第5章　クラス分類
悪い回答を判別する

西　友佑

想定

- 第３章で述べたQ&Aシステムに回答への評価を付け加えたい
- 回答の評価を自動で判別してくれる機能
- 実際の開発になぞらえて、順序立てて説明

前提

- 人によって良い悪いの意見が別れる
- １００％正しい答えを求めるのは不可能
- 良い回答と悪い回答２つのクラスに分類して判断する

クラス分類を行うには

- データをクラスに分類する（ラベル付け）
- データをどのように表現するかを決める
- 分類器はどのようなモデルかを決める

データの用意

- Stackoverflowの提供するデータを使用
- これには以下の様なデータが含まれる
  - 文書ID
  - 質問か回答か
  - 閲覧回数
  - Etc.
- ラベルとして質問者が回答を受け付けたかどうか（２値）を使用

クラス分類を行うには

- データをクラスに分類する（ラベル付け）
- データをどのように表現するかを決める
- 分類器はどのようなモデルかを決める

データの表現方法は？

- 分類に役立ちそうな特徴量を選択する
  - 回答のID
  - 質問か回答か
  - 投稿日時
  - コミュニティからの評価
  - 本文
  - 回答が受理されたか

回答の良し悪しを定義する

- 回答に対してのスコアが
  - 0より大きければ良い回答
  - 0以下ならば悪い回答

- この法則に基づいて訓練データを作成

クラス分類を行うには

- データをクラスに分類する（ラベル付け）
- データをどのように表現するかを決める
- 分類器はどのようなモデルかを決める

分類器の作成　その1

- 最近傍法を用いて分類器を作成

- どのような特徴量を分類器に入れるのか
  - ソースコード以外のURL（参照）が多ければ良い回答といえるのでは？

- kNN分類器の最適なkの数は？
  - まだ判断できない

分類器の評価

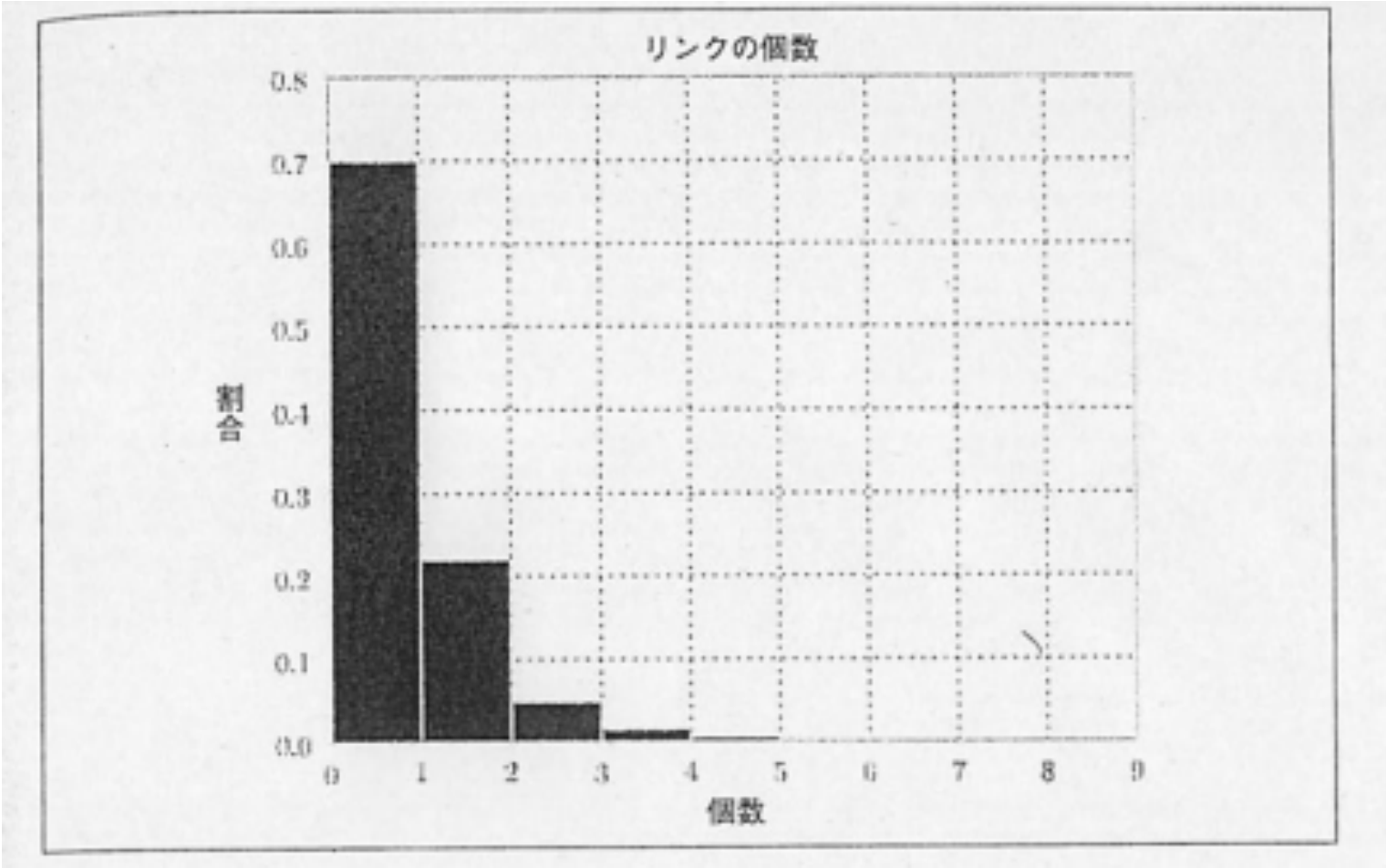
- 交差検定を行い、作成した分類器の精度を求める
- 標準偏差を計算し、それぞれの正解率のばらつきを見る

Mean(scores) = 0.49100

Stddev(scores) = 0.02888

どうしてこうなった

殆どの文書がリンクを持たないために、有効な特徴量ではなかった



特徴量をふやしてみる

- ソースコードの行数
- ソースコード以外の単語の数

Mean(scores) = 0.58300

Stddev(scores) = 0.02216

更に特徴量をふやしてみる

- １文の単語数の平均
- 文書中の各単語の文字数の平均
- 全ての文字が大文字である単語の個数
- 感嘆符の数

Mean(scores) = 0.57650

Stddev(scores) = 0.03557

なぜ精度が落ちた？

- 新しい文書のクラスは、５近傍法なので5つの近傍文書の中で最も多くを占めるクラス
- 文書間の距離はユークリッド距離において計算される

- 全ての特徴量が同じ比率で計算される

具体的に

表5-3　文書の特徴量

文書	NumLinks（リンクの数）	NumTextTokens（単語の数）
A	2	20
B	0	25
New	1	23

リンクの数を重視して類似度を出したい
新しい文書はAに似ていると判断して欲しい

ではどうするか

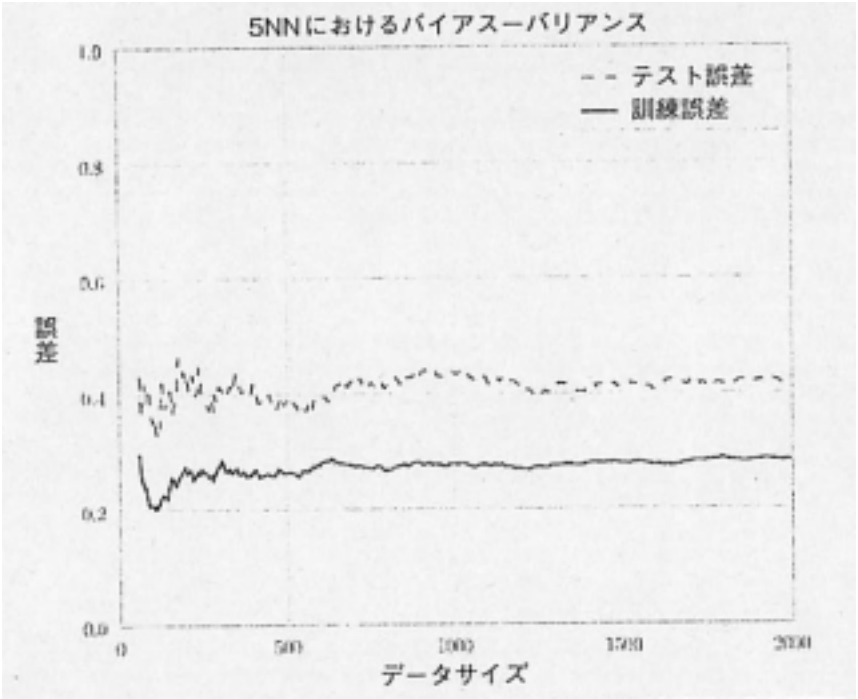
- データを追加する
  - 学習データの追加
- モデルの複雑さを調整する
  - K近傍のkを変更する
- 特徴量を修正する
  - 追加や削除、スケールの変更
- モデルを変更する
  - K近傍法に変わるモデルの使用

バイアス–バリアンス

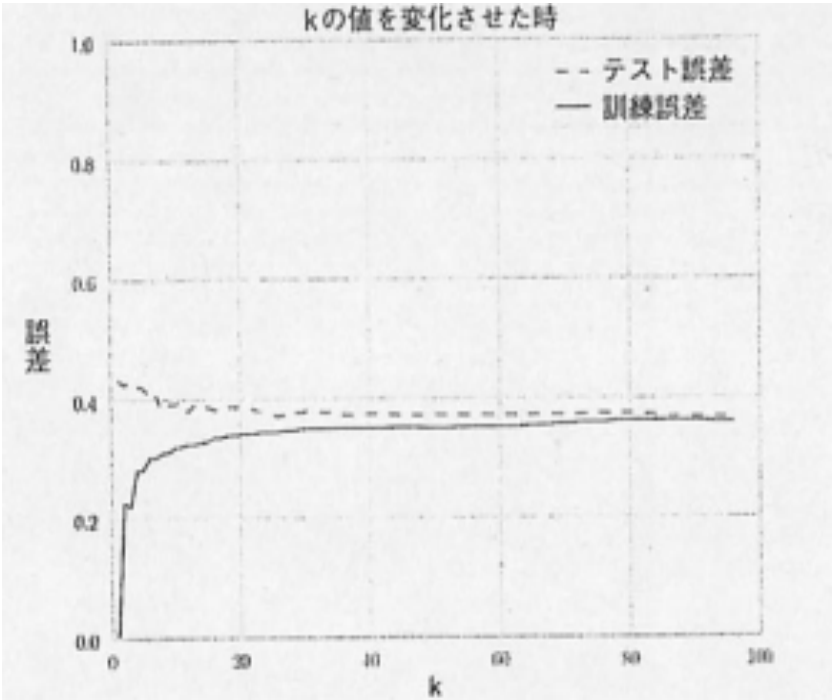
- バイアスが大きい（未学習）
  - 特徴量を増やす、モデルを複雑にする、モデルの変更
- バリアンスが大きい（過学習）
  - モデルを簡単にする、データ量を多くする

今回のシステムでは？

- データサイズやkを変化させた時の訓練誤差とテスト誤差を調べる



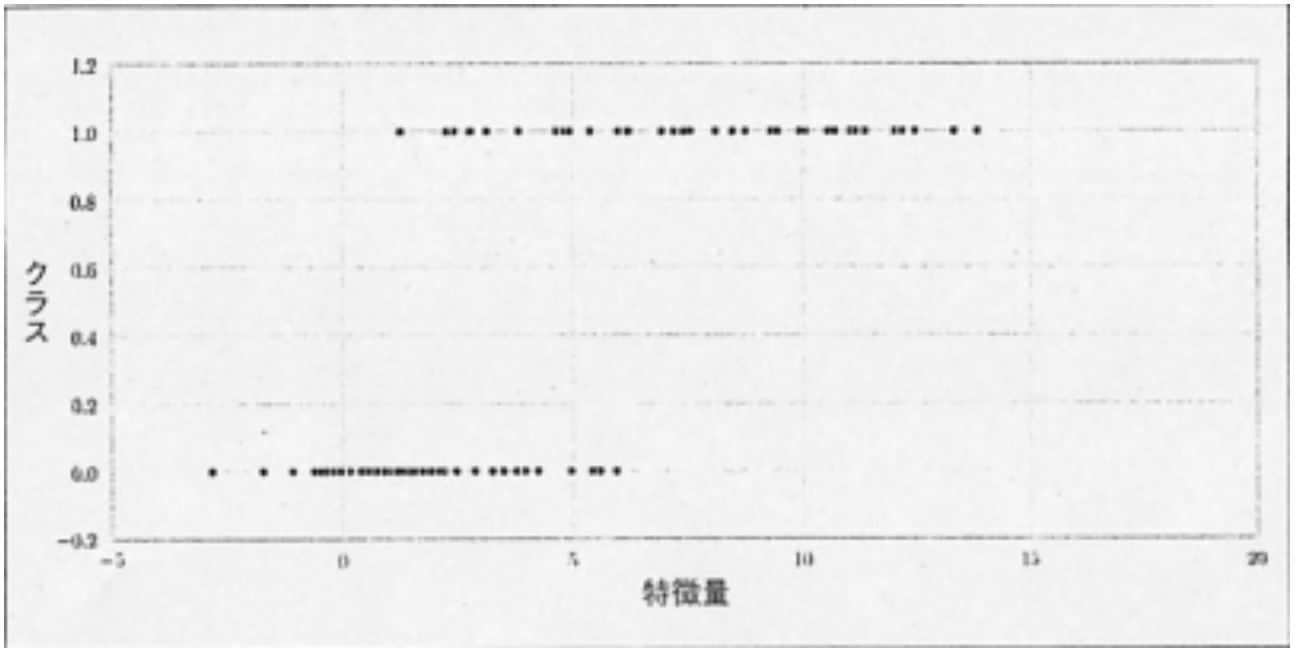
バリアンス大



k→大（時間がかかる）

ロジスティック回帰1

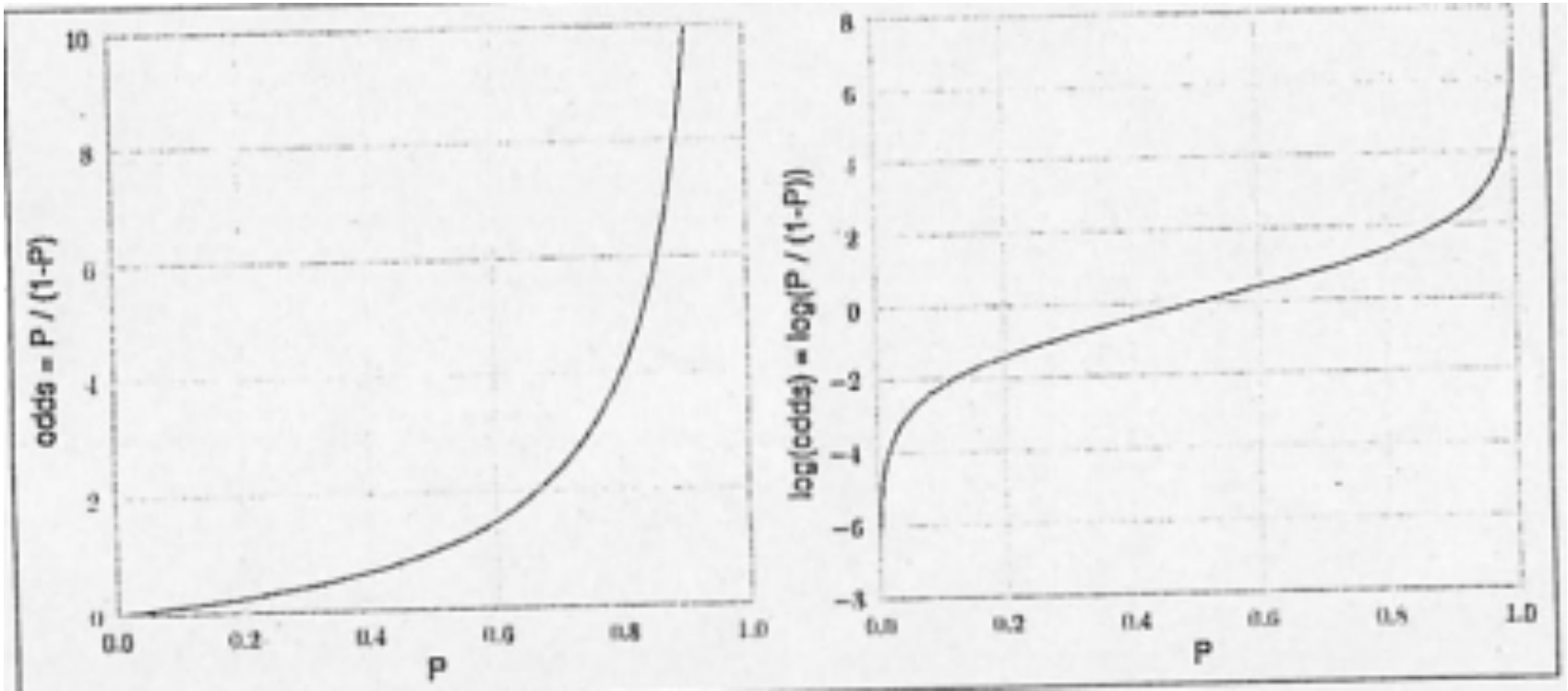
- 0と1のクラスを持つデータ(ノイズあり)



- どちらのクラスに属するかを確率で表したい

ロジスティック回帰2

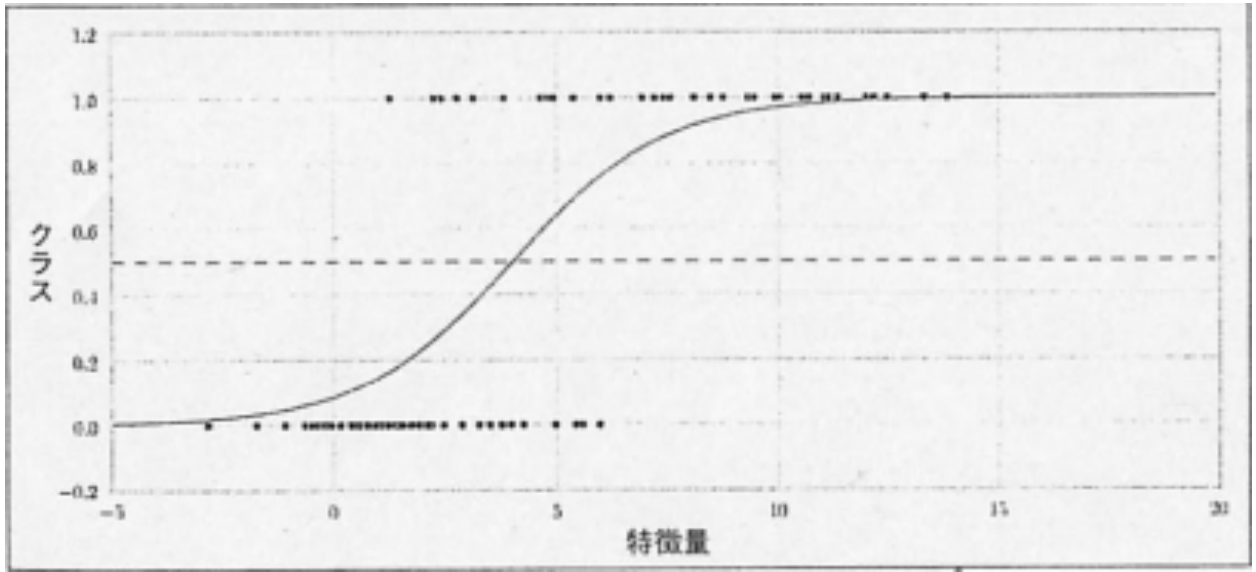
- オッズ比
  - 確率同士の比率（50%でA、50%でBなら1対1）
  - オッズの取りうる値（左図）とその対数（右図）



ロジスティック回帰3

- マイナス無限大からプラス無限大の範囲を0から1の範囲に収められる

- 特徴量の組み合わせをlog(odds)にフィッティングさせる



ロジスティック回帰4

- フィッティングのための式は1章の
  線形方程式より（特徴量が1次元の場合）

$$y_i = c_0 + c_1 x_i$$

$$\log\left(\frac{p_i}{1-p_i}\right) = c_0 + c_1 x_i$$

$$p_i = \frac{1}{1+e^{-(c_0+c_1 x_i)}}$$

- c0とc1をscikit-learnにより求める

実際に使ってみる

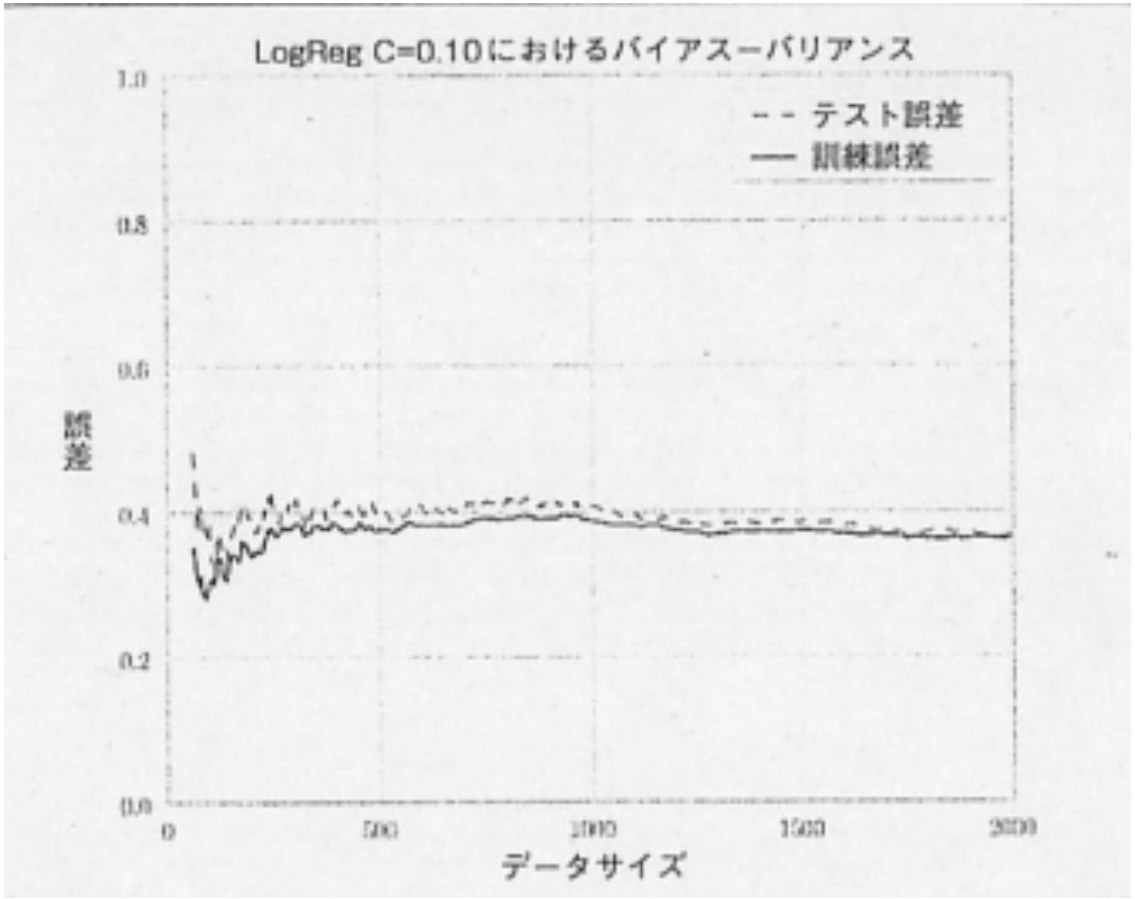
- ロジスティック回帰を用いた場合の正解率の平均と標準偏差
- Cはモデルの複雑さを調整する変数

表5-5　平均値と標準偏差の算出結果

手法	平均値	標準偏差
LogReg C=0.1	0.6310	0.02791
LogReg C=100.00	0.6300	0.03170
LogReg C=10.00	0.6300	0.03170
LogReg C=0.01	0.6295	0.02752
LogReg C=1.00	0.6290	0.03270
90NN	0.6280	0.02777

バイアス–バリアンス

- 誤差が大きい位置で安定し線同士が近いので未学習



目標の変更

- 回答の良し悪しではなく、良いデータか
  悪いデータかどちらかが検索できれば良い

- 適合率と再現率を指標として考えていく

適合率と再現率1

		分類器の予測	
		陽性	陰性
事実	陽性	True positive (TP)	False negative (FN)
	陰性	False positive (FP)	True negative (TN)

- 適合率は、陽性と予測した場合に実際に陽性である割合を表す

- 再現率は、陽性と予測した場合に実際の陽性のデータの何割を占めるかを表す

適合率と再現率2

- 先のようなパターンがあった時
  適合率は以下の式で表せる

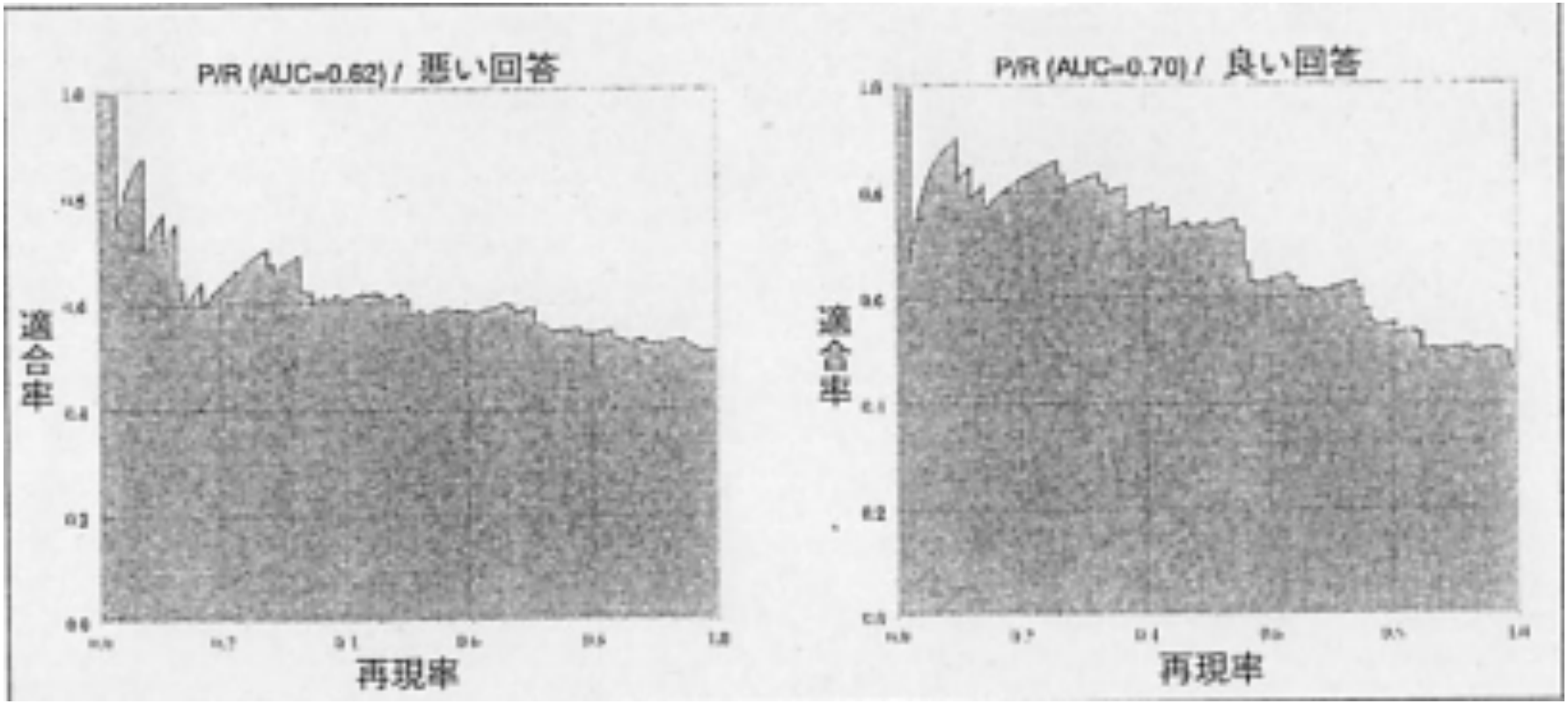
$$Precision = \frac{TP}{TP + FP}$$

- 先のようなパターンがあった時
  再現率は以下の式で表せる

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

良し悪しどちらを予測するのか

- 以下の図は適合率と再現率のグラフ



良い回答についての予測

- 良い回答と判定するための閾値を求める
  - 閾値を変えることによって精度が変わるため

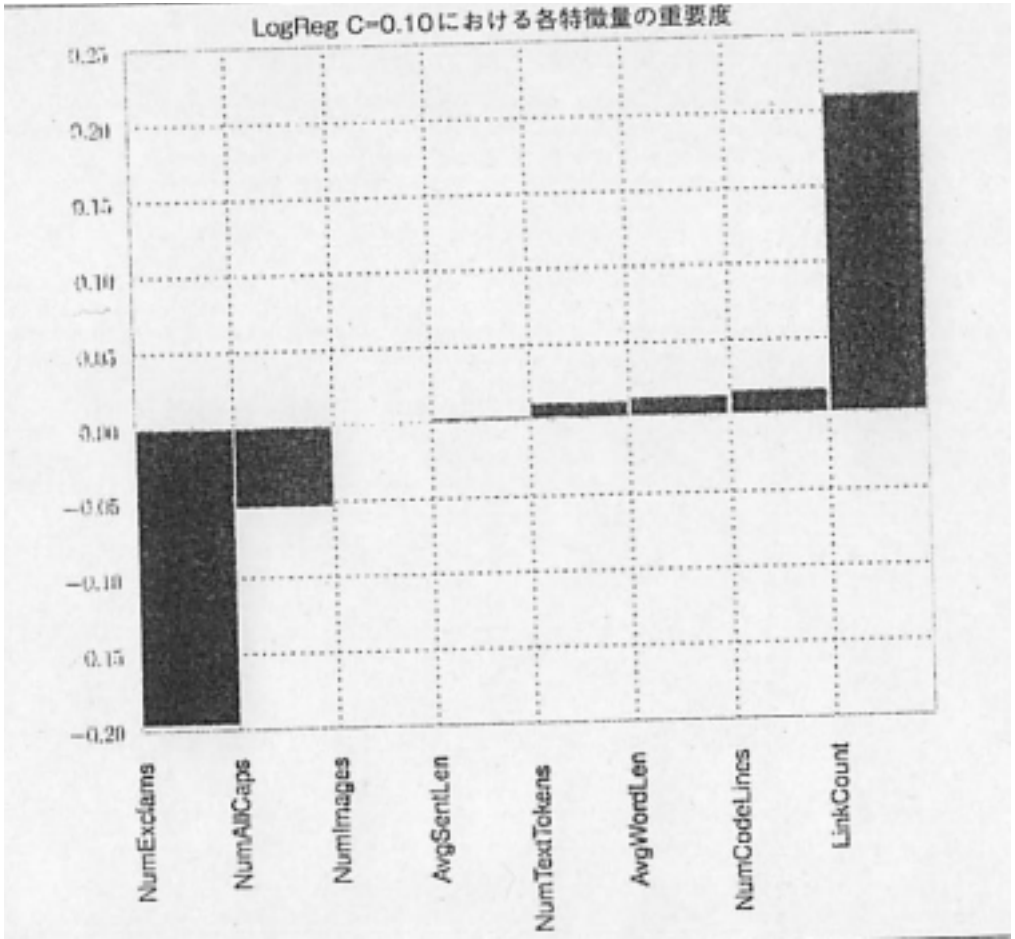
- 最適な閾値は以下のようになり、その場合の確率Pも求められる

  thresh = 0.63

  P = 0.81

分類器の最適化1

- 各特徴量が分類を行うためにどれほど影響があったかを表すのが以下の図



分類器の最適化2

- リンクの数が良い回答を判別する際に影響大
- 感嘆符の数が悪い回答を判別する際に影響大
- １文の平均長と画像の数はほぼ影響しない