Semi-Supervised Learning and Domain Adaptation in Natural Language Processing

3.2 CLUSTERS-AS-FEATURES
3.3 SEMI-SUPERVISED NEAREST NEIGHBOR

吉田拓夢

3.2 CLUSTERS-AS-FEATURES

- semi-supervised learning algorithms

  →wrapper methods applicable to any supervised learning algorithm

similar

- clusters-as-features

note

stacked learning with a clustering algorithm as label 0 learner

3.2 CLUSTERS-AS-FEATURES

The technique is simple↓

- L:labeled U:unlabeled
- clusteringmodel m_U ←learn from{U or L ∪ U}
- augment every data point x_n ($\in$ L)

with a variable that takes the value $m_U(x_n)$

3.3 SEMI-SUPERVISED NEAREST NEIGHBOR

The next section considers
semi-supervised learning algorithms
that are tied to a particular supervised algorithm

- LABEL PROPAGATION
- SEMI-SUPERVISED NEAREST NEIGHBOR EDITING
- SEMI-SUPERVISED CONDENSED NEAREST NEIGHBOR

3.3.1 LABEL PROPAGATION

- one of the earliest algorithms in the class of graph-based semi-supervised algorithms

- similar to self-training with k-nearest neighbor with weighted voting,

  but with important technical differences

(↑the version using k-nearest neighbor kernel)

3.3.1 LABEL PROPAGATION

- construct a k-nearestneighbor graph
- The algorithm first propagates node labels to neighboringnodes by weighted votes.
- Each node then collects votes on which class it belongs to, including its own, if labeled.

$$w_{ij} = \exp\left(\frac{-E(\mathbf{x}_i, \mathbf{x}_j)^2}{\sigma^2}\right)$$

E:Euclidean distance

E,σ: heuristically set

This is implemented in the SKlearn module semi-supervised.

3.3.2 SEMI-SUPERVISED NEAREST NEIGHBOR EDITING

Nearest neighbor methods ←lazy, impractical

∵no model is learned from the labeled data points

classification time $\propto$ the number of labeled data points

Many algorithms have been proposed to make nearest neighbor learning more efficient.

3.3.2 SEMI-SUPERVISED NEAREST NEIGHBOR EDITING

intuition

only a subset of data points are really important for the decision boundary

The algorithms look for good representatives of clusters in the data or discard the data points that are far from the decision boundaries.

Such algorithms are called editing, or condensation,algorithms

Figures for each algorithms

Dataset $L = \{(y_l, \mathbf{x}_l)\}_{l=1}^{N}, c_{k,L}$
for $n \in N$ **do**
  if $c_{k,L}(\mathbf{x}_n) = y_n$ **then**
    $L = L \setminus \{(y_n, \mathbf{x}_n)\}$
  end if
end for
return $c_{k,L}$

Figure 3.11: Nearest neighbor editing.

Dataset $L = \{(y_l, \mathbf{x}_l)\}_{l=1}^{N}, L' = \emptyset, c_{k,L'}$
for $n \in N$ **do**
  if $c_{k,L'}(\mathbf{x}_n) \neq y_n$ **then**
    $L' = L' \cup \{(\mathbf{x}_n, y_n)\}$
  end if
end for
return $c_{k,L'}$

Figure 3.12: Condensed nearest neighbor.

1: $L = \{(\mathbf{x}_l y_l)\}_l^N, L' = \emptyset, L'' = \emptyset$
2: $U = \{\langle \mathbf{x}'_l \rangle\}_l^M$ # unlabeled data
3: **for** $(\mathbf{x}_n, y_n) \in L$ **do**
4: **if** $c_{k,L'}(\mathbf{x}_n) \neq y_n$ or $P_{c_{k,L'}}((\mathbf{x}_n, y_n)|\mathbf{x}_n) < 0.55$ **then**
5: $L' = L' \cup \{(\mathbf{x}_n, y_n)\}$
6: **end if**
7: **end for**
8: **for** $\langle \mathbf{x}'_l \rangle \in U$ **do**
9: **if** $P_{c_{k,L'}}((\mathbf{x}'_m, c_{k,L'}(\mathbf{x}'_m))) > 0.90$ **then**
10: $L' = L' \cup \{(\mathbf{x}'_m, c_{k,L'}(\mathbf{x}'_m))\}$
11: **end if**
12: **end for**
13: **for** $(\mathbf{x}_n, y_n) \in L'$ **do**
14: **if** $c_{k,L''}(\mathbf{x}_n) \neq y_n$ **then**
15: $L'' = L'' \cup \{(\mathbf{x}_n, y_n)\}$
16: **end if**
17: **end for**
18: **return** $c_{k,L''}$

Figure 3.13: Semi-supervised condensed nearest neighbor.

3.3.3 SEMI-SUPERVISED CONDENSED NEAREST NEIGHBOR

- CNN can be seen as a simple technique for approximating such a subset of labeled data points.
- Ideally, CNN returns one point for each cluster, namely the center of each cluster.
- CNN sometimes needs several points to stabilize the representation of a cluster.

∵ a sample of labeled data may not include data points that are near the center of a cluster.