11章

11.0 冒頭文

1. 本書で紹介している手法は、多くの訓練データが必要という制限がある
    a. これを用意するのは難しい
    b. 少数データで分析する方法を模索する
2. 以下の手法で、そのような制限下の分析を可能にする
    a. データ拡張
    b. 転移学習
    c. 半教師あり学習
    d. 自己教師学習
    e. マルチタスク学習
    f. ドメイン適応
    g. 少数事例学習
    h. 能動学習

11.1 はじめに

前提条件1 データが少ない

1. 手元のデータからタスク$\mathcal{T}$のために学習したい
    a. 主にクラス分類を想定する
    b. しかし、大部分はクラス分類以外でも使える
2. 手元のデータは$\mathcal{D} = \{(\mathbf{x}_n, d_n)\}_{n=1,\ldots,N}$
    a. Nは小さい
    b. だが、これを用いて学習用の準備を行う

そのための主な手法

1. データ拡張(11.2節)
    a. 後述の手法(前述のc.からh.の手法)のもとになる

    b. 単体でもよくつかわれる
2. 転移学習(11.3節)
    a. 広い意味ではタスク$\mathcal{T}'$の学習経験を利用することを指す
        i. 日経平均株価の予測で使ったモデルをダウ平均株価の予測に活かす、のような話？
    b. 狭い意味だと特定の手法を指す
        i. ここでは具体的な言及なし

前提条件2 ラベルなしデータ

1. 手元のデータは$\mathcal{D}_{\mathrm{UL}} = \{(\mathbf{x}_n)\}_{n=1,\ldots,N}$
    a. 正解ラベルがない
    b. 半教師あり学習や自己教師あり学習は、このようなデータを対象としている

それ以外の手法

その他の手法の概要は、以下の通り。

マルチタスク学習	異なる複数タスクを1つのネットワークで学習する
ドメイン適応	訓練データとテストデータの間で統計的な分布にずれがある場合に適用する
少数事例学習	訓練データの数が極端に小さい場合を対象にした学習
能動学習	ラベルなしデータから限られたサンプルを選んで正解ラベルをつける(よいサンプルを選ぶ方法)

11. 2 データ拡張

11.2.1 基本的な考え方

前提条件 データ拡張

1. $\mathcal{D} = \{(\mathbf{x}_n, d_n)\}_{n=1,\ldots,N}$をもっている
    a. $\mathbf{x}_n$を適当な変換φで$\mathbf{x}'_n \equiv \varphi(\mathbf{x}_n)$とする
    b. 新たなサンプル$(\mathbf{x}'_n, d_n)$として利用する

2. 変換は自由度が高いが、以下の制約などがある
    a. 変換によってラベルが変わらない
    b. テストデータに出てきそうなデータにしないといけない
3. 数だけは際限なく増やすことができる

11.2.2 主に画像に対して有効な方法

- そもそもデータ拡張は画像データに対して有用
- 画像データを変化させても正解ラベルが変化しないような変換をする必要がある
- 図11.1にあるような変換を加える
- 図11.2にあるよう、正解ラベルが変わってしまいそうな加工等はよくない
    - また、タスクによって有効な加工が違う点にも注意
    - カットアウト、ランダム消去は物体認識に効果が高い
- 音声でも同様の有効な加工が存在する

11.2.3 主に言語に対して有効な方法

- 言語の加工は、意味が同じ言い換えを作る方法が考えられる
- 具体的には以下の方法がある
    - 文中の単語を類義語に入れ替える
    - 機械翻訳で別言語にしたものを再翻訳する
- 文の埋め込みベクトルにノイズを足す方法も考えられる
- 多様な言い換えができるかどうかが重要

11.2.4 その他の方法

- クラス分類を対象とするmixupという方法がある
- 2つのサンプルを足し合わせる方法
- 特徴量、正解ラベルをそれぞれ適当な加重和で表現する
    - $(\mathbf{x}_i, d_i)$と$(\mathbf{x}_j, d_j)$について適当な$\boldsymbol{\lambda}$で以下のようにサンプルを作る
        - $(\lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \lambda\, d_i + \ (1-\lambda) d_j)$

- λは適当なベータ分布からサンプルしたもの

11.2.5 データ拡張の自動探索

- 変換の候補が多数あるとき、その中から選択するのは重要
- バリデーションデータへの予測精度が最大化されるように変換パラメータを選択するような手法がある
  - パラメータ空間の探索方法もよく提案されている

11.3 転移学習

11.3.1 概要

1. $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1,\ldots,N}$と$\mathcal{D}'$をもっている
   a. 前者については量が十分でなく、後者のデータは十分にある
      i. 新規に上場した銘柄の株価を既存の銘柄の株価から推定するような話？
   b. $\mathcal{D}'$で学習したタスク$\mathcal{T}'$と$\mathcal{D}$で学習したタスク$\mathcal{T}$には類似性が必要
   c. 出力の空間は違っていてかまわないが、特徴量・入力は同じ必要がある
2. 以下のような方法で転移学習を行う
   a. あるネットワークで$\mathcal{D}'$を使いタスク$\mathcal{T}'$の学習をし、その結果$\mathbf{f}'$を利用する
      i. このタスク$\mathcal{T}'$の学習を$\mathcal{T}$の事前学習と呼ぶ
      ii. 逆に、タスク$\mathcal{T}$は$\mathcal{T}\mathcal{T}'$を下流タスクと呼ぶ
      iii. タスク$\mathcal{T}'$を十分学習できたものとする
   b. 結果$\mathbf{f}'$を下流タスクのための特徴抽出器として使うことができる
   c. 結果$\mathbf{f}'$の構造、一部重みをコピーして別のネットワークを作る方法もある
      i. この方法はファインチューニングと呼ばれる
3. 難しい点もある
   a. タスクの類似度を測るのは難しい
   b. 事前に転移学習の成否を予測するのも難しい

11.3.2 特徴抽出器としての利用

1. $\mathbf{f}'$の中間層を1つ選び、入力に対する中間出力とする
    a. 入力$\mathbf{x}_i$に対して選んだ層の出力$\mathbf{z}\left(\mathbf{x}_i\right)$を中間出力とする
    b. $\mathcal{D}=\left\{\left(\mathbf{x}_n, \mathbf{y}_n\right)\right\}_{n=1,\ldots,N}$から$\tilde{\mathcal{D}}=\left\{\left(\mathbf{z}\left(\mathbf{x}_n\right), \mathbf{y}_n\right)\right\}_{n=1,\ldots,N}$をつくる
    c. 新たに作ったデータがよいものであれば、学習アルゴリズムはシンプルでも性能がでると想定できる
    d. 層の選び方、
    e.
    f.
    g.
    h.
    i.
    j.
    k.
    l.
    m.
    n.
    o.
    p.
    q.
    r.
    s.
    t.
    u.
    v.
    w.
    x.
    y.
    z.

aa.

ab.

ac.

ad.

ae.

af.

ag.

ah.

ai.

aj.

ak.

al.

am.

an.

ao.

ap.

aq.

ar.

as.

at.

au.

av.

aw.

ax.

ay.

az.

ba.

bb.

bc.

bd.

be.

bf.

bg.

bh.

bi.

bj.

bk.

bl.

bm.

bn.

bo.

bp.

bq.

br.

bs.

bt.

bu.

bv.

bw.

bx.

by.

bz.

ca.

cb.

cc.

cd.

ce.

cf.

cg.

ch.

ci.

cj.

ck.

cl.

cm.

cn.

co.

cp.

cq. 複数の

cr. $\left(\mathbf{x}_i, d_i\right)$を入力として選んだ中間層の出力を利用して新たなデータ$\left(\mathbf{z}\left(\mathbf{x}_i\right), d_i\right)$を作って学習する

cs. シンプルな予測器でも、選択した中間層がよければ性能がでると想定できる

ct. 選択する中間層、および複数の層の出力を組み合わせるバリエーションに自由度がある

    i. 選び方は、経験的な検証に依存する

11.3. 3 ファインチューニング

1. $\mathbf{f}'$の一部をコピーして、新たなネットワーク$\mathbf{f}$をつくる

    a. $\mathcal{T}'$と$\mathcal{T}$が近ければネットワークの変化はごくわずかだと想定され、それが名前の由来となっている

    b. 単純に出力層以外をコピーする方法などが典型

        i. 出力層のみ、クラス数の増減などの影響から作り直す必要がある

        ii. コピーした層に追加で中間層を追加、あるいは部分的にコピーから削減するなどをしてもよい

    c. コピー、加工して作ったネットワーク$\mathbf{f}$を$\mathcal{D}$を用いて学習する

        i. 追加した層の重みのみランダムで、そのほかはコピー元の値で初期化する
        ii. この状態の$\mathbf{f}$を事前学習済みであると呼ぶ
    d. 学習するパラメータ数は調整できる
        i. 訓練は一部の層の重みだけ更新することもできる
        ii. 少数データによる学習となるので、すべて更新する方法が裁量ではないと思われる
            1. 過剰適合のおそれがあるため
        iii. 典型的には、出力層に近い一部の層のみを訓練する傾向にある
            1. 入力に近い層よりも、出力に近い層のほうが具体的なタスク依存度が高いとされているため

11.3. 4 ImageNet事前学習

1. 物体認識をするための大きなデータセットを用いた事前学習
    a. 学習結果のネットワークを前節、前々節の方法で学習できる
    b. 手持ちの訓練データが小さくても、学習に利用できる
        i. 物体認識から、タスクとしては遠い単眼深度推定までそれが可能

11.3. 5 事前学習の有効性

1. 事前学習に3つの有効性が示唆されている
    a. 予測精度の向上
    b. 学習の収束速度の向上
    c. 推論時の頑健性の向上
2. 予測精度の向上について
    a. 事前学習を通じて獲得した特徴抽出の能力が、下流タスクでも再利用されるという理解
    b. 下流タスクのデータ量が十分な場合にはそれほどの効果がないことも報告されている(179より)
3. 学習の収束速度の向上
    a. 少ない重みの更新で精度が向上すること

        i. 下流タスクのデータセットが大きくても期待できる能力とされる(180、236より)
        ii. 事前学習がランダムな初期値よりも統計的によい初期値に重みを重みに与えているためという理解
4. 推論時の頑健性の向上
    a. 以下について性能が向上することを指す(181、182より)
        i. GAN
        ii. 分布外入力の検出
        iii. 確信度の校正精度
    b. 事前学習なしだと最低限の特徴空間しか得られないからとされている
5. 残された課題
    a. 以下の解明されていない部分がある(180、326より)
        i. 統計的な性質が異なる、特徴の再利用がほぼないタスク間でも性能が向上する
            1. 事前学習用のデータセットが大きいから平気…というような理解

11.4 半教師あり学習

11.4.1 概要

1. $\mathcal{D}_{\mathbf{L}} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1,\ldots,N}$と$\mathcal{D}_{\mathbf{UL}} = \{(\mathbf{x}'_n)\}_{n=1,\ldots,N'}$をもっている
    a. 前者についてはラベルがあるが少数で、後者についてはラベルがないが多い、といった状態
        i. アノテーションコストが高いが、データ自体は結構ある…といった状態
    b. $\mathcal{D}_{\mathbf{L}}$は普通の教師あり学習で、それ以外はいろいろ損失関数などを準備して訓練に利用する流れ
        i. 各手法の特徴はラベルなしデータの利用方法にある
    c. 最小化したいのは2つの損失関数$E'(\mathbf{w}; \mathcal{D}_{UL})$と$E(\mathbf{w}; \mathcal{D}_L)$の和
        i. 後者は普通の教師あり学習として考える
        ii. 前者の特徴づけの異なる3つの手法を紹介する

1. 一致性正則化
2. 疑似ラベル
    a. 教師データの話に言及していて損失関数そのものの取り扱いについてポイントがあるタイプの手法ではない気がする
    b. kaggleでよくみかけた手法、として認識している
3. エントロピー最小化

11.4. 2 一致性正則化

1. $E'(\mathbf{w};\mathcal{D}_{UL})$を以下のように特徴づける場合

$$\lambda \sum_{x_i \in D_{UL}} \delta(y(x_i;w), y(x_i';w))$$

2. 式の特徴は、以下の通り
    a. 各記号の特徴
        i. λは教師あり学習部分に対してどれだけ別の損失関数を重要視するか？を表現している
        ii. $\delta(y(x_i;w), y(x_i';w))$部分は、2つの値の差を表現するものとなっている
            1. yは入力に対する予測結果
            2. xとx'はデータ拡張などによって前者から後者を作成する
            3. 予測結果が変動によってあまり変わらないように学習を行う
    b. データの作成方法について
        i. 敵対的事例からくる仮想敵対的学習
            1. 予測結果を大きく変える「小さな変動」を逆算して作る
        ii. ネットワークを確率的に変動させたとき、同じ入力に対する予測が不変になることを要請する(δで定義していた差の評価をより厳しくする…？)
        iii. 学習中のネットワークの重みの時間変化を変動の源とするような方法(temporal ensemble)
        iv. ネットワークのパラメータの時間方向の移動平均を求めて、それを正解ラベルの作成に利用する方法(mean teacher)

11.4. 3 疑似ラベル

1. 2つのネットワークS, Tを利用した方法
    a. 教師ネットワークT、生徒ネットワークSとする
    b. 2つのネットワークが同一の場合というのもあり、その場合をself-trainingと呼ぶ
2. 知識蒸留とつながりがある方法
    a. ラベルなしのデータを使ってラベル付きデータの学習結果を転移する知識蒸留といえる
3. $\mathcal{D}_{UL} = \{\mathbf{x}_n\}_{n=1}^N$を以下のように用いる
    a. ネットワークTに$\mathbf{x}_n$入力してラベルの予測y_nを得る
    b. ラベル付きということにしたデータ$\mathcal{D}'_{UL} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$を得る
4. $E'(\mathbf{w}; \mathcal{D}_{UL})$を以下のように特徴づける

$$\lambda \sum_{x_i \in D_{UL}} \delta(y(x_i; w), y_i)$$

5. 式の特徴は、以下の通り
    a. 各記号の特徴
        i. λは教師あり学習部分に対してどれだけ別の損失関数を重要視するか？を表現していて何か特殊な理由等ない限りは1となる想定
        ii. $\delta(y(x_i; w), y_i)$部分は、2つの値の差を表現するものとなっている
            1. yはネットワークSへの入力に対する予測結果
            2. y_iはネットワークTによる疑似ラベル
            3. 予測の差をとっている
        iii. 教師あり学習部分をSの学習に使わない場合というのもある(損失は上の式だけ)
6. 疑似ラベルの作り方について
    a. 決定的な方法はない
    b. 予測結果のクラス分布を疑似ラベルとする
    c. 予測結果のクラスラベルを疑似ラベルとする

d. 予測クラス分布に従ってクラスをサンプリングする
e. 疑似ラベル、というより自己訓練法の有効性が確認されている
f. しかし、どの方法が良いのかははっきりしていない
g. 転移学習を上回る場合があり、それはタスクのちがいというものが疑似ラベル法には存在しないからと理解されている

11.4.4 エントロピー最小化

1. $E'(\mathbf{w};\mathcal{D}_{UL})$を以下のように特徴づける場合

$$-\sum_{k=1}^{K} y_k \log\left(y_k\right)$$

2. ネットワークが出力するクラススコアのエントロピーが小さくするような方法
    a. ラベル付きでないデータもなんらかのクラスには属するだろう、という期待による

11.5 自己教師あり学習

11.5.1 概要

1. 事前学習用のラベル付きデータがない場合の方法
    a. これは、転移学習の前提を満たさないことを意味する
    b. しかし、正解ラベルのない大量のデータ、少数のラベル付きデータならあるという想定
    c. クラスタリングのように、なんらかのラベルをつくってそれを当てはめることができるデータをもってはいる状態
        i. このなんらかのラベルをつくってそれを当てはめること、を事前学習とした転移学習のイメージ
        ii. 具体的には以下のようなタスクがあり、このような単独では意味をなさないタスクをプレテキストタスクと呼ぶ
            1. データのうち一部単語をカットアウトして穴埋めするタスクなど
            2. 画像に幾何学変換をくわえて別の画像と見分けるタスクなど

d. 教師なし学習と同じデータの状態を想定
    i. しかし疑似ラベルをつけて行うプレテキストタスクに対する教師あり学習ではある
    ii. 手持ちのデータに対する学習まで含むと半教師あり学習といえる
    iii. word2vecのようなものを想定していると思うが、目的とするタスクに正解ラベルが必要ない場合もありそれも含めて自己教師学習と呼ぶことがある
        1. しかし、この設定の問題はここでは扱わない

11.5.2 プレテキストタスク

1. 自己教師学習はプレテキストタスクの教師あり学習
    a. そのタスク$\mathcal{T}'$はラベルを自動で付与できるようでなければならない
    b. 目的に対して有用な事前学習のタスクを選ばなければならない
    c. 具体的に、自然言語分野では以下のようなものがある
        i. 完全な文から一部を取り除きその一部を文脈から予測させるタスク
        ii. 2つの文を入力として、その文が実際に連続しているかを予測させるタスク
    d. 音声や画像でも同じようなことは可能
        i. 穴埋め、色情報の予測等、ラベル付与が簡単な内装が基本
        ii. 画像では距離計量学習に近い話題もある

11.5.3 対照表現学習

1. 画像分類のためのプレテキストタスクにはいくつかの方法が知られている
    a. 画像の一部をマスクして周囲の画像からその部分を穴埋めするタスク(inpainting)
    b. カラー画像から色情報を除去してグレースケール画像を作り元の色情報を予測(colorization)
    c. 画像を矩形複数に分割し、順序をランダムに入れ替えたものを入力としてその正しい順序を予測させる(jigsaw puzzle)
    d. 画像分類に対して上の3つの方法はそこまで効果がないが、データ拡張で得た画像と元画像の同一性を予測するタスクなどは効果がある

2. contrasitive representation learningの概要は以下のとおり
    a. 画像分類のプレテキストタスクに紐づいた話題
    b. 対照損失というものを用いる方法
    c. 特徴表現空間を得ることが期待される
3. 方法としては以下のとおり
    a. 基準となるサンプル$\mathbf{x}$、正例$\mathbf{x}^+$、負例$\mathbf{x}_1^-, ..., \mathbf{x}_K^-$d用意する
        i. 基準となるサンプルはクエリとも呼ぶ
        ii. 正例は普通、クエリのもととなる同じと画像に異なる処理をしたもの
        iii. 負例はそれとちがった別の画像複数
    b. 入力$\mathbf{x}$から特徴$\mathbf{z} = \mathbf{f}(\mathbf{x}; \mathbf{w})$を出力するネットワークを考える
        i. 出力はクエリなどでそれぞれ$\mathbf{z} = \mathbf{f}(\mathbf{x}),\ \mathbf{z}^+ = \mathbf{f}(\mathbf{x}^+), \mathbf{z}_i^- = \mathbf{f}(\mathbf{x}_i^-)$となる
        ii. 正例と負例をまとめて$\{\mathbf{x}_i\}_{i=0:K}$、その特徴を$\{\mathbf{z}_i\}_{i=0:K}$、$\mathbf{x}_0$は正例、とする
    c. fをエンコーダと呼ぶことにして、次の損失関数を考える

$$E(w; x, x_i)_{i=0,...,K} = -\log \frac{\exp(\mathrm{sim}(z, z^+)/\tau)}{\sum_{i=0}^K \exp(\mathrm{sim}(z, z_i^-)/\tau)}$$

    d. simは内積とする
    e. 上のような損失関数をInfoNCEとする
    f. 正例に近づけるよう、負例から遠ざかるように学習する
    g. 10.1.2項の数式と同じ
4. InfoNCEにもいい点だけでなく、ほかの手法で改善が試みられている。
    a. 負例が大きいと必要な記憶域が嵩む
    b. 特徴を保持するような方法も計算量の観点から難しい
    c. MoCoは正例と負例の特徴が滑らかに更新されるようにし、負例をキューに保持する
        i. 計算上のミニバッチは必要だがキューの長さとは関係がなく計算量を圧迫しない

    d. simCLRは正例と負例のペアをかなり多くの数準備して用意する
        i. 計算量が問題にならない場合に用いる方法？
    e. いずれも3.のf.を目指して学習する点は共通
5. これまで取り上げた学習方法では、エンコーダの構造に特徴がある
    a. エンコーダには画像分類用CNNがよく用いられるが、これをそのまま使うわけではない
        i. 全結合層を出力側にいくつかはさむ
        ii. この全結合層をprojectorと呼ぶ
6. bootstrap your own latentでは負例を利用しない
    a. 学習は不安定になる
    b. この全結合層をprojectorと呼ぶ
    c. BYOL同様に負例を用いない方法は数多く提案されている
    d. predictorやその周りの変換等の話もありますが、詳細は以降機会があれば補足します

11. 6 マルチタスク学習

11.6.1 概要

1. 同時に複数タスクを学習するというもの
    a. 特徴量セットから別々の複数金融指数(だうと日経平均など)を予測するようなもの？
    b. 定石はなく、個別のケーススタディが大事とのこと
2. 用語的に大事なのはハードシェアリングとソフトシェアリング
    a. ハードではタスク間で同一のネットワークを使った訓練を行う
        i. ハードはネットワークの構造デザインが重要？
    b. ソフトではタスクごとに別々のネットワークを持つ
        i. 学習時にそれらのネットワークのパラメータ同士を制約をつけて訓練する
        ii. ソフトは訓練時の最適化手法に注目している？

c. 損失関数の考え方にも違いがある

    i. ハードは各タスクに重要度をつけて重みづけ

    ii. ソフトは重みパラメータのありようについて損失を定義して、たとえば選んだ層を引かうして重みが近くなるような正則化とともに最小化

d. 以下の図が詳しい

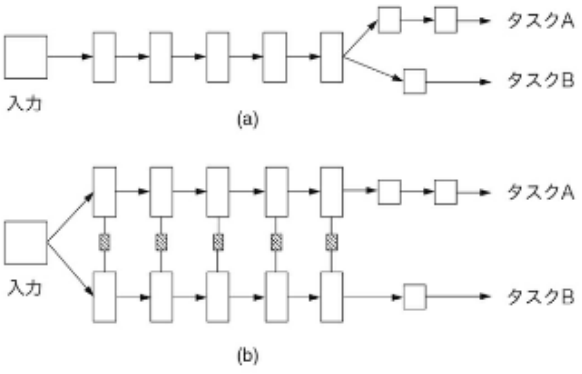

図 11.3 マルチタスク学習の 2 つの方法．(a) ハードシェアリング．(b) ソフトシェアリング．いずれも入力空間を共有する場合．

11.6.1 概要

1. 同時に複数タスクを学習するというもの

    a. 特徴量セットから別々の複数金融指数(だうと日経平均など)を予測するようなもの？

    b. 定石はなく、個別のケーススタディが大事とのこと

2. 用語的に大事なのはハードシェアリングとソフトシェアリング

    a. ハードではタスク間で同一のネットワークを使った訓練を行う

        i. ハードはネットワークの構造デザインが重要？

    b. ソフトではタスクごとに別々のネットワークを持つ

        i. 学習時にそれらのネットワークのパラメータ同士を制約をつけて訓練する

        ii. ソフトは訓練時の最適化手法に注目している？

    c. 損失関数の考え方にも違いがある

        i. ハードは各タスクに重要度をつけて重みづけ

ii. ソフトは重みパラメータのありようについて損失を定義して、たとえば選んだ層を引かうして重みが近くなるような正則化とともに最小化

d. 以下の図が詳しい

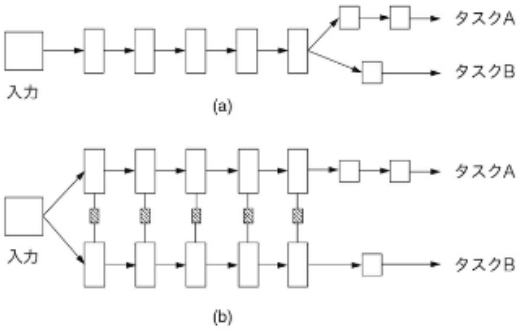

図 11.3 マルチタスク学習の 2 つの方法. (a) ハードシェアリング. (b) ソフトシェアリング. いずれも入力空間を共有する場合.

11. 7 ドメイン適応・汎化

ドメイン適応は、入力の分布が学習時と推論時でことなるような場合を対象押します。

ドメインシフトとは、そのような入力の分布のずれを指します。

もちろん、入力だけでなく、目標出力についても統計的分布がことなるときがあります。

そのような現象を包括的にコンセプトドリフト(データセットシフト)と呼びます。モデルの予測精度は、これが起きた場合低下します。

コンセプトドリフトは以下の二種類があります。

1. 実コンセプトドリフト(クラスドリフト)
    a. 入力で条件付けたクラスラベルの分布がことなるような場合を指します。
        i. 入力の分布は同じでもいいし、違ってもこう呼びます。
    b. いままで有効だった金融の先行指標が有効じゃなくなってきた、のような状況？
2. クラスドリフト仮想コンセプトドリフト(共変ドリフト)
    a. 入力の分布のみが変化する場合

i. 気候変動関連の金融資産価格予測の入力に気候情報をつかっていたが、ここ数年でハチャメチャな気候変動が発生してしまった…といったような場合

11.7.1 データのドメイン

1. データのドメインとは、データの出所を表す概念で数学的には統計的な分布を指す
    a. データのドメインが異なるとは、以下のような例
        i. データの集合が分布Aに従っているが、別のデータの集合が分布Bに従っているとき２つのデータの従っている分布は違う
    b. 訓練時のドメインをソースドメイン、推論時のドメインをターゲットドメインと呼ぶ
    c. 訓練時と推論時にドメインが違うと訓練が役に立たない傾向にある
        i. 今の日経平均とコロナ禍初期の日経平均のリターンの従っている分布など…
        ii. 訓練時に引き続いて推論をうまく行いたい、といった問題をドメイン汎化と呼ぶ
        iii. 転移学習と似ているようで、ドメインの違いについてこだわる点が別物
            1. ただし、転移学習は基本的に同一ドメインを想定している
2. 実際、同じタスクでも別のデータで学習すると困ることがある
    a. 手書き数字の分類だが、データセットが違うMNISTとUSPSの例など

図 11.4　ドメインの違うデータの例．上段：数字認識のデータセット．MNIST，USPS，SVHN[356]．下段：車載カメラ画像のセグメンテーションのデータセット．Cityscapes[357] と GTA5[358]．

b. MNISTで訓練してほぼ正解できるモデルでもUSPSではダメ、など
c. もっと難しいデータセットでも同じことが言える
    i. 合成画像データでも適用できるなドメイン適応手法が欲しい
    ii. 実画像をターゲットにして、合成画像で訓練したモデルの威力を発揮できるととてもよい

11.7.2 教師なしドメイン適応

1. 普通、ドメイン適応は以下のような設定で行われる
    a. ソースドメインについて、ラベルつきのデータセットが十分に存在する
2. 以下のような設定を考え、その条件下の問題をUDAと呼ぶ
    a. ソースドメインについては1.を満たす
    b. ターゲットドメインのデータは、正解ラベルのないデータのみ
3. UDAには以下の種類がある
    a. ソースドメインとターゲットドメインでクラスラベルが違う(閉じたDA)
    b. ソースドメインのクラスラベルがターゲットドメインのクラスラベルをすべて含むが逆はそうでない(部分的DA)
    c. ソースドメインのクラスラベルをターゲットドメインのクラスラベルがすべて含むが逆はそうでない(オープンセットDA)
    d. そのようなクラスラベルのありかたが不明(ユニバーサルDA)
4. UDAの懸念点
    a. ターゲットドメインの正解ラベルつきデータがなければいけない
        i. そもそも性能評価がおぼつかない
        ii. そして、あれば普通転移学習なりで利用するのでUDAの出番ではない
        iii. しかしアイデアに有効な点はあるので紹介している、といったスタンスで以降の内容を記載

11.7.3 敵対的学習

1. 分布の位置合わせを行う方法がある
    a. 入力のソースドメイン、ターゲットドメインの分布にずれがある。

    b. 今回は、その入力$\mathbf{x}$に対して特徴$\mathbf{z}$を使い分布のずれを緩和した予測がしたい。
2. そのために、写像を2つにわける。

$$\mathbf{x} \xrightarrow{\mathbf{z}=\mathbf{g}(\mathbf{x};\mathbf{w}_g)} \mathbf{z} \xrightarrow{y=f(\mathbf{z};\mathbf{w}_f)} y$$

3. 特徴抽出と分類用にネットワークを2つに分ける。
    a. この$\mathbf{g}$をネットワークでうまく学習することで分布のずれを感じないようにする
        i. このために、ソースとターゲットどちらかを予測させる問題をはさむ
        ii. その予測が困難になるように敵対的学習を行い、識別不可能な同じような特徴量を作る
    b. 同じようなものとして扱えるようにしてしまえば、ソースドメインをターゲットドメインのためにうまく使うことができる

11.7.4 疑似ラベルによる方法

1. 設定が半教師あり学習と同じ
    a. ソースドメインが教師付き、ターゲットドメインが教師なしの状況
    b. なので、クラスラベルを疑似ラベル法でつけて実践に持っていける

11.8 少数事例学習

11.8.0 概要

1. 本当にデータ少ないが多クラス分類したい、といったような場合を想定
2. メタ学習の一分野として認識されている
3. 扱うべき用語としては、以下が存在する
    a. few-shot-learning
        i. クラスあたりN個の学習データがあるとN-shot問題と呼ぶ
            1. Kクラス分類、クラスあたりN個の訓練サンプルが与えられるならば$K-shot N-way$問題と呼ばれる

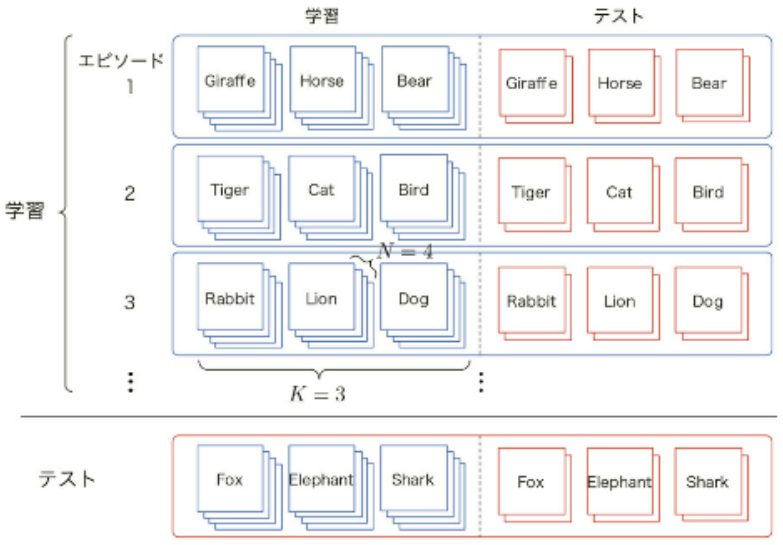

図 11.5 少数事例学習の研究において想定される問題設定.「3-way 4-shot」(クラス数 $K = 3$, クラスあたりの事例数 $N = 4$) の場合.

2. 訓練と性能評価をあわせてエピソードと呼ぶ
3. エピソードの連続で学習が進む
    a. クラスはエピソードごとに異なることもあるのでK(M+N)個のデータが必要なわけではない
    b. 少数事例学習は、このクラスラベルの入れ替わる小規模な学習を多く行うことで進む
4. すごく少ない、とはいえこのくらいがfew-shot-learningの想定範囲

b. prototyical Network

i. 入力を埋め込むネットワークを与える

ii. 学習結果、パラメータを得て少数事例だけ正しく分類できるような特徴空間を作る

iii. 数式的な仕組みは、以下の通り

$$\mathcal{S}_k = \{(\mathbf{x}_{k,i}, d_{k,i})\}_{i=1,\ldots,N}$$

$$\mathbf{c}_k = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}(\mathbf{x}_{k,i};\mathbf{w})$$

$$p(d = k|x) = \frac{\exp(-\mathrm{dist}(f(x;w), c_k))}{\sum_{k'}\exp(-\mathrm{dist}(f(x;w), c_{k'}))}$$

iv. テストサンプルに関しては、上記の仕組み(クラスの平均ベクトルとの比較を使って学習を行うというもの)に対して損失を以下のように定義できる

$$E = -\frac{1}{M}\sum_{i=1}^{M}\log p(d = d_i|x)$$

v. 学習時とテスト時にエピソードが同じ構造を持つ必要はない

vi. 経験的にはN=Mがよく、Kは大きければ大きいほうが良い

vii. メタ学習によって学習スキルを得ようとする仕組み自体には疑問も呈されている

viii. そもそもImageNetなどの事前学習モデルをそのまま使うだけでワークする場合というのもある

c. zero-shot-learning

i. よく聞く単語

ii. 1つのデータもないクラスの画像を新しいクラスの画像として分類する、というような話

4. これは、転移学習の前提を満たさないことを意味する

11.9 active learning

1. 正解ラベルを与えるための学習

a. どのデータに正解ラベルを付与すればよいか？を考える

i. 学習後の推論性能が最大化されるように考える

b. データが整備されてないところだと一番重要な視点かもしれない

i. しかし、ラベルのないデータセットがあることを想定

ii. これらの一部にラベルをつけて、それで満足いく推論性能がでるか？をラベルをつけながら確認していく

iii. 都度ラベル付きデータを増やし、満足いくまでデータを増やす

1. あくまでコストの許す範囲で

2. active learningの流れ

a. ラベル付けは一定数の集合単位で行う

b. 最初は情報がないので、ランダムでつける
c. 残りのサンプルに優先順位をつける
    i. 学習するネットワーク$\mathbf{f}$のモデルに依存しないようなものもある
d. その後優先順位的に上から一定数選んでラベルを付与する
    i. 精度向上の速度が上がるようにするのがactive learning
e. 取得関数を設けて、例えば推論を誤りやすいサンプルをみつけてそれにラベルを付与するなどの施策をする
    i. 予測の不確かさを誤りやすさの代わりに使うことを想定
    ii. 予測の不確かさの算出をする方法は、確信度やエントロピー、MC-dropoutなどを利用する
f. モデルに依存しない方法もある
    i. データの分布を代表するようなデータにラベルを付与する
    ii. コアセットと呼ばれる重要なデータを選びたい
g. これといったベストプラクティスは存在しない
    i. タスクとデータ依存なのがつらい
    ii. ラベルが誤っている、などの可能性もあるなどが難しい
    iii. ラベルを付与するコストに差があるのも難しい
h. ラベル付与の途中段階では半教師あり学習の適用チャンスではあり、その点に新しい進展があるかもしれない