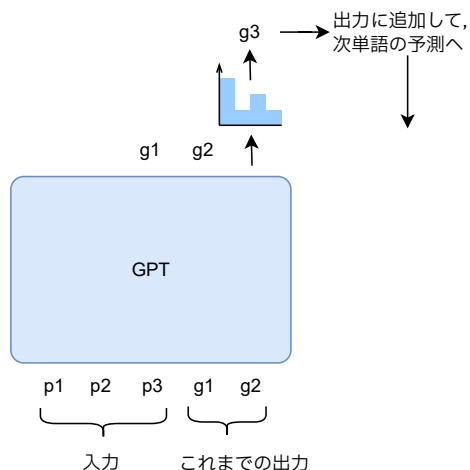


13. 系列データの学習



- 13.1 系列を入力とする問題の分析
- 13.2 系列ラベリング
- 13.3 系列識別
- 13.4 系列変換
- 13.5 時系列信号の予測



- 荒木雅弘：『Pythonではじめる機械学習』（森北出版、2025年）
- スライドとコード

13.1 系列を入力とする問題の分析

- 系列を入力とする問題の分類
 - 入力系列長と出力系列長が等しい
 - 入力の各要素に対して出力ラベルを付与する系列ラベリング問題
 - 例) 形態素解析, 固有表現抽出
 - 入力系列長に関わらず出力系列長が1
 - 不定長の系列全体を1つのクラスに対応させる系列識別問題
 - 例) 動画像の分類, 文書分類
 - 入力系列長と出力系列長の間に, 明確な対応関係がない
 - 入力系列を出力系列に変換する系列変換問題
 - 例) 連続音声認識, 機械翻訳, 時系列信号の予測

13.2 系列ラベリング (1/6)

- 系列ラベリング問題とは
 - 入力系列の個々の要素に対して出力ラベルを付与する
 - 個々の出力ラベルの出現確率は、前後の出力ラベル系列に依存
 - 1入力1出力の識別器を連続的に適用する方法では性能が低い
 - ⇒ 入力や出力の系列としての特徴を使う
 - 可能な出力ラベル系列の数は膨大
 - すべての出力ラベル系列をリストアップすることは現実的に不可能
 - ⇒ 探索によって（準）最適解を求める

13.2 系列ラベリング (2/6)

- 系列ラベリング問題の事例

- 形態素解析

入力 系列 で 入力 さ れる 各 要素

出力 名詞 助詞 名詞 動詞 接尾辞 接頭辞 名詞

- 固有表現抽出

- B: begin, I: inside, O: outside

入力 サム は 浅葱 町 で プリウス を 見た

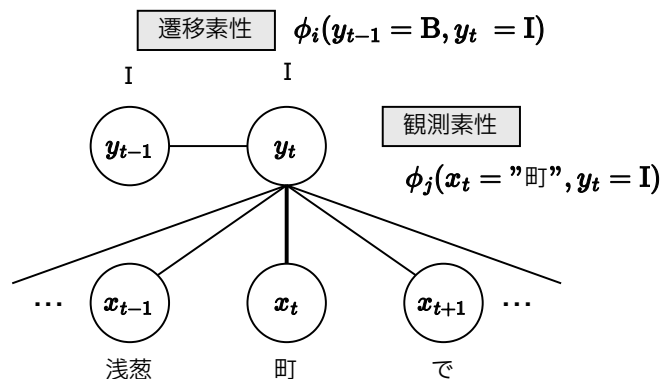
出力 B-Per O B-Loc I-Loc O B-Prod O O

13.2 系列ラベリング (3/6)

- CRF (conditional random field) による系列ラベリング

- 素性関数の導入

- 入力系列 \mathbf{x} と出力系列 \mathbf{y} との間に定義される関数 $\phi(\mathbf{x}, \mathbf{y})$
 - 観測素性は \mathbf{x} と \mathbf{y} の特定の要素間の関係、遷移素性は \mathbf{y} の隣接する要素間の関係を表す
 - 素性は明確に定義できるものであれば、品詞や文字種など、要素でなくてもよい
- ϕ で定義する関係が成立すれば1, 不成立なら0を値とする



13.2 系列ラベリング (4/6)

- 対数線型モデル（多クラスロジスティック回帰）
 - \mathbf{w} の学習は、動的計画法を導入した勾配降下法によるクロスエントロピーの最小化

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z} \exp(\mathbf{w}^\top \phi(\mathbf{x}, \mathbf{y})), \quad \text{ただし } Z = \sum_{\mathbf{y}} \exp(\mathbf{w}^\top \phi(\mathbf{x}, \mathbf{y}))$$

- 出力系列の決定法

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}) = \arg \max_{\mathbf{y}} \frac{1}{Z} \exp(\mathbf{w}^\top \phi(\mathbf{x}, \mathbf{y})) = \arg \max_{\mathbf{y}} \mathbf{w}^\top \phi(\mathbf{x}, \mathbf{y})$$

- 遷移素性を隣接するものに限定すると、ビタビアルゴリズムによって探索が可能

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} \left(\sum_t \mathbf{w}^\top \phi(\mathbf{x}, y_t, y_{t-1}) \right)$$

13.2 系列ラベリング (5/6)

- ビタビアルゴリズム

- すべての出力記号について先頭の要素 (y_1) についてのスコアを計算

$$\alpha(1, y_1) = \mathbf{w}^\top \phi(\mathbf{x}, y_1)$$

- $t = 2$ から始めて, 最後の出力要素に至るまで以下を繰り返し

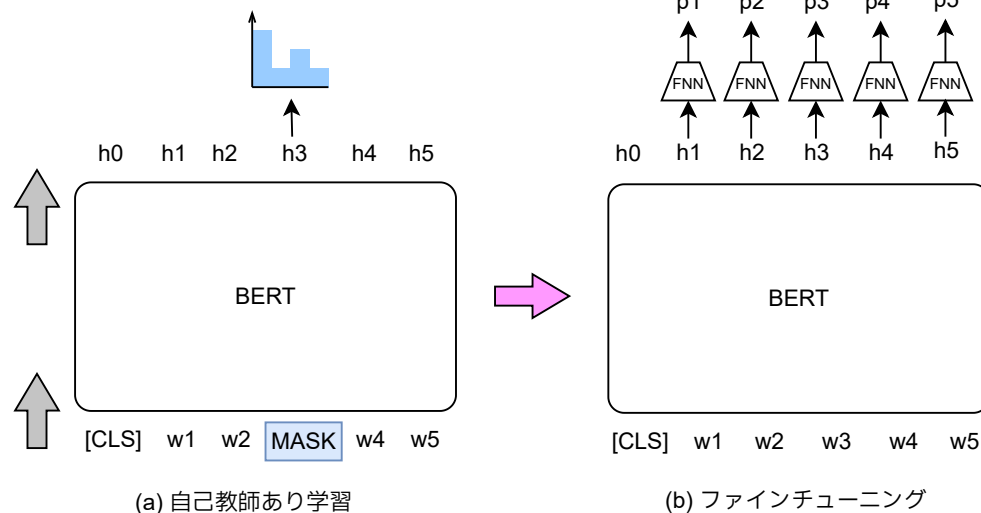
$$\alpha(t, y_t) = \max_{y_t} \{ \mathbf{w}^\top \phi(\mathbf{x}, y_t, y_{t-1}) \}$$

$$B(t, y_t) = \arg \max_{y_{t-1}} \{ \mathbf{w}^\top \phi(\mathbf{x}, y_t, y_{t-1}) \}$$

- 最終要素での最大値に対応する B を先頭まで遡る

13.2 系列ラベリング (6/6)

- BERT による系列ラベリング
 - 自己教師あり学習で事前学習を行った BERT を用いて、入力系列全体の文脈情報を考慮した各要素の特徴ベクトルを取得
 - 各要素の特徴ベクトルを入力として、出力ラベルを予測する FNN を学習



13.3 系列識別 (1/5)

- 生成モデルによるアプローチ
 - 事後確率 $P(y \mid \boldsymbol{x})$ が最大となるクラス \hat{y} を識別結果とする
 - ベイズの定理を用いて事後確率を尤度と事前確率の積に変形する
 - 系列識別問題ではクラスの事前確率 $P(y)$ が得られることが多い
 - クラスごとの \boldsymbol{x} の生成確率を推定するモデル $P(\boldsymbol{x} \mid y)$ を構築する

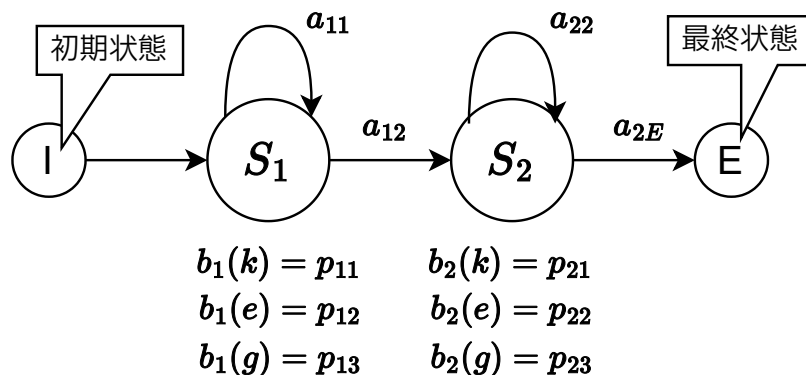
$$\hat{y} = \arg \max_y P(y \mid \boldsymbol{x}) = \arg \max_y P(\boldsymbol{x} \mid y)P(y)$$

13.3 系列識別 (2/5)

- 系列識別問題の事例
 - PC操作系列による熟練度の判定
 - k: キーボード, g: マウス, e: エラー
 - 初心者の入力系列例
 - k e k g k e k g g k g k k e g e e k e e e g e
 - キーボード, マウスの操作が行き来し, 後半になるほど疲労してエラーが多い
 - 熟練者の入力系列例
 - k k e k g k k k e k g k g g g e g k g
 - キーボード, マウスの操作が比較的集中して効率が良く, エラーが少ない
 - 識別したい入力系列
 - k g e k g k k g e k g e k e e k e g e k
 - これは初心者／熟練者のどちらか

13.3 系列識別 (3/5)

- 不定長入力に対する生成確率の計算法
 - 自己遷移を持つ確率オートマトンを用いる
 - I: 初期状態, E: 終了状態
 - 確率計算はビタビアルゴリズムを用いる



13.3 系列識別 (4/5)

- HMMの学習：EMアルゴリズム

- Eステップ

- 現在のHMMのパラメータで、隠れ変数のすべての取り得る値について $p(\mathbf{x})$ を計算

- Mステップ

- 隠れ変数が取り得る値全てについてHMMのパラメータを最尤推定し、 $p(\mathbf{x})$ を重みとして加算

E ステップ

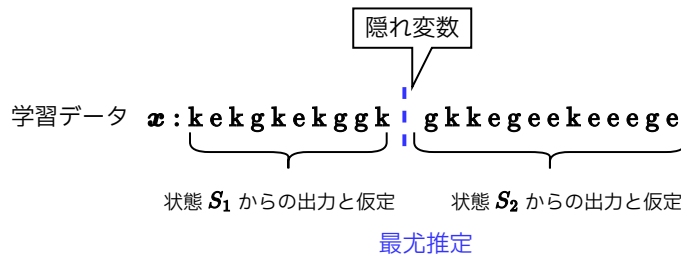
現在の HMM のパラメータで、
隠れ変数のすべてのとりうる値
について $p(\mathbf{x})$ を計算

繰り返す

M ステップ

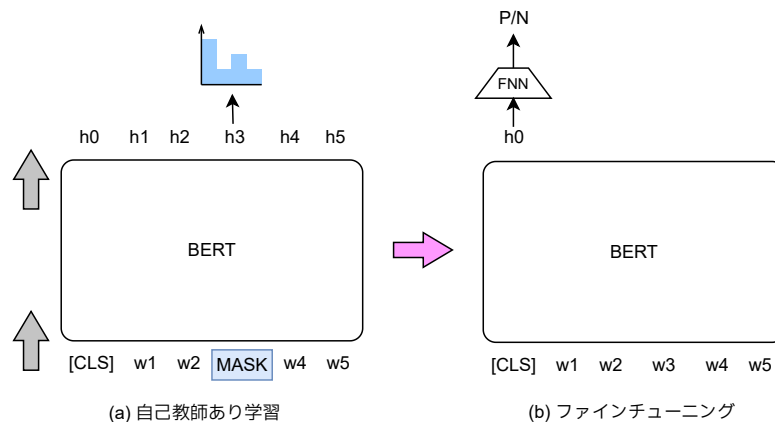
隠れ変数のすべてのとりうる
値について、HMM のパラメータ
を最尤推定し、 $p(\mathbf{x})$ を重みとし
て足し合わせる

確率計算 $p(\mathbf{x}) = b_1(k)a_{11}b_1(e)a_{11}\cdots b_1(k)a_{12}b_2(g)a_{22}\cdots b_2(e)$



13.3 系列識別 (5/5)

- ニューラルネットによる系列識別
 - 双方向 RNN による系列識別
 - 前向き・後向きそれぞれの最終状態を結合
 - 対象タスクのデータが多い場合に有効
 - BERT による系列識別
 - 出力系列の平均や、先頭の [CLS] トークンの特徴ベクトルを入力として、出力クラスを予測する FNN を学習



13.4 系列変換 (1/2)

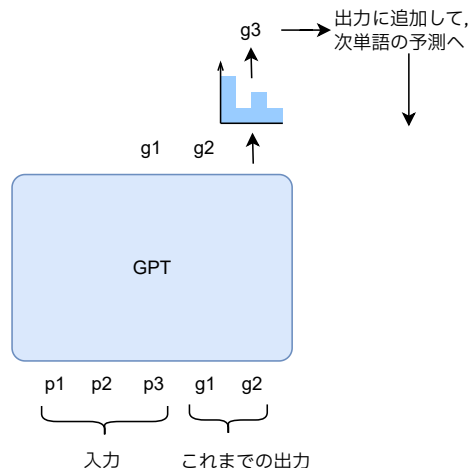
- 系列変換問題の定式化
 - 入力系列 $\mathbf{x} = x_1, \dots, x_T$
 - 出力系列 $\mathbf{y} = y_1, \dots, y_L$

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x})$$

- RNN（エンコーダーデコーダモデル）による系列変換
 - 入力系列 \mathbf{x} をエンコードして文脈ベクトル \mathbf{c} を取得
 - \mathbf{c} をデコードして出力系列 \mathbf{y} を生成（ \mathbf{x} と \mathbf{y} の長さが異なっても対応可能）
 - 学習は入出力事例を与えて、出力系列のクロスエントロピーを最小化する End-to-End アプローチ

13.4 系列変換 (2/2)

- GPT による系列変換
 - 自己回帰型モデルを用いて、出力系列を1要素ずつ生成し、文末記号の生成で終了
 - 変換内容の指示を入力系列の先頭に付与することで、さまざまな変換タスクに対応可能
 - 例) 翻訳, 要約, コード生成
 - LLM を質問応答データでファインチューニングしたものが, Instruction モデル



13.5 時系列信号の予測 (1/4)

- 時系列信号をモデル化するための要素
 - 自己相関
 - k 時点離れたデータとの関係性を示し, これを k 次の自己相関と呼ぶ
 - トレンド
 - 全体の増減の傾向を指し, その増減の量はドリフトとよばれる
 - 季節性
 - 周期性とほぼ同義で, 春夏秋冬による変化の傾向が典型例
 - 外因
 - 突発的なイベントなど, 過去の値以外の変数をモデルに組み込むことを指す
 - ノイズ
 - データを上記の要因で説明しても, なお残る変動部分を指す

13.5 時系列信号の予測 (2/4)

- 時系列信号を定常過程と仮定した場合の予測モデル
 - 定常過程：平均値と分散が観測時刻によらず一定の時系列信号 y_1, \dots, y_n
 - AR (Auto Regressive) モデル
 - 時刻 t の値 y_t を, p 個前までの値の重み付き和と誤差で表現するモデル

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t$$

- MA (Moving Average) モデル
 - 時刻 t の値 y_t を, q 個前までの誤差の重み付き和と現在の誤差との差で表現するモデル

$$y_t = \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q}$$

- ARMAモデル：ARモデルとMAモデルを組み合わせたもの

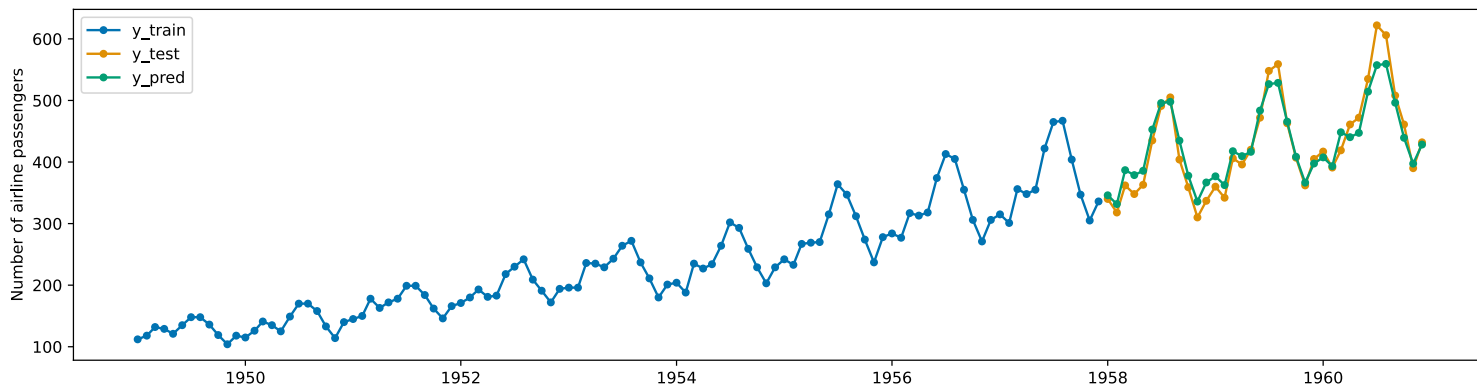
$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q}$$

13.5 時系列信号の予測 (3/4)

- 時系列信号を非定常過程と仮定した場合の予測モデル
 - 平均値が時間的に変動する時系列信号 y_1, \dots, y_n
 - 変動をモデル化するために時系列の階差を計算する
 - 1次階差 $x_t = y_t - y_{t-1}$
 - 2次階差 $z_t = x_t - x_{t-1}$
 - ...
 - ARIMA (Auto Regressive Integrated Moving Average) モデル
 - d 次階差をとった時系列信号に対する p 次のAR, q 次のMAモデル
 - 平均値の変動がなくなるまで d を大きくした時系列信号に対してARMAモデルを適用する
 - ハイパーパラメータ d, p, q は自動調整可能

13.5 時系列信号の予測 (4/4)

- 時系列機械学習用ライブラリ sktime
 - 時系列信号に対する予測・識別・回帰・クラスタリングをサポート
 - sktime の ARIMA モデルを用いた事例
 - 旅客機の乗客数の予測
 - Airline Passengersデータ：1949年から1960年までの月ごとの飛行機の乗客数
 - 直近36ヶ月のデータをテストデータ，それ以前を学習データとして予測問題を設定



まとめ

- ラベル系列に対する識別問題
 - 入力の系列長と出力の系列長が等しい
 - 識別モデル CRF を用いた手法はビタビアルゴリズムで効率的に最適解を探索
 - BERT を用いたアプローチも有効で、CRF と組み合わせることも可能
 - 入力の系列長に関わらず出力の系列長が1
 - 生成モデル HMM で可変長系列の処理が可能
 - RNN や BERT も有効
 - 入力の系列長と出力の系列長に対応関係がない
 - RNN や GPT が有効
- 時系列信号の予測
 - 自己回帰計数や誤差のインパクトをパラメータとして回帰