# DL講座 E資格対策補足資料

## マルチヌーイ分布
マルチヌーイ分布とは、カテゴリカル分布のことである。ベルヌーイ分布が2値の確率変数が従う離散分布であるのに対し、カテゴリカル分布はk個の確率変数が従う離散分布である。k=2の時、ベルヌーイ分布となる。

## 深層学習ライブラリ
DNN を実装するためのライブラリは、その特性から以下の二つの方式に大別される。
　define-and-run：ネットワークアーキテクチャの構成を先に行ったあとに順伝播処理を行う方式
　define-by-run： 順伝播処理を行いながらネットワークアーキテクチャの構成を順に行う方式
define-by-run 方式で著名なライブラリに、PreferredNetworks 社（PFN）のChainerがある。なお、PFNは2019 年12 月5 日、Chainer のフレームワーク開発を終了してメンテナンスフェーズ へ移行し，自社開発はChainer からFacebook が主導するPyTorchに順次移行することを発表した。

## 教師あり学習などの種類
Supervised learningは、ラベル付きデータを使って学習させる方法を指す。教師あり学習と訳される。
Unsupervised learningは、ラベルなしデータを使って学習させる方法を指す。半教師なし学習と訳される。
Semi-supervised learningは、大量のラベルなしデータと少量のラベル付きデータを使って学習させる方法を指す。半教師あり学習と訳される。
Weakly supervised learning(weak supervisionともいう)は、弱い教師がついたデータで教師あり学習を行う方法を指す。弱教師あり学習と訳される。*1 *2
Self supervisedは、データ自身の中に教師を設定して学習させる方法を指す。自己教師あり学習と訳される。ラベルを与えていないため、ある種の教師なし学習であるが、従来の教師なし学習で行う特徴抽出やクラスタリングと区別したいときにこの言葉が使われることが多い。
*1 https://emtiyaz.github.io/aip_iith_workshop_2019/slides/Sugiyama.pdf
*2 https://www.slideshare.net/MLSE/ss-97568525

## タスクT、性能指標P、経験E
Tom M. Mitchellは、1997年、機械学習における学習の意味を次のように簡潔に定義し

た。
「あるコンピュータプログラムが性能指標Pで測定されるタスクTに関する性能を経験Eにより改善した場合、そのコンピュータプログラムはタスクTおよび性能指標Pに関する経験Eから学習したと言える」(意訳)
出典：Tom M. Mitchell,Machine Learning,1997
http://profsite.um.ac.ir/~monsefi/machine-learning/pdf/Machine-Learning-Tom-Mitchell.pdf

タスクTは、コンピュータプログラムにやらせたいことである。例えば、上記出典には、「ボード**ゲーム**のチェッカーをする」「手書き文字の画像を認識し分類する」「ビジョンセンサーを用いて4車線高速道路を走行する」などが例として挙げられている。また、Goodfellow氏らの「Deep Learning」では以下が例として挙げられている。
・分類・・・入力データがk 個のカテゴリのどれに分類されるかを推定する問題。
・欠損値のある入力の分類・・・分類において、特に入力データに欠損値を含む場合の問題。
・回帰・・・入力データから数値を推定する問題。
・転写・・・構造化されていないデータを変換する問題。文字認識や音声認識等が該当する。
・機械翻訳・・・ある言語のシンボルの系列から別の言語のシンボルの形式に変換する問題。
・構造出力・・・入力データの構造を推定する問題。自然言語処理の構文解析等が該当する。
・異常検知・・・入力データが異常か正常かを推定する問題。
・合成とサンプリング・・・訓練データと類似した新たなデータを生成する問題。音声合成や画像生成等が該当する。
・欠損値補完・・・入力データの欠損値を推定する問題。
・ノイズ除去・・・入力データのノイズを推定する問題。
・密度推定・・・入力データの確率密度（その入力データが得られる確率はどのくらいかを推定する問題。
性能指標Pは、通常タスク毎に設定される。例えば、チェッカーのタスクであれば「勝率」、手書き文字の画像を認識し分類するタスクであれば「正しく分類できた割合」、ビジョンセンサーを用いて4車線高速道路を走行するタスクであれば「エラーが発生するまでの平均走行距離」、である。
経験Eは、データ集合のことである。機械学習では過去に得た経験を用いることが多いが、強化学習のように学習の過程で経験を得る場合もある。例えば、チェッカーのタスクであれば「自分自身と対戦した際の記録」、手書き文字の画像を認識し分類するタスクであれば「正解クラスがついた手書き文字のデータセット」、ビジョンセンサーを用いて4車線高速道路を走行するタスクであれば「人間が運転している時のハンドル捌きと一連の画像」、である。

[Ian Goodfellow, DeepLearning, 5.1節参照]

## 過剰適合、過少適合
過剰適合とは、訓練データに対する誤差（訓練誤差）が未知データに対する誤差(汎化誤差)に対して低い状態を表す。過適合や過学習とも呼ばれる。過剰適合を抑える方法として、正則化が有効。
過少適合とは、訓練誤差が大きい状態を表す。この場合、汎化誤差も大きくなる。未学習とも呼ばれる。
[Ian Goodfellow, DeepLearning, 5.2節参照]

## ハイパーパラメータ
ハイパーパラメータとは、学習を行う前に分析者が指定するパラメータのことである。機械学習アルゴリズムの内部では調整されない。
[Ian Goodfellow, DeepLearning, 5.3節参照]

## 訓練データ、検証データ、テストデータ
訓練データ、検証データ、テストデータという言葉は、一般的に以下のように使い分けられる。
　訓練データ・・・学習するために用いるデータ。学習用データから取り出される。
　検証データ・・・ハイパーパラメータ等の調整をするために用いるデータ。学習用データから取り出される。
　テストデータ・・・完成したモデルの汎化性能を推定するために用いるデータ。学習用データとは別物。
[Ian Goodfellow, DeepLearning, 5.3節参照]

## ホールドアウト法
汎化性能を推定するための方法。学習用データを訓練データとテストデータに分けて、訓練データで学習した後、テストデータで誤差を算出する。この誤差を汎化誤差の推定値として扱う。

## 交差検証法
汎化性能を推定するための方法。学習用データをk個の集合に分割し、テストデータ役を入れかえながら、k個分の汎化誤差を算出する。その平均誤差を汎化誤差の推定値として扱う。k分割交差検証法ともいう。

[Ian Goodfellow, DeepLearning, 5.3節参照]

## 2乗誤差を最小化することと尤度を最大化することの等価性

ある回帰問題において、2乗誤差を最小化することは、出力が正規分布に従うと仮定した元で尤度を最大化することと等価である。このことは、以下の手順で確認できる。

変数 $(x, y)$ に対応する $n$ 個のデータの組 $(x_i, y_i)$ （ただし，$i = 1, \ldots, n$ とする）が与えられたとき，関数 $f(x)$ によって $y$ を推定する問題を最小二乗法によって解くことは，二乗和誤差 $E$ を最小化する関数 $f$ を求めるものとして定式化できます．このとき，二乗和誤差は

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

で定義されます

また，$y_i$ が平均 $f(x_i)$，分散 $\sigma^2$ の正規分布にしたがうということを確率密度関数によって表せば，

$$p(y_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(y_i - f(x_i))^2}{2\sigma^2}\right\}$$

です

このとき，全データに対する対数尤度関数は

$$\begin{aligned}\log\prod_{i=1}^{n}p(y_i) &= \sum_{i=1}^{n}\log p(y_i)\\ &= -n\log\left(\sqrt{2\pi}\sigma\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))^2\\ &= C - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))^2\end{aligned}$$

です　　ただし，$C = -n\log\left(\sqrt{2\pi}\sigma\right)$ と置き直しています．ここで，分散 $\sigma^2$ はデータに依らず一定と仮定しているため，上式の最大化問題は

$$\sum_{i=1}^{n}(y_i - f(x_i))^2$$

の最小化問題に等価です

## 次元の呪い

データの次元が高いと、機械学習の問題は解決が極めて難しくなることが多い。この現象は次元の呪いと呼ばれる。次元の呪いという言葉は、Richard E. Bellman著のDynamic Programmingで初めて導入された。
次元の呪いを回避する方法としては、主成分分析などを用いた次元削減や使用する特徴の組み合わせを最適化する特徴選択によって，次元数を減らすことが有効である。
参考文献：Dynamic Programming,Richard E. Bellman,1957 https://www.gwern.net/docs/statistics/decision/1957-bellman-dynamicprogramming.pdf

## 性能指標
性能指標とは「精度」や「誤差」といった指標であり、タスクの性能を評価するために用いるものである。回帰問題であれば、MSE、RMSE、MAEがよく用いられる。分類問題であれば、Accuracy、Recall、Precision、F値、AUCなどがよく用いられる。
[Ian Goodfellow, DeepLearning, 11.1節参照]

## グリッドサーチ、ランダムサーチ、ベイズ的最適化
ハイパーパラメータの探索は、手動で行っても良いが、探索の範囲が広いと探索作業が大変になるため自動化したくなる。自動的に探索する方法として、グリッドサーチ、ランダムサーチ、ベイズ的最適化がある。
候補となるハイパーパラメータの組みわせ全てを探索することをグリッドサーチという。これに対して、指定した範囲内において、ランダムに探索点を決めていく方法をランダムサーチという。
探索したいハイパーパラメータの中に性能への影響が小さいハイパーパラメータを含んでいる場合、グリッドサーチよりもランダムサーチの方が効率が良い。
ベイズ的最適化は、モデルに基づくハイパーパラメータ探索手法の1つである。ガウス過程回帰を用いて誤差の期待値と分散を予測し、次の探索点を決めていく。ニューラルネットワークなど学習に時間がかかるモデルで使用されることが多い。
ベイズ的最適化の参考文献：ガウス過程と機械学習 (機械学習プロフェッショナルシリーズ),持橋ら,講談社
[Ian Goodfellow, DeepLearning, 11.4節参照]

## 正則化
正則化とは、「学習時に重みの自由度を制約することにより過適合を抑えようとする方法(深層学習,岡谷,講談社)」のことであり、線形回帰やニューラルネットワーク等でよく用いられる。代表的な正則化として、L1 正則化、L2 正則化がある。これらの正則化項にはそれぞれ、L1ノルム、L2ノルムが用いられる。ノルムとは、ベクトルの長さを表す指標であ

り、n 次元の重みベクトルw のLp ノルム は以下の式で定義される。

$$||\mathbf{w}||_p = \sqrt[p]{|w_1|^p + |w_2|^p + |w_3|^p + \ldots + |w_n|^p}$$

p>=1であり、p=1 の場合をL1 ノルム、p=2 の場合をL2 ノルム、、、と呼ぶ（L はアンリ・ルベーグのLに由来すると言われている）。L1ノルムおよびL2ノルムを正則化項として損失関数に組み込んだ場合、以下の特徴 がある。

• L1 ノルム・・・w_i の絶対値の合計が正則化項の値になる。そのため、正則化項の影響を強くするほど重みの合計量が減っていく。

• L2 ノルム・・・実際に計算する際は2ノルムの2乗で計算するため、w_iの2 乗値の合計が正則化項の値になる。正則化項の影響を強くするほど重みの値が原点に近づいていく。

[Ian Goodfellow, DeepLearning, 7.1節参照]

正則化の仕組みを理解するためのノートブック：Reguralization.ipynb

## サポートベクトルマシン

サポートベクトルマシンの仕組みを理解するためのノートブック：SVM.ipynb