

統計的モデリング基礎⑫ ～ベイズモデリング（つづき）と異常検知～

鹿島久嗣
(情報学科 計算機科学コース)

ベイズモデリングの応用

ベイズ的統計モデリングの考え方： 最尤推定の尤度の代わりに事後分布を考える

- ベイズ統計では事後分布 $P(\text{パラメータ} | \text{データ})$ を考える

- 事後分布ではパラメータを確率変数と考える

- 事後分布：

$$P(\text{パラメータ} | \text{データ}) = \frac{P(\text{データ} | \text{パラメータ})P(\text{パラメータ})}{P(\text{データ})}$$

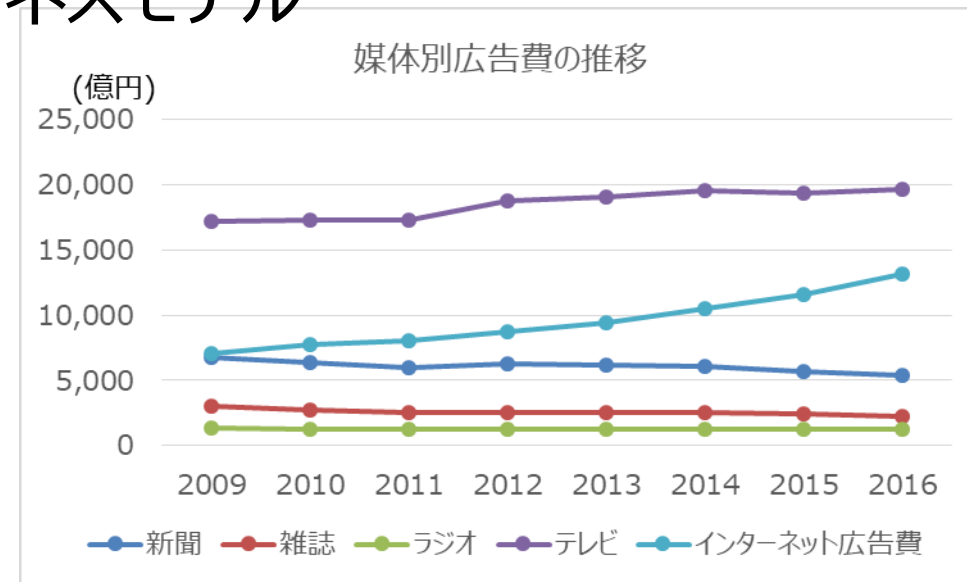
ベイズの定理

- 対数事後分布：

$$\begin{aligned} \log P(\text{パラメータ} | \text{データ}) \\ = \underbrace{\log P(\text{データ} | \text{パラメータ})}_{\text{尤度}} + \underbrace{\log P(\text{パラメータ})}_{\text{事前分布}} + \underbrace{\log P(\text{データ})}_{\text{定数}} \end{aligned}$$

インターネット広告： あらゆる場所に潜むオンライン広告

- インターネット上は広告だらけ
 - Webページ上のバナー広告
 - 検索エンジンのリスティング広告
- 多くのWeb系企業の主要なビジネスモデル
 - 広告表示・クリック・成約に応じた課金
 - アフィリエイト



電通「日本の広告費」 http://www.dentsu.co.jp/knowledge/ad_cost/2011/media.htmlより作成

ネット広告配信の最適化： 適切な広告を 適切な人に届ける

- ネット広告の特徴：
 - 細かい配信調整ができる
 - ◆ ページ閲覧ごとに別の広告を提示できる
 - 効果が測定しやすい
 - ◆ 誰が広告をみて商品を買ったかがデータとして取得できる
- 細かい配信最適化によって広告の効果を最大化する
 - どの広告を提示するべきか（← コレを考えてみる）
 - 誰に広告を提示すべきか
 - どこに広告を出すべきか

提示する広告選択の最適化： 広告効果を推定して効果の高い広告に集中投下

- 広告配信の最適化問題：
 - 100種類の広告から100万回配信する
 - 広告がクリックされる回数を最大化したい
- 各広告の効果（クリック率）は未知
 - 実際に広告を配信してみないと広告の効果はわからない
- 推定と利用のトレードオフ
 1. 推定：すべての広告をまんべんなく配信し、クリック率を推定
 2. 利用：クリック率が高いものを集中的に配信の両者をバランスよく行う必要がある



バンディット問題： 広告配信最適化問題のモデル

- バンディット問題：
 - 当たりの確率が不明な複数のスロットマシンがある
 - 1回につきひとつのスロットマシンを選んでプレイ
 - 当たり回数を最大化したい
- 広告配信最適化問題はバンディット問題として定式化できる
- ポイント：
推定と利用のバランスをいかにとるか



http://en.wikipedia.org/wiki/File:Las_Vegas_slot_machines.jpg

バンディット問題のモデル化：

各スロットマシンが未知の当たり確率をパラメータとしてもつ

- m 台のスロットマシンがある
- スロットマシン i の当たり確率は θ_i （未知）とする
 - つまり、スロットマシンの当たりをベルヌイ分布としてモデル化する
- データ：スロットマシン i をこれまでに n_i 回プレイして h_i 回当たった
- 次にどのマシンをプレイすべきかを決定する問題



スロットマシン1
当たり確率 θ_1



スロットマシン2
当たり確率 θ_2

...



スロットマシン m
当たり確率 θ_m

バンディット問題の問題： 推定精度のばらつき

- 当たり確率の最尤推定値は $\hat{\theta}_i = \frac{h_i}{n_i}$ なので、 $\hat{\theta}_i$ が最大のスロットマシンをプレイすればよいだろうか？
- でも、当たり確率の推定値が同じでも、過去のプレイ回数が多い（少ない）ほど推定値の信頼度は高い（低い）はず...



?

=



$$\frac{1}{2} = 50\%$$

$$\frac{5}{10} = 50\%$$



?

>



$$\frac{2}{3} = 66\%$$

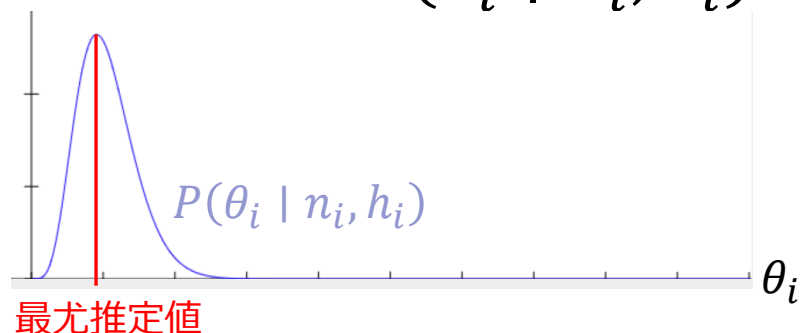
$$\frac{10}{20} = 50\%$$

当たり確率のベイズモデリング： 推定精度を考慮した当たり確率の推定

- 当たり確率の推定値のばらつきをベイズモデリングによって考慮する
- 当たり確率の事後分布は $P(\theta_i | n_i, h_i)$
 - マシン i をこれまでに n_i 回プレイして h_i 回当たったという状況
 - 事後分布の広がり具合が推定の曖昧さに対応
- 例によってベイズの定理を用いると：

$$P(\theta_i | n_i, h_i) = \frac{P(n_i, h_i | \theta_i)P(\theta_i)}{\int_0^1 P(n_i, h_i | \theta_i)P(\theta_i)d\theta_i}$$

事前分布



当たり確率の事前分布：

ベータ分布はベルヌイ分布の共役事前分布

- 事前分布をベータ分布とする：

$$P(\theta_i) = \frac{1}{B(\alpha, \beta)} (\theta_i)^{\alpha-1} (1 - \theta_i)^{\beta-1}$$

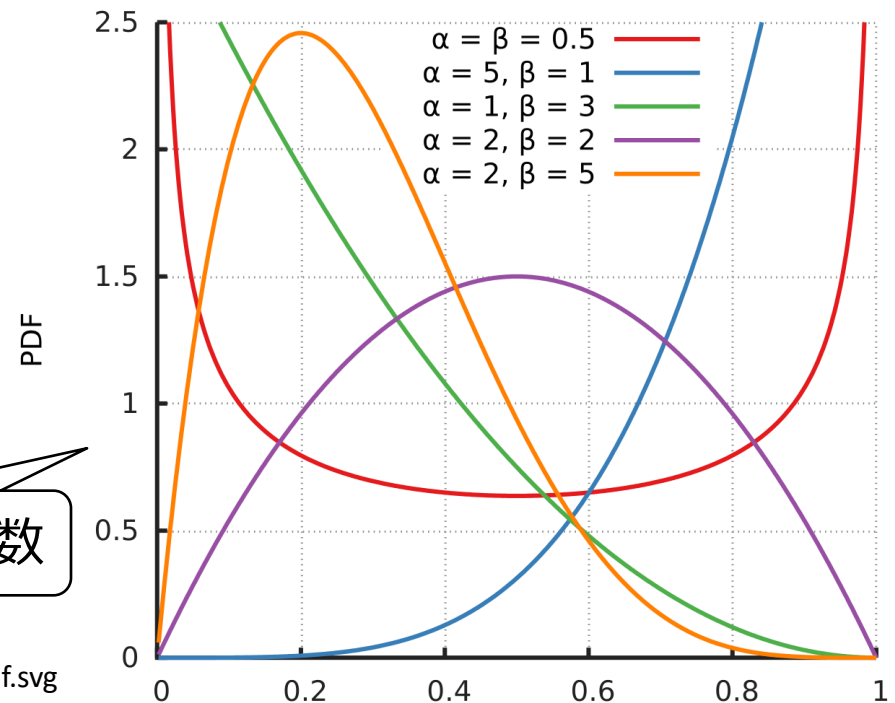
ディリクレ分布で
 $k = 2$ の場合に相当

- ベータ関数 $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$

- 分散は $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

(α, β が大きくなると小さくなる)

確率密度関数



https://en.wikipedia.org/wiki/Beta_distribution#/media/File:Beta_distribution_pdf.svg

当たり確率の事後分布： 事後分布もベータ分布になる

- 事後分布はベータ分布になる（共役事前分布）：

$$P(\theta_i | n_i, h_i) = \frac{1}{B(h_i + \alpha, n_i - h_i + \beta)} (\theta_i)^{h_i + \alpha - 1} (1 - \theta_i)^{n_i - h_i + \beta - 1}$$

- 導出：

$$\begin{aligned} P(\theta_i | n_i, h_i) &\propto P(n_i, h_i | \theta_i) P(\theta_i) \\ &= (\theta_i)^{h_i} (1 - \theta_i)^{n_i - h_i} \frac{1}{B(\alpha, \beta)} (\theta_i)^{\alpha - 1} (1 - \theta_i)^{\beta - 1} \\ &= \frac{1}{B(\alpha, \beta)} (\theta_i)^{h_i + \alpha - 1} (1 - \theta_i)^{n_i - h_i + \beta - 1} \end{aligned}$$

改めて $\frac{1}{B(h_i + \alpha, n_i - h_i + \beta)}$ で正規化する

つぎのマシンの決定：

事後分布から最も当たり確率が高そうなマシンをプレイ

- 戦略：「マシン*i*の当たり確率が最も高い確率」をもとめ、これに従ってマシンを選択する

$$\Pr[P(\theta_i | n_i, h_i) > P(\theta_j | n_j, h_j), \forall j \neq i]$$

- 「スロットマシン*i*の値が最も高い確率」は直接、解析的に評価することは困難

トンプソン抽出：

サンプリングによって次のマシンを選択する

- 「スロットマシン i の値が最も高い確率」は直接評価が困難：
$$\Pr[P(\theta_i \mid n_i, h_i) > P(\theta_j \mid n_j, h_j), \forall j \neq i]$$

- トンプソン抽出：サンプリングで解決する

1. 各マシンの当たり確率の事後分布に基づいて当たり確率 $\{\theta_i\}_{i=1,\dots,m}$ をサンプリングする
2. θ_i が一番大きいスロットマシンを選ぶ
※ ここまでがマシンの決定に関する部分
3. 結果に基づき θ_i の事後分布を更新
※ n_i, h_i が更新されることに伴って事後分布が変わり次の決定に影響する

異常検知

障害や災害：

大規模で複雑なシステムの障害は一旦起こると大損失

生産ラインの故障



コンピュータの
ウィルス感染・不正侵入



ほか、カード不正使用、オレオレ詐欺、監視カメラ,...

監視システム： 障害の予兆を早期発見できれば対処可能



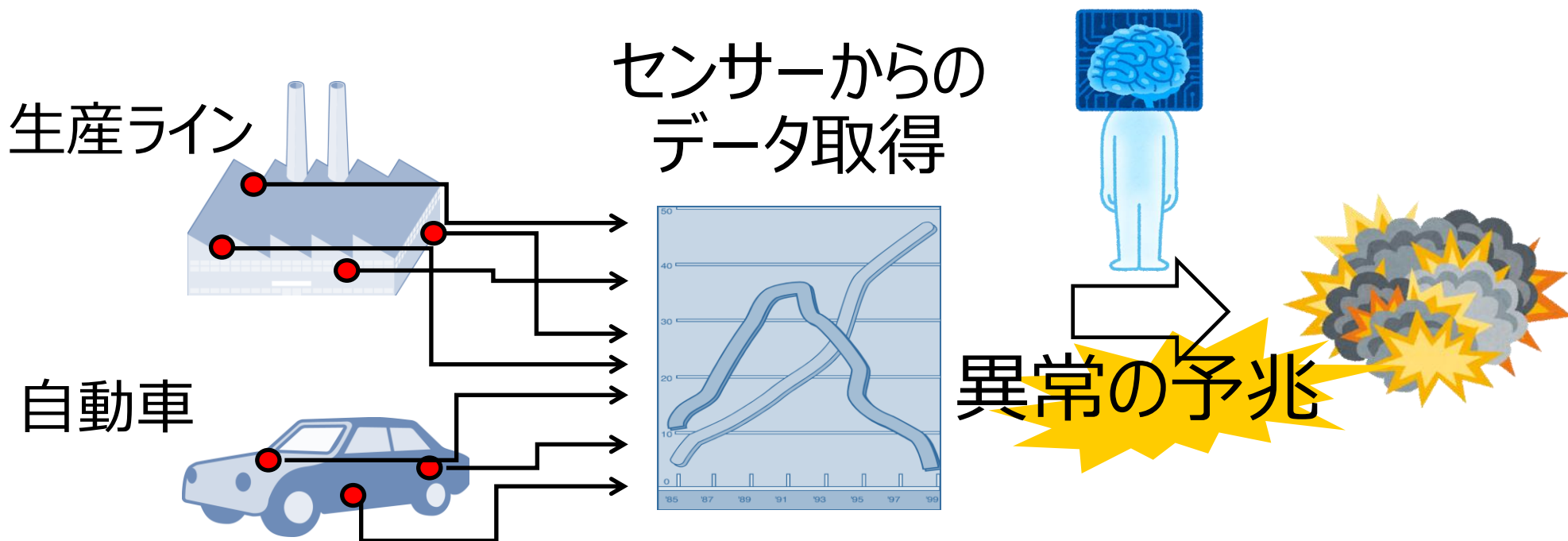
まだ間に合う！



異常検知：

システムの問題を事前に察知し知らせる

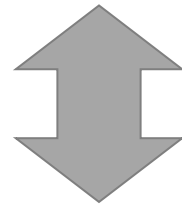
- 設置されたセンサーからの取得データをもとに異常を早期検知したい



異常検知の難しさ：

障害のデータがないため、判別モデルが適用できない

- 障害発生時のデータが集められる場合：
 - 予め検出したいタイプの異常がわかっている場合
 - ある程度の頻度で起こる場合
- 判別モデル（ロジスティック回帰等）が適用可能



- しかし、重大な障害ほど初めて出会うものが多い
- 過去のデータがない（判別モデルが使えない）

異常検知の考え方：

正常時を捉え、そこからの逸脱を見つける

■ 文書や会話：

- 正常：よく出てくる単語・表現
- 異常：めったに出てこない単語・表現

■ センサー：

- 正常：普段の値の範囲
- 異常：普段では出てこない値



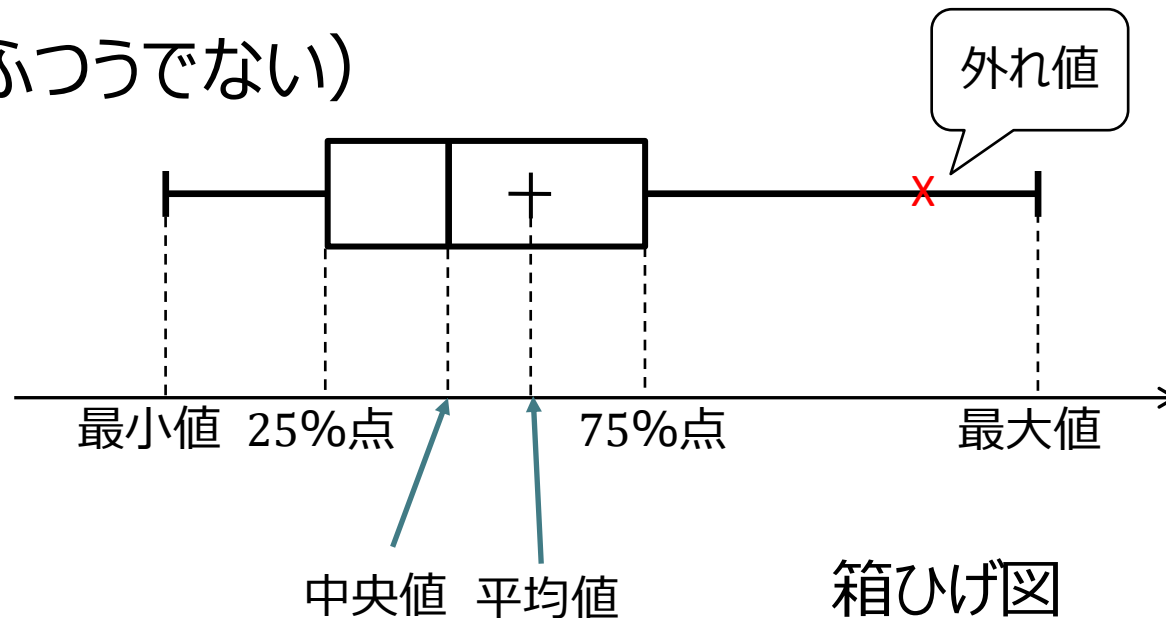
異常検知技術：

「普通でない」データのパターンをみつけたす技術

- 異常：データの中に含まれる「普通でない」パターン
 - 対象のシステムのなんらかの異常を原因として表れる
 - ◆ クレジットカード不正使用・システム侵入・テロ
・システムダウン...
 - あるいは対象システムの状態変化
 - ◆ 新規ニューストピック出現・システムの設定変更
・環境変化...
- これらをデータの中から発見し報告するのが異常検知

異常検知の基本的な考え方： 稀な値を異常と考える

- 単純な場合として1変数（例えば温度）を対象とした異常検知を考える
- 通常値の範囲をとらえる（温度は通常20～50℃の間）
- そこから逸脱した値を検出し、異常として報告する
（80℃はふつうでない）



統計的な異常検知：

モデルからの生成確率が小さいデータを異常と考える

- 正常なデータ x_1, x_2, \dots, x_n が与えられる
- 正常なデータからモデル $p(x)$ を推定する
 - たとえば最尤推定によって（正常時モデルの）パラメータ推定値を得る
- 検証対象のデータ x^{NEW} に対して、モデルが与える確率 $p(x^{\text{NEW}})$ を算出する
- これがある閾値 τ より低ければ $(p(x^{\text{NEW}}) < \tau)$ 異常として報告する

多次元データのモデル： 多次元正規分布の最尤推定

- データが多次元の場合は、たとえば多次元正規分布：

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right)$$

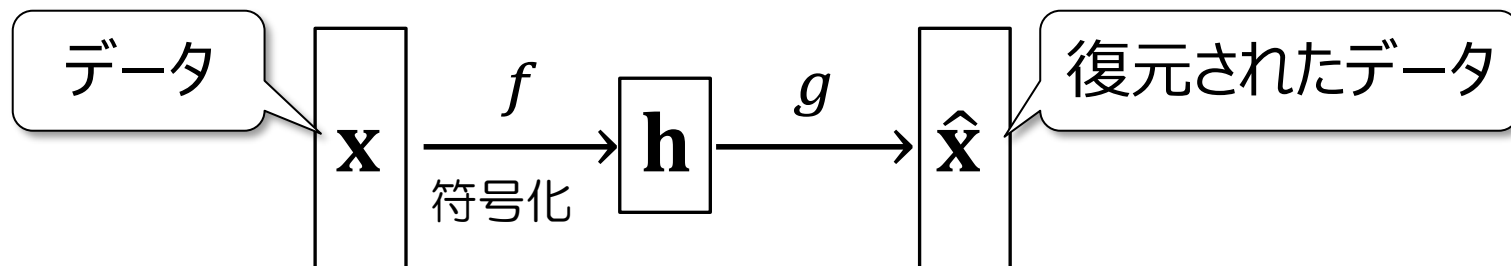
- 正常データ $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$ からパラメータを最尤推定

- 平均： $\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$

- 分散共分散行列： $\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}}) (\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top$

再構成誤差を用いた異常検知の考え方： モデルによる再構成誤差が大きいデータを異常と考える

- 符号化 $\mathbf{h} = f(\mathbf{x})$ と復号化 $\hat{\mathbf{x}} = g(\mathbf{h})$



- 符号化器はデータを低次元に射影する
- 復号化器はもとの空間に戻る
- データ \mathbf{x} が正常なデータなら小さな誤差で復号できるはず
 - 例：ユークリッド距離の場合： $e(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - g(f(\mathbf{x}))\|_2^2$

再構成誤差を用いた異常検知の考え方：

再構成誤差は本質的なデータ表現に載るノイズ

- 低次元空間がデータの本質的な形を捉えているとすると、再構成誤差は「ノイズ」として解釈できる

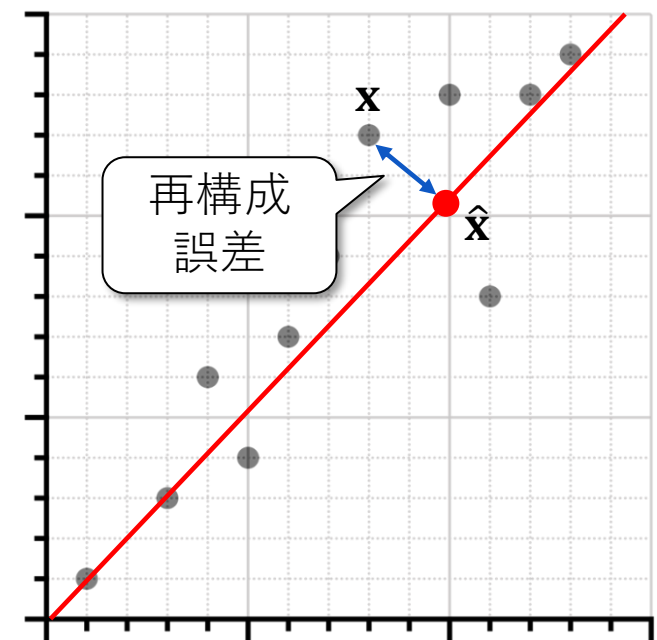
- 大きいノイズを含むものを外れ値とする
- 射影の距離が再構成誤差にあたる

- 再構成誤差 $\| \mathbf{x} - \hat{\mathbf{x}} \|_2^2$ が正規分布：

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \exp \left(-\frac{1}{2} \| \mathbf{x} - \hat{\mathbf{x}} \|_2^2 \right)$$

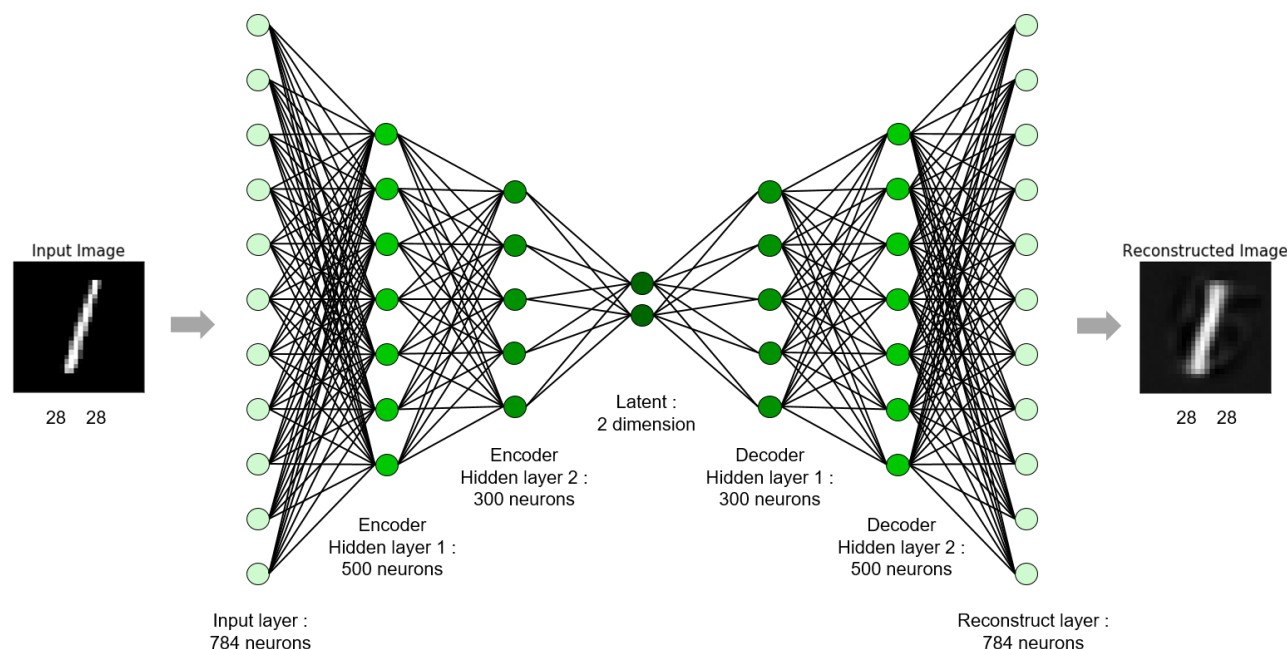
に従うとして $p(\mathbf{x}^{\text{NEW}}) < \tau$ で判定すると
前述の枠組みで解釈できる

2次元データを1次元に射影した例



ニューラルネットによる符号化・復号化： 自己符号化器（オートエンコーダ）

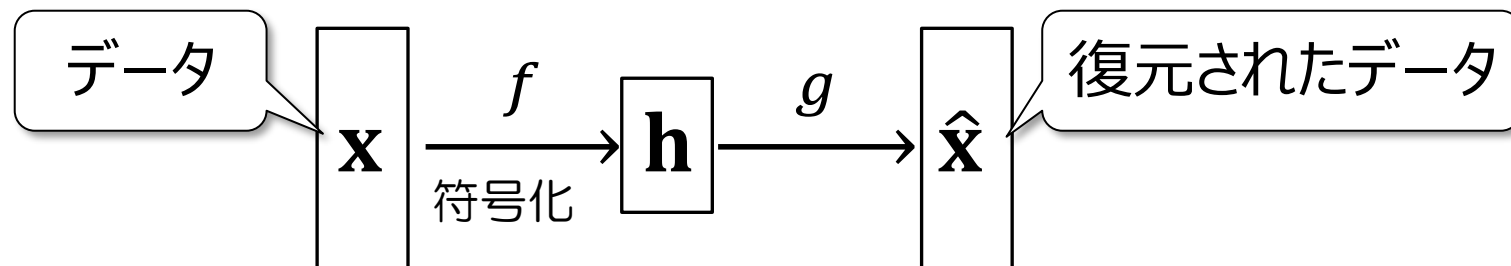
- 符号化器 $\mathbf{h} = f(\mathbf{x})$ と復号化器 $\hat{\mathbf{x}} = g(\mathbf{h})$ をどう実現するか
- 主成分分析（PCA）：符号化 $\mathbf{h} = \mathbf{F}\mathbf{x}$ ・ 復号化 $\hat{\mathbf{x}} = \mathbf{G}\mathbf{h}$ とした場合（ \mathbf{F} と \mathbf{G} は行列、つまり線形変換）
- 自己符号化器は符号化・復号化をニューラルネットでおこなうもの



自己符号化器による異常検知：

自己符号化器の再構成誤差を異常度とする

- 符号化 $\mathbf{h} = f(\mathbf{x})$ と復号化 $\hat{\mathbf{x}} = g(\mathbf{h})$



- 正常なデータから符号化器・復号化器を推定：

$$(f, g) = \operatorname{argmin}_{f, g} \sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$$

- 新たなデータ \mathbf{x}^{NEW} の再構成誤差を異常度とする：

$$\|\mathbf{x}^{\text{NEW}} - \hat{\mathbf{x}}^{\text{NEW}}\|_2^2$$

まとめ：

ベイズモデリング（つづき）と他の話題（異常検知）

■ ベイズモデリングの応用例：

- 広告配信最適化問題はバンディット問題として定式化できる
- ベイズ事後分布によって、当たり（広告への反応）確率の推定のばらつきを考慮する
- トンプソン抽出によって、情報の獲得と利用をバランスした、効率的な決定ができる

■ 異常検知：

- システムの障害は一旦起こると大損失であり早期検出したい
- 異常データは稀なので、正常時を確率モデル化する（例：多次元正規分布）
- 確率モデルからの生起確率の小ささによってデータの異常度を判定