{\partial p_i} \frac{\partial p_i}{\partial w}$$
$$= -\sum_{i=1}^{n} \left(\frac{t_i}{p_i} - \frac{1 - t_i}{1 - p_i} \right) \frac{\partial p_i}{\partial w}$$
$$= -\sum_{i=1}^{n} \left(\frac{t_i}{p_i} - \frac{1 - t_i}{1 - p_i} \right) p_i (1 - p_i) x_i \quad \cdots シグモイド関数の微分$$
$$= -\sum_{i=1}^{n} \{ t_i (1 - p_i) - p_i (1 - t_i) \} x_i$$
$$= -\sum_{i=1}^{n} (t_i - p_i) x_i$$
$$\frac{\partial E(w、b)}{\partial b} = -\sum_{i=1}^{n} (t_i - p_i)$$

勾配を逐次的に求める式は以下の通り。

$$w^{k+1} = w^k + \eta \sum_{i=1}^{n} (t_i - p_i) x_i$$
$$b^{k+1} = b^k + \eta \sum_{i=1}^{n} (t_i - p_i)$$

計算量大→勾配降下法のデメリット
ディープラーニングでは確率的勾配降下法（SGD）で回避

モデルの評価
・混同行列

		検証用データの結果	
		生存	死亡
モデルの予測結果	生存	TruePositive	FalsePositive
	死亡	TrueNegative	FalseNegative

・正解率

$$\frac{TP + TN}{TP + FN + FP + TN}$$

・適合率
見逃しが多くても正確な予測がしたい場合に使用。

$$\frac{TP}{TP + FP}$$

・再現率
誤りが多少多くても抜け漏れを少なくしたい場合。

$$\frac{TP}{TP + FN}$$

・F値
適合率と再現率の調和平均。

主成分分析
・多変量データの持つ構造をより小数個の指標にまとめる（大きな次元のものを低次元に圧縮）

変量の個数を減らすことに伴う情報の損失はなるべく少なくする必要
→学習データの分散が最大となる方向への線形変換を求める
・係数ベクトルを変えると線形変換後の値が変わる

$s_j = (s_{1j}, s_{2j} \cdots s_{nj})^T = \overline{X} a_j$ とおくと分散は

$$Var(s_j) = \frac{1}{n} s_j^T s_j$$

$$= \frac{1}{n} (\overline{X} a_j)^T (\overline{X} a_j)$$

$$= \frac{1}{n} a_j^T \overline{X}\,\overline{X} a_j$$

$$= a_j^T Var(\overline{X}) a_j$$

以下の最適化問題となる。

$$\underset{a \in \mathbb{R}^m}{argmax}\ a_j^T Var(\overline{X}) a_j$$

$$subject\ to\ a_j^T a_j = 1$$

この問題を解くために、以下のラグランジュ関数を置き、係数ベクトル a j で微分する。

$$E(a_j) = a_j^T Var(\overline{X}) a_j - \lambda(a_j^T a_j - 1)$$

$$\frac{\partial E(a_j)}{\partial a_j} = 2 Var(\overline{X}) a_j - 2\lambda a_j = 0$$

$$Var(\overline{X}) a_j = \lambda a_j$$ ・・・解は分散共分散行列 $Var(\overline{X}) = \frac{1}{n} \overline{X}\,\overline{X}$ の固有ベクトル

分散共分散行列は実対象行列であるため、固有ベクトルはすべて直交となる。
・主成分
第１主成分：最大固有値に対応する固有ベクトルで線形変換された特徴量
第ｋ主成分：ｋ番目の固有値に対応する固有ベクトルで線形変換された特徴量
・寄与率
第ｋ主成分の寄与率：第ｋ主成分の分散の全分散に対応する割合

ｋ近傍（ｋＮＮ）法
・分類問題のための機械学習法。
・データから近い順にｋ個のデータを見て多数決で所属クラスを決定する。
・ｋを変化させると結果も変わる。ｋを大きくすると決定境界が滑らかになる。

ｋ平均（ｋ－ｍｅａｎｓ）法
・教師なし学習のクラスタリング手法。与えられたデータをｋ個のクラスタに分類。
・クラスタの中心をランダムに生成。
各データと各クラスタ中心との距離を計算し最も距離が近いクラスタに各データを所属させる。
各クラスタの平均ベクトル（中心）を計算して中心を重心の位置にずらす。
クラスタの再割り当てを行い中心の更新を行う。
以上をクラスタの中心が変化しなくなるまで行う。