

# Variational AutoEncoder を利用した感情分析の領域適応

芝山 直希      新納 浩幸      古宮 嘉那子  
茨城大学 工学部 情報工学科

{15t4032g, hiroyuki.shinnou.0828, kanako.komiya.nlp}@vc.ibaraki.ac.jp

## 1 はじめに

本論文では Variational AutoEncoder (以下、VAE) を次元縮約器として利用し、文書データをソース領域とターゲット領域の共通の空間に埋め込む。その埋め込みベクトルを追加の素性として利用することで、感情分析の領域適応を試みる。

感情分析とはレビュー文書 (例えば映画のレビュー) が肯定的なものか、否定的なものかを判定するタスクである。これは文書分類の一種であり、教師あり学習を用いて解決できる。しかし判定先の文書が学習データの領域とは異なる領域の文書 (例えば書籍のレビュー) であった場合に、教師あり学習で得られた分類器の精度が下がってしまう。これが領域適応の問題である。

領域適応の問題に対する有効な手法の一つとして、データをソース領域とターゲット領域の共通の空間に埋め込むことが行われている。共通の空間で学習と識別を行えば、領域適応の問題は起こらない。問題はどのようにして共通の空間に埋め込むかである。簡易な方法として、ソース領域の空間とターゲット領域の空間を合わせた空間を考え、主成分分析 (特異値分解) を用いて次元を縮約することが考えられる。一方、近年、深層生成モデルとして VAE が提案された [3][2]。VAE の Encoder はデータに対してその潜在変数を求めるが、その潜在変数はデータを次元縮約したベクトルと見なすこともできる。このため VAE から構築できる潜在変数を、ソース領域とターゲット領域の共通の空間に埋め込まれたベクトルとして扱える。本論文では上記の考えに基づき、VAE を次元縮約器として利用し、感情分析の領域適応に応用する。具体的にはソース領域とターゲット領域の文書ベクトル  $x$  を VAE の Encoder により  $z$  に次元縮約する。学習と識別には  $x$  の代わりに  $x$  と  $z$  を連結した  $[x; z]$  を用いる。実験では次元縮約の手法として特異値分解 (SVD) を用いた物と比較し、感情分析の領域適応に対して、VAE の利用可能性について検討する。

## 2 関連研究

ターゲット領域のラベル付きデータを用いない教師なしの領域適応の手法としては、一般に素性ベースの手法が用いられる [4]。素性ベースの手法の基本的な考え方はソース領域とターゲット領域の共通空間にそれぞれのデータをマッピングし、共通空間上で学習と識別を行うというものである。共通空間の求め方としては、単純には、ソース領域とターゲット領域を合わせた空間を特異値分解などで次元縮約を行えばよい。古典的手法である SCL [1] や近年提案されている CORAL [6] など、この考え方に基づいている。本論文ではこの次元縮約する部分に VAE を利用したものと捉えられる。次元縮約したベクトルをそのまま利用することもできるが、SCL のように基のベクトルに連結して利用する方が、精度が高くなるため、本論文でもそのような使い方をを行う。

## 3 VAE による次元縮約

次元縮約の標準的手法は特異値分解である。これは行列分解を利用した低ランク近似であり、次元縮約は線形変換により行われるため、その能力には自ずと限界がある。そのため非線形な変換法がいくつか提案されている。本論文で行う VAE による次元縮約もその一つと捉えられる。

VAE は生成モデルであり、データ  $x$  を生成する確率モデル  $p(x)$  を構築するのが目的である。基本的には  $p(x)$  の対数尤度を最大化するモデルを求めればよい。Jensen の不等式を用いると以下が示せる。

$$\log p(x) \geq -D_{KL}(q(z)||p(z)) + \int q(z) \log p(x|z) dz \quad (1)$$

VAE は  $p(x)$  の対数尤度を最大化する代わりに、その下限である式 1 の右辺を最大化する。つまり式 1 の右辺の符号を反転させた式 2 を最小化することで  $p(x)$

のパラメータを求める。

$$D_{KL}(q(z)||p(z)) - \int q(z) \log p(x|z) dz \quad (2)$$

式2で $z$ は $x$ の潜在変数であり、 $q$ は $z$ の分布である。 $q$ に制限はなく、正規分布に設定すれば、式2の第1項は解析的にパラメータの式に変形される。また式2の第2項は Reparametrization trick を用いることで、パラメータのサンプリングの形で表現できる。

図1に VAE のネットワーク図を示す。VAE ではデータ  $x$  からその潜在変数  $z$  への Encoder と  $z$  から  $x$  を生成する Decoder を学習している。本論文では VAE を次元縮約として利用する。 $z$  を  $x$  の次元縮約したベクトルとして考える。

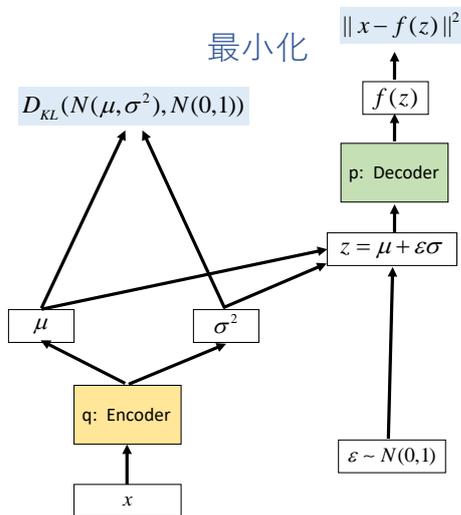


図1: VAE のネットワーク図

## 4 提案手法

ここでは以下の3ステップにより訓練およびテストのデータを変換する(図2参照)。

1. ターゲット領域とソース領域の全データを用いて VAE の学習を行う。
2. 各領域のデータ  $x$  に対して VAE の Encoder を用いて、その次元縮約ベクトル  $z$  を得る。
3.  $x$  に  $z$  を連結したデータ  $[x; z]$  を作成する。

以降の分類器の学習、テストデータの識別は、上記の  $[x; z]$  を用いて行う。

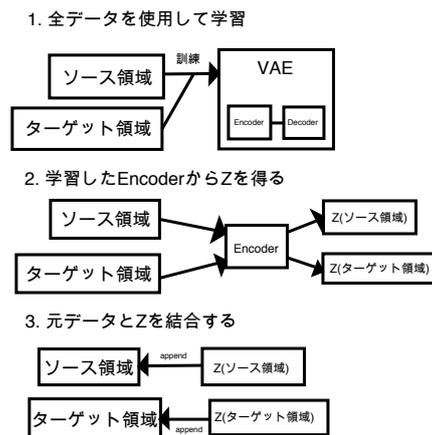


図2: 提案手法概要

## 5 実験

実験には Webis-CLS-10[5] の日本語データ<sup>1</sup>を使用した。このデータセットは books, DVD, music の3つの領域を持ち、それぞれ Bag of Words 形式で記録された文章データの集合である。各データは positive, negative の二値に分類・ラベル付与された Amazon カスタマーレビューであり、訓練用、テスト用、その他 (Unlabeled) の3つのファイルに分けられている。

### 5.1 前処理

まず全領域の全データを用いて Bag of Words 形式から配列上の index に対応させる変換表を作成した。良い結果を得るために、作成工程において Unlabeled ファイルを最後に読み込むようにした。Unlabeled ファイルの情報を含まない変換表も別途作成した。

その後変換表を元に Python のライブラリの一つである SciPy の疎行列型の配列に変換して保存した。この際訓練用、テスト用のデータは2つの変換表から次元数の小さい配列と大きい配列にそれぞれ変換し保存した。以後 Unlabeled データに対応した変換表を用いて作成したデータを Large データ、対応していない変換表を用いて作成したデータを Small データとする。

<sup>1</sup>[5]の本来のタスクはクロスリンガル感情分析であるが、各言語のデータは「対象言語での感情分析をタスクとするデータセット」として流用できる。

## 5.2 提案手法の適用

実験を簡略化する為、全ての Large データを用いて VAE を学習した。VAE は Chainer のサンプルプログラムのモデルを利用した。学習時のハイパーパラメータを表 1 に示す。

表 1: Variational AutoEncoder 学習時のモデルのパラメータ

epoch 数	100
隠れ層の次元数	100
縮約後ベクトルの次元数	100
ミニバッチサイズ	400

その後、学習したモデルの Encoder を用いて訓練・テスト用の Large データの次元縮約を行なった。そして、各 Small データに対応する縮約された Large データを付加した。付加が完了したデータを実験データとする。また、SVD を用いて同様の次元縮約及び負荷を行った。SVD による縮約データが付加されたデータを SVD データとする。

## 5.3 検証用分類器の訓練

訓練用実験データを用いて各領域の分類器を訓練した。分類器には scikit-learn にて実装されている SVM を用いた。訓練時のハイパーパラメータを表 2 に示す。

表 2: SVM 学習時のモデルのパラメータ

使用モデル	SVC
事前処理	MinMaxScaler
C	1,10,100,1000 GridSearchCV を用いて最適なものを選択
備考	表記のないパラメータはモデルの初期値を使用

その後、各分類器に各領域のテスト用実験データを分類させ、精度を記録した。また、SVD データに対して同様の操作・記録を行った。

## 5.4 実験結果

実験の結果を表 3 に示す。表 3 における None は領域適応の手法を用いず、ソース領域の訓練データのみから分類器を学習し、その分類器を用いてターゲット

領域のテストデータを識別した結果である。SVD は特異値分解を利用して次元縮尺する領域適応の手法、VAE は VAE を利用して次元縮尺する領域適応の手法（提案手法）を表す。

表 3: 実験結果（正解率 %）

領域適応	None	SVD	VAE
B → D	<b>77.75</b>	77.65	76.65
B → M	72.50	<b>75.20</b>	74.80
D → B	<b>77.65</b>	76.90	77.00
D → M	76.40	77.65	<b>77.75</b>
M → B	76.65	<b>76.80</b>	75.90
M → D	<b>80.00</b>	79.45	78.70
平均	76.83	<b>77.28</b>	76.77

提案手法となる VAE を用いた領域適応の手法は、領域適応の手法を全く利用しないものよりも悪い結果となり、本実験に限れば提案手法は有効でなかったと言える。

## 6 考察

提案手法を用いる場合よりも SVD を用いる方が領域適応時の平均精度が 0.51% 高いという結果が得られた。この結果から提案手法及び実験の問題点を考察していく。

まず考えられる実験時の問題点が、VAE の訓練エポック数が適切でなかった可能性である。当実験と同条件下で訓練した VAE の損失は次のように遷移した。

図 3 から学習進行度に比例して損失のばらつきが大きくなっている印象を受けた。また、全体を通して損失が激増する頻度が高い。これは特定領域への適応度が高い状態のモデルに適応度の低い領域のデータが入力される現象が多発していると見ることが出来る。この点から、訓練スクリプトに問題があり、訓練時の入力順に領域依存の偏りが発生していると推測した。

訓練スクリプトは大規模なデータに対応可能にするために SciPy 疎行列型で読み込んだ全データを結合することなく、プログラム引数の順にミニバッチ学習させる方式を用いている。引数は訓練・テストデータ、Unlabeld データ（領域順は book, DVD, music）の順となっていた。読み込む全てのデータを単一の配列に

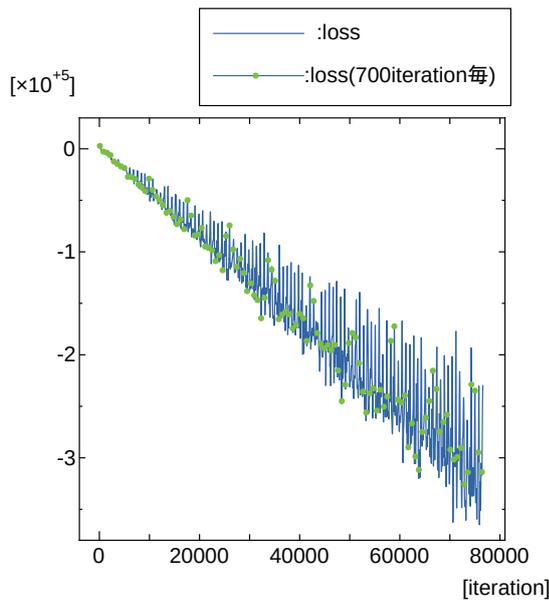


図 3: VAE の損失の推移

統合し、ランダムな順番でミニバッチ学習させればより領域依存度の低い出力が得られるだろう。

## 7 おわりに

本論文では感情分析の領域適応に対して、Variational AutoEncoder の Encoder を利用した領域適応の手法を提案した。ソース領域とターゲット領域のデータを用いて訓練した Variational AutoEncoder の Encoder で両データを次元縮約し、得られたデータを対応するデータに付加することでターゲット領域に適応する。実験では、提案手法と SVD の 2 つの手法を SVM で識別精度を用いて比較した。その結果は、SVD を使用した方が有効度がわずかに高いというものだった。

実験結果及び実験に使用したプログラムを再度確認・推敲したところ、提案手法のモデル訓練の過程で、領域に依存しない出力が得られるような訓練ができていなかった可能性が存在することが分かった。まずは提案手法の訓練過程の問題点を解消して出力の領域依存度を低下させ、VAE の利用可能性を再度検証していきたい。

## 参考文献

- [1] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *EMNLP-2006*, pages 120–128, 2006.
- [2] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [4] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.
- [5] Peter Prettenhofer and Benno Stein. Cross-Language Text Classification using Structural Correspondence Learning. In *48th Annual Meeting of the Association of Computational Linguistics (ACL 10)*, pages 1118–1127. Association for Computational Linguistics, July 2010.
- [6] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of Frustratingly Easy Domain Adaptation. *AAAI*, 2016.