Predictive Naive Bayes Classifier の提案と言語処理への適用

高村 大也 †　Dan Roth‡
† 東京工業大学 精密工学研究所 ‡ イリノイ大学アーバナ・シャンペーン校
takamura@pi.titech.ac.jp, danr@cs.uiuc.edu

1　序論

機械学習は，今日の自然言語処理において欠くことのできない存在となっている．特に，サポートベクターマシン（SVM）(Burges, 1998) は，文書分類，チャンキング，語義曖昧性解消，など自然言語処理のあらゆるタスクに適用されており，優れた結果を示している．一方，ナイーブベイズ分類器（NB）(Mitchell, 1997) は SVM より早い時期に自然言語処理に導入され，多くのタスクに適用されてきたが，現在では新手法の有効性を示すための比較相手として使われることが多く，分類性能という点においては SVM などの手法より劣るとされる．しかし我々は NB を拡張した手法を提案し，いくつかのタスクで実験を行うことにより，あらためて比較をする．

ここで，SVM に関する問題点をいくつか指摘する．まず，学習速度が非常に遅い．巨大なデータを扱う場合は無視できない問題である．また，超変数の調整が必要という問題もある．ここでは特に，訓練データへの偏りを制御するソフトマージン超変数 C について考える．この値の調整は軽視されがちであるが，実際の分類性能はこの C の値に大きく影響を受けることが多くある．また，交差検定などにより超変数を調整することも提案されているが，学習速度が遅いという第一の問題点により，非常に大量の計算リソースを必要とすることになる．

NB においては，事後確率最大化法（MAP) を用いて変数を推定するのが一般的である．また，事前確率は多項分布の共役分布であるディリクレ分布が用いられる．この場合，MAP は各素性の頻度に擬似頻度を加えることに等価となる．しかし，この疑似頻度をいくつにするか，すなわちどのディリクレ分布を用いるかについて決まった手法はなく，根拠のないまま 1.0 や 0.5 などが使用されている．このような状況は SVM の C を取り巻く状況と類似している．そこで，我々はこの疑似頻度を求める考案することにより，NB を拡張する．具体的には，事例 i の尤度を測る際に，i 自身の影響を取り除いて算出される予測的尤度（predictive または leave-one-out 尤度）を定義し，これを最大化するような疑似頻度を求める．また，leave-one-out 推定による予測正解率も考え，場合に応じてこちらも使用する．

提案手法の長所は，学習が速いことと，疑似頻度などの超変数を設定する必要がないことである．一方，提案手法の短所の一つは，頻度データ（事例が事象の生起回数を表わす整数の集合で表現されるようなデータ）にしか使えないことである．しかし，自然言語処理の多くのタスクにおいて，事例集合は頻度データの形で表現されていることを考えれば，これは決して致命的な欠点ではない．また，カーネル法が使えないという短所もある．こちらは，SVM などのカーネル分類器に対して PNB が劣る部分である．これについては，素性を工夫するなどして対応していく必要がある．

いくつかのタスクで実験を行い，提案手法と SVM を比較する．この比較を通して，状況に応じて適切な分類器を利用することの重要性についてあらためて考える．

2　関連研究

Hofmann ら (1998) は，EM アルゴリズムの E ステップにおいて各事例の隠れ変数の事後確率を計算する際に，その事例自身の影響を取り除いた変数値を使うことを提案している．Minka (2000) は予測的尤度を言語モデルの構築に利用することを提案している．我々の手法は，予測的尤度を分類器に利用している点，Minka が事象（言語モデルならば単語の生起）を一つずつ取り除いて尤度を測っているが我々は事例を一つずつ取り除いている点が異なる．しかし，数学的なモデルは Minka のものを参考に構築した．Zhai ら (2002) は，言語モデルの構築に，条件付予測的尤度が高くなるように超変数を求めることを提案している．これは本稿で提案する手法の考え方と類似しているが，Zhai らの手法は分類器に適用したものでなく，また調整できるのは疑似頻度の和に対応するスカラー値のみである．Suzuki ら (2006) は，EM アルゴリズムの超変数の選択に予測正解率を使用している．しかし，我々の手法が提案するような最適化の機構はない．

3　提案手法

提案手法の説明を行う．まず，基本的な多項分布ナイーブベイズモデルの説明をし，次に，予測的尤度の導入とそれを用いた超変数（疑似頻度）の推定方法について述べる．さらに，提案手法の変種を紹介する．

我々が数式中に使用する文字は以下のとおりである：

D	:	訓練データセット,
θ	:	変数集合,
$\mathbf{x}_i$	:	事例,
c_i	:	事例 $\mathbf{x}_i$ の所属するクラス,
w	:	素性,
V	:	素性集合,
n_{iw}	:	w の $\mathbf{x}_i$ での生起回数,
$P(\theta\|\alpha)$	:	ディリクレ事前分布,
α_w	:	w の超変数,
α	:	超変数集合.

3.1　多項分布ナイーブベイズモデル

本稿で用いる多項分布ナイーブベイズモデルについて説明する (McCallum and Nigam, 1998)．このモデルでは，データの尤度は，

$$P(D|\theta) = \prod_{i\in D} P(c_i|\theta) \prod_{w\in V} P(w|c_i)^{n_{iw}} \qquad (1)$$

と表わされる．

変数 $P(w|c)$ の事前分布として，多項分布の共役分布であるディリクレ分布 $P(\theta) = \prod_{w,c} P(w|c)^{\alpha_w - 1}$ を採

用する．$\alpha_w - 1$ が素性 w の疑似頻度に対応する．$P(c)$ についても事前分布を考えることができるが，$P(c)$ は分類結果にあまり影響しないことが知られているので，簡単のためここでは考えない．MAP 推定により変数を求めるには，

$$\begin{aligned}\log P(D|\theta)P(\theta|\alpha) &= \sum_{i\in D}\sum_{w\in V} n_{iw}\log P(w|c_i) + \\ &\quad \sum_i P(c_i) + \sum_{w,c}(\alpha_w - 1)\log P(w|c)\end{aligned}$$

を最大にするような変数を求めればよく，これは，

$$P(w|c,\alpha_w) = \frac{(\alpha_w - 1) + \sum_i \delta(c,c_i)n_{iw}}{\sum_w \left((\alpha_w - 1) + \sum_i \delta(c,c_i)n_{iw}\right)} \quad (2)$$

と表わされる．

3.2 Predictive Naive Bayes 分類器 (PNB)

さて，ここで予測能力が高くなるような超変数 α_w を求めたい．単純に尤度最大となるような α_w を求めようとすると，訓練データに過学習し，すべての α_w が 0 になってしまう．これは，事例 i の尤度を測る際に用いる変数値に i 自身の影響が含まれていることに起因する．そこで，i による影響だけを取り除いた変数値 $P_i(w|c,\alpha)$ を考えるとこれは

$$\frac{\left((\alpha_w - 1) + \sum_j \delta(c,c_j)n_{jw}\right) - \delta(c,c_i)n_{iw}}{\sum_w \left((\alpha_w - 1) + \sum_j \delta(c,c_j)n_{jw}\right) - \delta(c,c_i)n_i} \quad (3)$$

と表わされる．この変数値は i に依存するものの，$\sum_j \delta(c,c_j)n_{jw}$ などを保存しておけば定数時間で計算できることに注意してほしい．つまり，頻度データであることを利用することにより，事例を取り除いたときの変数値が高速に計算できるのである．

これにより，訓練データ中のある事例 i の尤度を測る際には，その事例自身の影響を除いた変数値を用いて測った訓練データ全体の尤度を考える．この新しい尤度を，予測的尤度 L_{pred} とよぶことにする．簡単のため $\alpha'_w = \alpha_w - 1$ を導入すると，L_{pred} は，

$$\begin{aligned}&\sum_{i\in D}\sum_{w\in V} n_{iw}\log\frac{\left(\alpha'_w + \sum_j \delta(c_j,c_i)n_{jw}\right) - n_{iw}}{\sum_w \left(\alpha'_w + \sum_j \delta(c_j,c_i)n_{jw}\right) - n_i} \\ = &\sum_{i\in D}\sum_{w\in V} n_{iw}\log\left(\alpha'_w + \sum_j \delta(c_j,c_i)n_{jw} - n_{iw}\right) \\ &- \sum_{i\in D}\sum_{w\in V} n_{iw}\log\left(\sum_w\left(\alpha'_w + \sum_j \delta(c_j,c_i)n_{jw}\right) - n_i\right)\end{aligned}$$

と表わされる．事前分布を導入していない $P(c)$ は省略している．この L_{pred} を最大化する α を求めること目的とする．これによりデータ生成の予測性能が高くなるような変数値が求められることが期待できる．

しかし，直接的に L_{pred} を最大化することは困難であるので，$\log(t+x) \geq q\log x + (1-q)\log t$ や，$\log(x) \leq ax - 1 + \log\frac{1}{a}$ を用いて，扱いやすい下限を求めると（ただし，$q = \hat{x}/(t+\hat{x})$，$1/a = \hat{x}$），

$$\begin{aligned}L_{pred} &\geq \sum_{i\in D}\sum_{w\in V} n_{iw}q_{iw}\log\alpha'_w \\ &\quad - \sum_{i\in D}\sum_{w\in V} n_{iw}a_i\sum_w \alpha'_w + const. \quad (4)\end{aligned}$$

なる不等式が得られる．ここで，

$$q_{iw} = \frac{\hat{\alpha}'_w}{\sum_j \delta(c_j,c_i)n_{jw} - n_{iw} + \hat{\alpha}'_w} \quad (5)$$

$$1/a_i = \sum_w\left(\alpha'_w + \sum_j \delta(c_j,c_i)n_{jw}\right) - n_i \quad (6)$$

である．

この下限を各 α_w について偏微分して 0 とおくことにより，α'_w の更新式が得られる：

$$\alpha'_w = \hat{\alpha}'_w \frac{\sum_i \frac{n_{iw}}{\sum_j \delta(c_j,c_i)n_{jw} - n_{iw} + \hat{\alpha}'_w}}{\sum_i \frac{n_i}{\sum_w\sum_j \delta(c_j,c_i)n_{jw} - n_i + \sum_w \hat{\alpha}'_w}}. \quad (7)$$

α が決定したら，式 (2) を用いて変数を求め，分類したい事例 j に対して，

$$argmax_c P(c)\prod_{w\in V} P(w|c)^{n_{jw}} \quad (8)$$

により分類をする．ここでは，$P(c)$ は最尤推定で求めることにする．

3.3 PNB の変種

超変数 α_w の取りうる値の範囲を限定することにより，PNB の様々な変種を考えることができる．制約を与えることにより，柔軟性は低くなるが，訓練データに過学習しにくくなる．これはトレードオフの関係である．

3.3.1 One-alpha PNB (OPNB)

PNB では，各素性が異なる α_w を持つことが許されたが，すべての素性について等しい値を持つという制約条件を加えることができる．これを One-alpha PNB（OPNB）とよぶことにする．PNB の場合と同様に，更新式を導出することができる：

$$\alpha' = \hat{\alpha}' \frac{\sum_i \sum_w \frac{n_{iw}}{\sum_j \delta(c_j,c_i)n_{jw} - n_{iw} + \hat{\alpha}'}}{|W|\sum_i \frac{n_i}{\sum_w\sum_j \delta(c_j,c_i)n_{jw} - n_i + \sum_w \hat{\alpha}'}}. \quad (9)$$

3.3.2 Constant-sum-of-alpha PNB (CPNB)

また，$\sum_w \alpha_w$ が一定値 A をとるという制約条件を付け加えることもできる．そのような分類器を Constant-sum-of-alpha PNB（CPNB）とよぶことにする．同様に更新式が導出できる：

$$\alpha'_w = \hat{\alpha}'_w \frac{A\sum_i \frac{n_{iw}}{\sum_j \delta(c_j,c_i)n_{jw} - n_{iw} + \hat{\alpha}'_w}}{\sum_w \sum_i \frac{\hat{\alpha}'_w n_{iw}}{\sum_j \delta(c_j,c_i)n_{jw} - n_{iw} + \hat{\alpha}'_w}}. \quad (10)$$

3.4 予測正解率

通常の NB（Simple NB, SNB）についても，スカラー値 α_{SNB} を決定する必要がある．そこで，i 自身の影響を除去した変数値を用いてクラス事後確率 $P_i(c|\mathbf{x}_i, \alpha_{SNB})$ を算出する：

$$P_i(c|\mathbf{x}_i, \alpha_{SNB}) \quad \propto \quad P(c) \prod_{w \in \mathbf{x}_i} P_i(w|c, \alpha_{SNB}). \quad (11)$$

そして $\max_c P_i(c|\mathbf{x}_i, \alpha_{SNB})$ で分類を行うことによりいくつかの異なる α_{SNB} について予測正解率 A_{pred} を算出する：

$$A_{pred} \quad = \quad \sum_{i \in D} \delta(c_i, argmax_c P_i(c|\mathbf{x}_i, \alpha_{SNB})). \quad (12)$$

このようにして，予測正解率が高い α_{SNB} を選択する．$P_i(w|c, \alpha)$ に対して用いたような高速計算方法が使えることに注意してほしい．

同様の手法を用いて，CPNB に対しても A の値を与える．

4 実験

4.1 実験設定

実験には，20NG，Spell，Movie 及び PPA の 4 種類のデータセットを用いた．20NG は，20 Newsgroups とよばれる文書分類の標準データセットであり，ニュースグループへの投稿メッセージから成る[1]．Spell は，context sensitive spelling correction(Roth, 1998) のデータセットである[2]．これは，綴りを間違えやすい英単語対（site と sight など）に対し，それらの単語が使用された文脈を与えて，正しい単語はどちらであるかを判定するタスクである．Movie(Pang and Lee, 2005) は，映画のレビューから抽出された文に対し，それが好意的意見か否かを判定するいわゆる感情極性分類である[3]．PPA は，前置詞句が動詞を修飾するか，名詞を修飾するかを判定する，Prepositional Phrase Attachment タスク (Roth, 1998) である（Spell と同じ場所から入手した）．各データセットのクラス数，サイズ，及び実験に用いた素性を表 1 にまとめる．訓練データとテストデータに分割されて配布されているものはその分割を用いて実験を行った．そうでないものは 10 分割交差検定を用いた．

評価した手法は，3 節で説明した SNB，OPNB，CPNB，PNB，及び比較用の SVM である．SNB の α_{SNB} と，CPNB の $A/|V|$ については，1.01，1.1，1.5，2.0，10.0，50.0，100.0 を試し，予測正解率の高いものを選択した．SVM のカーネル関数は線形カーネルを用いた．ソフトマージン超変数 C は，0.01，0.1，0.5，1.0，2.0，5.0，10.0 を試し，その中でテストデータに対する正解率が最も高いものを選んでいる．

評価には，いずれのデータセットに関しても正解率を用いた．

4.2 実験結果

実験結果を表 2 に示す．表中で超変数は，SNB ならば α_{SNB}，CPNB ならば $A/|V|$，SVM ならば C に対応する．交差検定を行ったものは，各分割で異なる超変数を選ぶことができるので，その平均値を記した．この実験結果からは，高い正解率を出している分類器はタスクによって異なることがわかる．古典的な手法である SNB が，超変数の調整機能を付け加えたことにより高い性能を示していることは注目に値する．SVM の超変数はテストデータに対する正解率が高くなるものを選んでおり，実際の正解率はこれより低くなりうることに注意してほしい．

また，PNB と SVM の 20NG に対する計算時間を表 3 に示す[4]．SNB は異なる α_{SNB} の値に対して計算をし，また予測正解率の計算も行っているので PNB と比較して遅くなっている．SVM については，正解率の最もよかった $C = 0.01$ の場合の計算時間を記載した．この比較からわかるように，SVM と比較して提案手法は（特に PNB）非常に学習が速いことがわかる．素性ベクトルの表現を，頻度表現でなく各素性が出現したか否かを示す二値表現にするなどの工夫により，SVM の学習を高速化することも可能であるが依然として差は大きいだろう．例えば能動学習など，学習時間が非常に重要となるタスクについては，分類器の選択に慎重になる必要がある．また，計算速度が遅い場合，交差検定などを用いて超変数を選択することも困難にな

表 2: 実験結果

タスク	手法	超変数	正解率 (%)
20NG	SNB	1.046	90.28
	OPNB	–	90.32
	CPNB	1.431	91.01
	PNB	–	**91.33**
	SVM	0.01	89.28
Spell	SNB	1.119	94.5
	OPNB	–	94.6
	CPNB	1.033	94.6
	PNB	–	94.8
	SVM	0.1	**95.1**
Sent	SNB	1.950	**77.78**
	OPNB	–	74.30
	CPNB	1.500	75.26
	PNB	–	73.59
	SVM	0.1	76.03
PPA	SNB	2.000	**84.47**
	OPNB	–	81.60
	CPNB	1.500	83.89
	PNB	–	81.63
	SVM	0.1	**84.47**

表 3: 20NG に対する計算時間

手法	学習時間
SNB	15 分 10 秒
PNB	10 秒
SVM	35 時間 21 分

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

[2] http://l2r.cs.uiuc.edu/~cogcomp/data.php

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data

[4] Intel(R) Pentium(R) 4 CPU 1.70GHz を搭載した計算機にて計算を行った．

表 1: 各データセットのクラス数，サイズ，及び用いた素性

データ	クラス数	訓練	テスト	素性
20NG	20	18828（10 分割交差検定）		文書内出現単語頻度
Spell	2	17082	4336	文内出現単語頻度
Movie	2	10662（10 分割交差検定）		snippet 内出現単語頻度
PPA	2	20801	3097	動詞，目的語，前置詞，前置詞句内出現名詞

るので，これも注意する必要があるだろう．

4.3 考察

分類正解率の点では，手法間の優劣はタスクに依存しており，どのような場合にどの手法が良いかについては現時点では不明である．しかし，SNB のような古典的な手法も良い結果を出すことは注目に値する．ここで，提案手法の特徴を SVM との比較を念頭に置いてまとめてみたい．

短所：

- 頻度データのみしか扱えない
- カーネル法が適用できない

長所：

- 学習が速い
- 超変数が自動的に決定できる

頻度データのみしか扱えないという短所だが，自然言語処理の少なからずのタスクにおいて，事例は頻度データとして表現される．よって，この短所は自然言語処理においては致命的なものではない．カーネル法が使えないことに対しては，素性を工夫するなどして対応していく必要があるだろう．長所に関して，超変数が自動的に決定できることは，注目に値する．実際のタスクでは，超変数の値によって実験結果が左右されることが多い．そのような場合にも，手法や素性の効果なのか超変数の効果なのか不明のまま実験結果が提示されるようなことが起こりにくくなるだろう．

5 結論

予測的尤度と予測正解率を用いることにより，NB を拡張した手法を提案した．提案手法は高速であり，また超変数の調整が不要であるという特長を持つ．また，頻度データにしか使用できない，カーネル法が使用できないなどの短所も持つ．タスクによっては SVM よりも優れた分類性能を示すことがわかった．しかし，どのような場合にどの手法がうまく働くのかについては，より詳細な調査が必要とされる．今回は訓練データの同時確率の予測的尤度を最大化することを試みたが，条件付確率の予測的尤度を直接最大化するような枠組みの開発も今後の方向性として考えている．

参考文献

Christopher J. C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Thomas Hofmann and Jan Puzicha. 1998. Statistical models for co-occurrence data. Technical Report AIM-1625, Artifical Intelligence Laboratory, Massachusetts Institute of Technology.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48.

Thomas Minka. 2000. Estimating a dirichlet distribution. Technical report, M.I.T.

Tom M. Mitchell. 1997. *Machine Learning*. McGraw Hill.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, pages 115–124.

Dan Roth. 1998. Learning to resolve natural language ambiguities: A unified approach. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-1998)*, pages 806–813.

Yasuhiro Suzuki, Hiroya Takamura, and Manabu Okumura. 2006. Application of semi-supervised learning to evaluative expression classification. In *Proceedings of the 7th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-06)*, pages 502–513.

ChengXiang Zhai and John Lafferty. 2002. Two-stage language models for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 49–56.