Machine Learning

Introduction (continued)

Resources: Journals

- Journal of Machine Learning Research www.jmlr.org
- Machine Learning
- IEEE Transaction on Neural Networks
- IEEE Transaction on Pattern Analysis and Machine Intelligence
- Annals of Statistics
- Journal of the American Statistical Association

Resources: Conferences

- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Neural Information Processing Systems (NIPS)
- Computational Learning
- International Joint Conference on Artificial Intelligence (IJCAI)
- ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)
- IEEE Int. Conf. on Data Mining (ICDM)

Framework

- 4 for Deep Learning:
  - TensorFlow (+Keras), MXNet, caffe, Theano
- 5 for generic machine learning:
  - scikit-learn, Spark MLLib, mlpack, Accord.NET Framework, Weka/Meka
- 3 for cloud (actually those are not frameworks):
  - Amazon Machine Learning
  - Google Cloud Machine Learning
  - Azure ML Studio

Covered Topics

- Concept learning
- Linear Regression
- Logistic Regression
- Naïve Bayes
- Decision Tree
- Support Vector Machine
- K-nearest Neighbors
- Neural Network
- K-means
- Feature Reduction
- ML Techniques
- Text Mining
- Big Data Framework

Classification Tasks

- Binary classification
- Multi-class classification
  - One-Vs-Rest
- Multi-label classification
  - Binary Relevance

Concept Learning and The General-To-Specific Ordering

Concept Learning

- Acquire **general** concepts from **specific** training examples.
- Each such concept can be viewed as describing some subset of objects or events defined over a larger set.

Concept Learning Task

- Target Concept: Days on which my friend enjoys his water sport (Attribute: EnjoySport)
- Task: Predict the value of EnjoySport based on the values of other attributes.
- Positive and negative training examples for the target concept EnjoySport

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Concept Learning Task

- Hypothesis Representation
  - A simple representation: each hypothesis consists of a conjunction of constraints on the instance attributes.
  - Each hypothesis be a vector of six constraints.
    - "**?**" indicate any value is acceptable for this attribute
    - A single required value for the attribute
    - **ϕ** indicate that no value is acceptable
  - Example
    - <?, Cold, High, ?, ?, ?>
    - <?, ?, ?, ?, ?, ?>
    - < ϕ, ϕ, ϕ, ϕ, ϕ, ϕ>

Notations

- Instance ***x***
- A set of instances ***X***
- Target Concept ***c***: The concept or function to be learned
- Hypothesis ***h***: a Boolean-valued function defined over ***X***. The goal of the learner is to find a hypothesis ***h*** such that ***h(x)=c(x)***
- Hypothesis Set ***H***
- Positive examples
- Negative examples
- Training dataset ***D***

Concept Learning Task

- Given
  - Instances X: Possible days, each described by the attributes
    - *Sky (with possible values Sunny, Cloudy, and Rainy)*
    - *AirTemp (with values Warm and Cold)*
    - *Humidity (with values Normal and High)*
    - *Wind (with values Strong and Weak)*
    - *Water (with values Warm and Cold), and*
    - *Forecast (with values Same and Change).*
  - Hypotheses ***H***: Each hypothesis is described by a conjunction of constraints on the attributes Sky, AirTemp, Humidity, Wind, Water and Forecast. The constraints may be "?" , ϕ or a specific value.
  - Target Concept ***c***: EnjoySport: X$\rightarrow$ {0,1}
  - Training Examples ***D***: Positive and negative examples of the target function
- Determine
  - A hypothesis ***h*** in ***H*** such that ***h(x)=c(x)*** for all ***x*** in ***X***

Inductive Learning Hypothesis

- Inductive Learning
  - Determine a hypothesis ***h*** over set of instances ***X***
- Inductive
  - Guarantee that the output hypothesis fits the target concept over the training data
- Fundamental Assumption
  - Hypothesis that best fits the observed training data
- The Inductive Learning Hypothesis
  - Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over unobserved examples.

General-to Specific Ordering of Hypotheses

- Two hypotheses
    - h_1=<Sunny, ?, ?, Strong, ?, ?>
    - h_2=<Sunny, ?, ?, ?, ?, ?>
  - Any instance classified positive by $\boldsymbol{h_1}$ will also be classified positive by $\boldsymbol{h_2}$, thus $\boldsymbol{h_2}$ is more general than $\boldsymbol{h_1}$

Find-S Algorithm

- Initialize ***h*** to the most specific hypothesis in ***H***
- For each positive training instance ***x***
  - For each attribute constraint a_i in h
    - If the constraint a_i is satisfied by ***x***
    - Then do nothing
    - Else replace a_i in ***h*** by the next more general constraint that is satisfied by ***x***
- Output hypothesis ***h***

Find-S: Finding a Maximally Specific Hypothesis

- Find-S on EnjoySport Task
  - $\boldsymbol{h_0} \leftarrow <\phi, \phi, \phi, \phi, \phi, \phi>$
  - $\boldsymbol{h_1} \leftarrow$<Sunny, Warm, Normal, Strong, Warm, Same>
  - $\boldsymbol{h_2} \leftarrow$<Sunny, Warm, ?, Strong, Warm, Same>
  - $\boldsymbol{h_3} \leftarrow$ The algorithm makes no change to ***h*** when encountering a negative example
  - $\boldsymbol{h_4} \leftarrow$<Sunny, Warm, ?, Strong, ?, ?>

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Find-S: Finding a Maximally Specific Hypothesis

- Find-S is guaranteed to output the most specific hypothesis within ***H*** that is consistent with the positive training examples.
  - Why anyways prefer the most specific hypothesis? Maybe a more general or something in between could be better.
  - The algorithm finds and returns only one hypothesis. But there usually are other hypotheses consistent with the training data. Why discard them?

Candidate-Elimination Algorithm

- Find-S only output one of the correct hypotheses
- The output of Candidate-Elimination Algorithm are all the hypotheses consistent with training data.
- Initialize ***G*** to the set of maximally general hypotheses in ***H***
- Initialize ***S*** to the set of maximally specific hypotheses in ***H***

Candidate-Elimination Algorithm

- For each training example d, do
  - If ***d*** is a ***positive*** example
    - Remove from ***G*** any hypothesis inconsistent with ***d***
    - For each hypothesis ***s*** in ***S*** that is not consistent with ***d***
      - Remove ***s*** from ***S***
      - Add to ***S*** all minimal generalizations ***h*** of ***s*** such that
        - » ***h*** is consistent with ***d***, and some member of ***G*** is more general than h
      - Remove from ***S*** any hypothesis that is more general than another hypothesis in ***S***
  - If ***d*** is ***negative*** example
    - Remove from ***S*** any hypothesis inconsistent with ***d***
    - For each hypothesis ***g*** in ***G*** that is not consistent with ***d***
      - Remove ***g*** from ***G***
      - Add to ***G*** all minimal specializations ***h*** of g such that
        - » ***h*** is consistent with ***d***, and some member of ***S*** is more specific than ***h***
      - Remove from ***G*** any hypothesis that is less general than another hypothesis in ***G***

An Illustrative Example

S_0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

↓

S_1: {<*Sunny, Warm, Normal, Strong, Warm, Same*>}

↓

S_2: {<*Sunny, Warm, ?, Strong, Warm, Same*>}

↓

G_0,G_1,G_2: {<*?, ?, ?, ?, ?, ?*>}

Training examples:
1. *<Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport=Yes*
2. *<Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport=Yes*

S_1: {<*Sunny, Warm, Normal, Strong, Warm, Same*>}

↓

S_2: {<*Sunny, Warm, ?, Strong, Warm, Same*>}

G_1,G_2: {<*?, ?, ?, ?, ?, ?*>}

Training examples:
1. *<Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport=Yes*
2. *<Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport=Yes*

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

An Illustrative Example

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

***S_2*, *S_3*:** {< *Sunny*, *Warm*, *?*, *Strong*, *Warm*, *Same* >}

***G_3*:** {< *Sunny*, *?*, *?*, *?*, *?*, *?* > < *?*, *Warm*, *?*, *?*, *?*, *?* > < *?*, *?*, *?*, *?*, *?*, *Same* >}

***G_2*:** {< *?*, *?*, *?*, *?*, *?*, *?*>}

An Illustrative Example

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

S_3 : {< *Sunny*, *Warm*, ?, *Strong*, *Warm*, *Same* >}

S_4 : {< *Sunny*, *Warm*, ?, *Strong*, ?, ? >}

G_4 : {< *Sunny*, ?, ?, ?, ?, ? > < ?, *Warm*, ?, ?, ?, ? > }

G_3 : {< *Sunny*, ?, ?, ?, ?, ? > < ?, *Warm*, ?, ?, ?, ? > < ?, ?, ?, ?, ?, *Same* >}

An Illustrative Example

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes