Connections to Machine Learning,1 (Semi-Supervised Learning)

22191160 김민수

BIOSTATISTICS

reference

Elements of Casual Inference (Jonas Peters, Dominik Janzing, Bernhard Scholkopf)

On Causal and Anticausal Learning (Bernhard Scholkopf, Donminik Zanzing, Jonas Peters Eleni Sgouritsa, Kun Zhang, Joris Mooij)

Semi-Supervised Learning (Oliver Chapelle, Bernhard Scholkopf, Alexander Zien)

What is Semi-supervised Learning

1. **Semi-Supervised Learning(SSL) is halfway between supervised and unsupervised Learning**

2. History of Semi-Supervised Learning

   Self-learning(Wrapper Algorithm) -> Transductive inference -> Mixture of Gaussians -> Theoretical Analysis -> text applications

3. In a more mathematical formulation, one could say that the knowledge on p(x) that one gains through the unlabeled data has to carry information that is useful in the inference of p(y|x)

4. **If this is not the case, semi-supervised learning will not yiled an improvement over supvervised learning. It might even happen that using the unlabeled data degrades the prediction accuracy by misguiding the inference.(assumption hold)**

Semi-supervised smoothness assumption

-The assumption is that the label function is smoother in high-density regions than in low-density regions.

-If two points x1, x2 in a high-density region are close, then so should be the corresponding outputs y1,y2.

-This assumption implies that if two points are linked by a path of high density(e.g., if they belong to the same cluster), then their outputs are likely to be close.

- At present, it is less clear how useful the assumption is for regression problems.

Cluster assumption

- Suppose we knew that the points of each class tended to form a cluster. Then the unlabeled data could aid in finding the boundary of each cluster more accurately.

- Cluster assumption : If points are in the same cluster, they are likely to be of the same class.

- It only means that, usually, we do not observe objects of two distinct classes in the same cluster.

- The Cluster assumption can be formulated in an equivalent way.

- Low density separation : The decision boundary should lie in a low-density region.

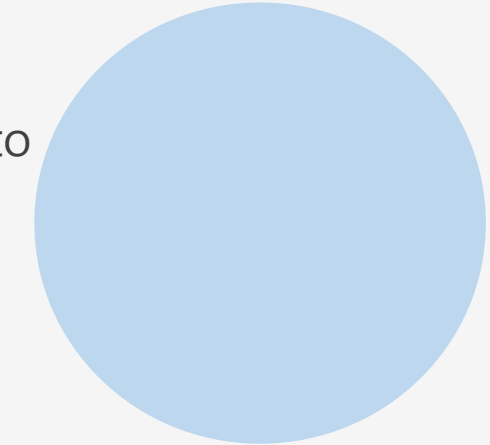
- Although the two formulations are conceptually equivalent, they can inspire different algorithms. (ex. Generative model, TSVM)

- And it is sensible in many real world problem. (ex. Handwriting 0, 1)

Usefulness of Semi-Supervised Learning

1. In speech recognition, it costs almost nothing to record huge amounts of speech, but labeling it requires some human to listen to it and type a transcript.

2. Billions of webpages are directly available for automated processing, but to classify them reliably, humans have to read them.



3. Protein sequences are nowadays acquired at industrial speed(by genome sequencing, computational gene finding, and automatic translation), but to resolve a three-dimensional (3D) structure or to determine the functions of a single protein may require years of scientific work.

Causality and Semi-Supervised Learning

1. It has been argued that statistical associations are always due to underlying causal structures.

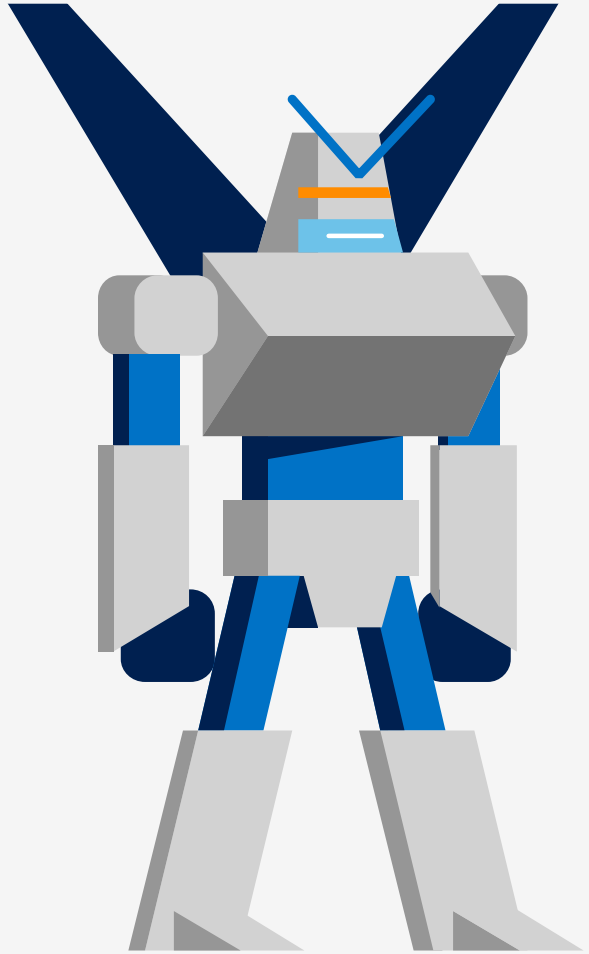
2. The functional point of view(SCM or SEM) allows us to come up with assumptions on causal models that would be harder to conceive in a pure probabilistic view.(Markov assumption, faithfulness) and such assumption allow us distinguish X->Y and X<-Y (eg. Dhsic)

3. Independence of mechanism and input : P(cause) and P(effect|cause) do not contain information about one another.

Simulation(TSVM, transductive SVM)

Maximum Margin Algorithm: Algorithm which try to directly implement the low-density separation assumption by pushing the decision boundary away from the unlabeled points.

1. Starting from the SVM solution as trained on the labeled data only.
2. The unlabeled points are labeled by SVM predictions.
3. **And the SVM is retrained on all points. This is iterated while the weight of the unlabeled points is slowly increased.**



■ *Large-margin inductive SVM with the universum*

Minimize the functional

$$R(w) = (w, w) + C_1 \sum_{i=1}^{\ell} \xi_i, + C_2 \sum_{s=1}^{u} \xi_s^*, \quad C_1 \geq 0$$

subject to the constraints (24.41) and (24.42).

Simulation1 – mushroom data (anticausal/confounded case)

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. gill-spacing: close=c,crowded=w,distant=d
8. gill-size: broad=b,narrow=n
9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
10. stalk-shape: enlarging=e,tapering=t
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

Data from UCI repository

1 Data setting

-"?" of V12 is removed, V17 is removed since it has only one value.

-300 train data(100 of them are Dl, 200 of them are Du)

-700 test data

2 **TUNING(10 FOLD CV)**

KERNEL is linear. Decide C(cost parameter of svm), Cstar(cost parameter of unlabeled objects.)of TSVM And C of SVM. After 10 fold cv, Select (0.1, 0.001), 0.1 respectively.

3 TEST DATA

ERROR(SVM) : 0.9814286

ERROR(TSVM) : 0.9828571

Simulation2 – Iris data (anticausal/confounded case)

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

Data from UCI repository

1 Data setting

-remove category "Iris-Setosa" in target variable for setting binary variable.

-20 train data(10 of them are Dl, 10 of them are Du)

-80 test data

2 **TUNING(10 FOLD CV)**

KERNEL is linear. Decide C(cost parameter of svm), Cstar(cost parameter of unlabeled objects.)of TSVM And C of SVM. After 10 fold cv, Select (0.1, 1), 0.1 respectively.

3 TEST DATA

ERROR(SVM) : 0.775

ERROR(TSVM) : 0.8625

Simulation3 – splice data (causal case)

Attribute Information:

1. One of {n ei ie}, indicating the class.
2. The instance name.
3-62. The remaining 60 fields are the sequence, starting at position -30 and ending at position +30. Each of these fields is almost always filled by one of {a, g, t, c}. Other characters indicate ambiguity among the standard characters according to the following table:

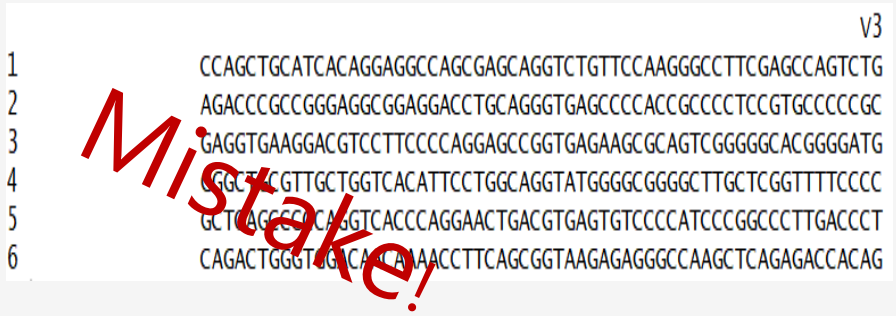
Data from UCI repository

1 Data setting

-remove category "IE" in target variable for setting binary variable. And Let V2 and V3 feature.(mistake)

-50 train data(20 of them are DI, 30 of them are Du)

-150 test data



2 **TUNING(10 FOLD CV)**

KERNEL is linear. Decide C(cost parameter of svm), Cstar(cost parameter of unlabeled objects.)of TSVM And C of SVM. After 10 fold cv, Select (1, 0.001), 0.001 respectively.

3 TEST DATA

ERROR(SVM) : 0.673333

ERROR(TSVM) : 0.673333

Simulation4 – balance scale (causal case)

Attribute Information:

1. Class Name: 3 (L, B, R)
2. Left-Weight: 5 (1, 2, 3, 4, 5)
3. Left-Distance: 5 (1, 2, 3, 4, 5)
4. Right-Weight: 5 (1, 2, 3, 4, 5)
5. Right-Distance: 5 (1, 2, 3, 4, 5)

Data from UCI repository

1 Data setting

-remove category "R" in target variable for setting binary variable.

-150 train data(50 of them are Dl, 100 of them are Du)

-150 test data

2 **TUNING(10 FOLD CV)**

KERNEL is linear. Decide C(cost parameter of svm), Cstar(cost parameter of unlabeled objects.)of TSVM And C of SVM. After 10 fold cv, Select (0.01, 0.01), 0.01 respectively.

3 TEST DATA

ERROR(SVM) : 0.873333

ERROR(TSVM) : 0.873333

Final Project Plan

-Ensemble and Semi-Supervised Learning

-Multinomial target variable and Semi-Supervised Learning

Thank you!