Fundamentals of Machine Learning

Il-Chul Moon
Dept. of Industrial and Systems Engineering
KAIST
icmoon@kaist.ac.kr

Weekly Objectives

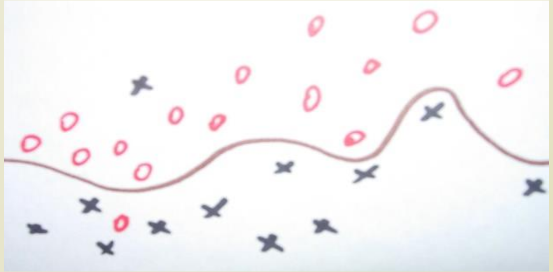
- Learn the most classical methods of machine learning
  - Rule based approach
  - Classical statistics approach
  - Information theory appraoch
- Rule based machine learning
  - How to find the specialized and the generalized rules
  - Why the rules are easily broken
- Decision Tree
  - How to create a decision tree given a training dataset
  - Why the tree becomes a weak learner with a new dataset
- Linear Regression
  - How to infer a parameter set from a training dataset
  - Why the feature engineering has its limit

RULE BASED MACHINE LEARNING

From the Last Week

Supervised Learning

You know the true answers of some of instances



- Definition of machine learning
  - A computer program is said to
    - learn from experience E
    - With respect to some class of tasks T
    - And performance measure P, if its performance at tasks in T, as measured by P, improves with experience E
- More experience → more thumbtack toss, more prior knowledge
  - Data: We have observed the sequence data of D with a_H and a_T
  - Our hypothesis
    - The gambling result of thumbtack follows the binomial distribution of θ
- Our first trial other than thumbtack
  - Rule based learning
  - Still, about choosing a better hypothesis

A Perfect World for Rule Based Learning

- Imagine
  - A perfect world with
    - No observation errors, No inconsistent observations
    - No stochastic elements in the system we observe
    - Full information in the observations to regenerate the system
  - A perfect world of "EnjoySport"
- Observation on the people
  - Sky, Temp, Humid, Wind, Water, Forecast → EnjoySport

Training data is error-free, noise-free

Target function is deterministic

Target function is contained in hypotheses set

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Function Approximation

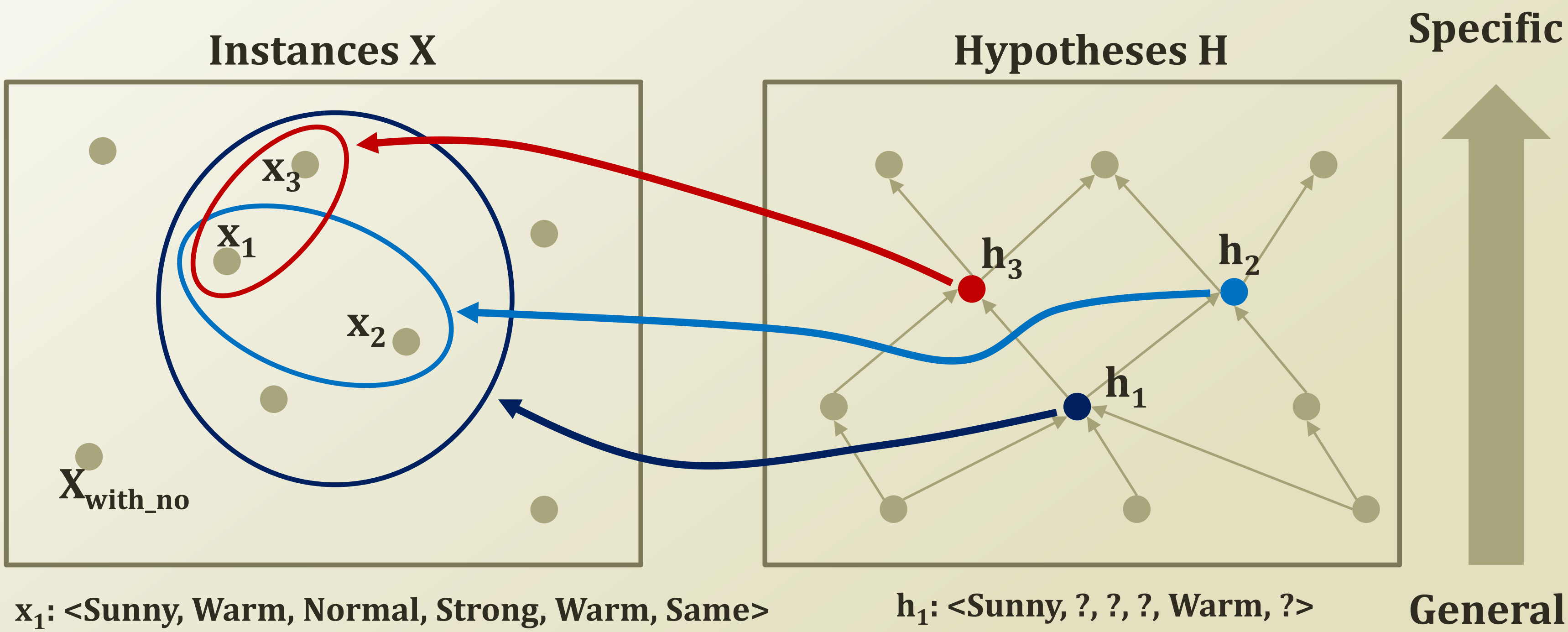
- Machine Learning?
  - The effort of producing a better approximate function
  - Remember PAC Learning Theory?
- In the perfect world of EnjoySport
  - Instance ***X***
    - Features: O: <Sunny, Warm, Normal, Strong, Warm, Same>
    - Label: Y: <Yes>
  - Training Dataset ***D***
    - A collection of observations on the instance
  - Hypotheses ***H***
    - Potentially possible function to turn X into Y
    - h_i: <Sunny, Warm, ?, ?, ?, Same> → Yes
    - How many hypotheses exist?
  - Target Function ***c***
    - Unknown target function between the features and the label

Determine
A hypothesis ***h*** in ***H*** such that ***h(x)=c(x)*** for all ***x*** in ***X***

↓

Determine
A hypothesis ***h*** in ***H*** such that ***h(x)=c(x)*** for all ***x*** in ***D***

Graphical Representation of Function Approximation



x_1: <Sunny, Warm, Normal, Strong, Warm, Same>

x_2: <Sunny, Warm, Normal, Light, Warm, Same>

x_3: <Sunny, Warm, Normal, Strong, Warm, Change>

h_1: <Sunny, ?, ?, ?, Warm, ?>

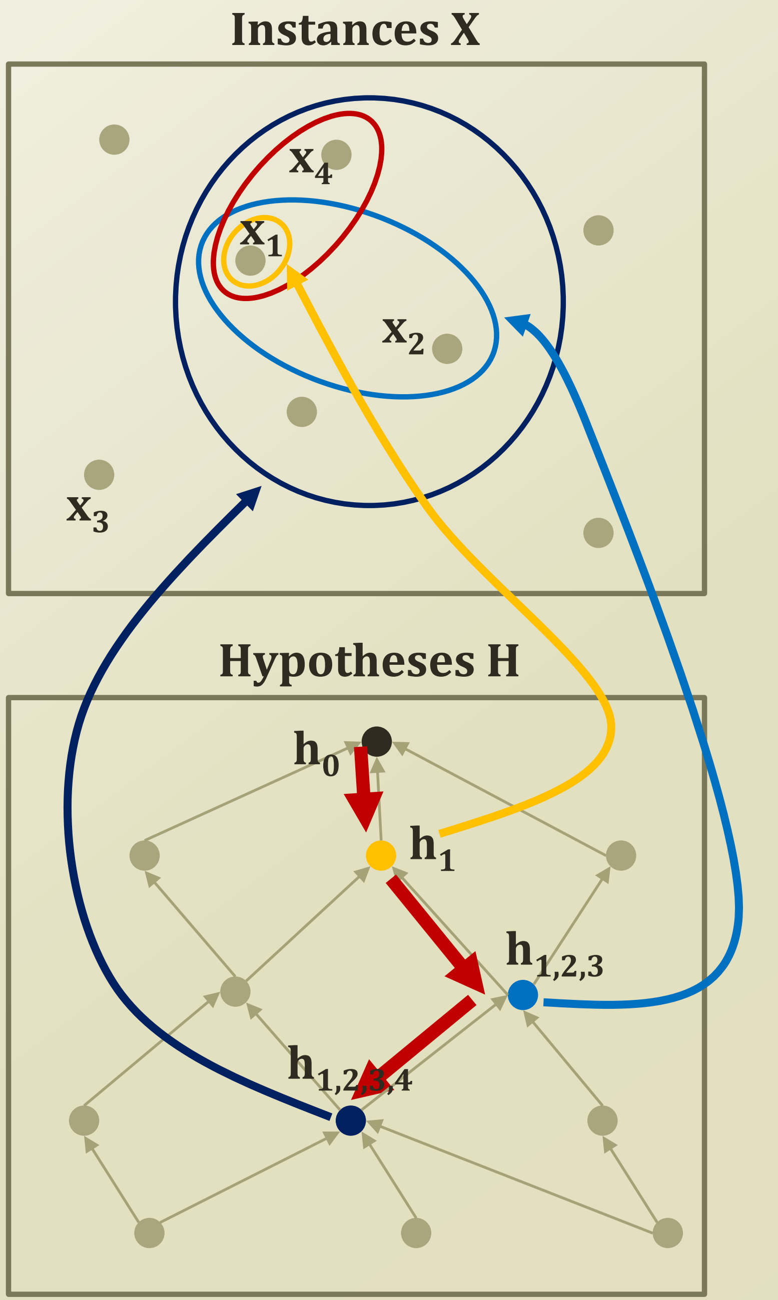
h_2: <Sunny, ?, ?, ?, Warm, Same>

h_3: <Sunny, ?, ?, Strong, Warm, ?>

- What would be the better function approximation?
  - Generalization vs. Specialization

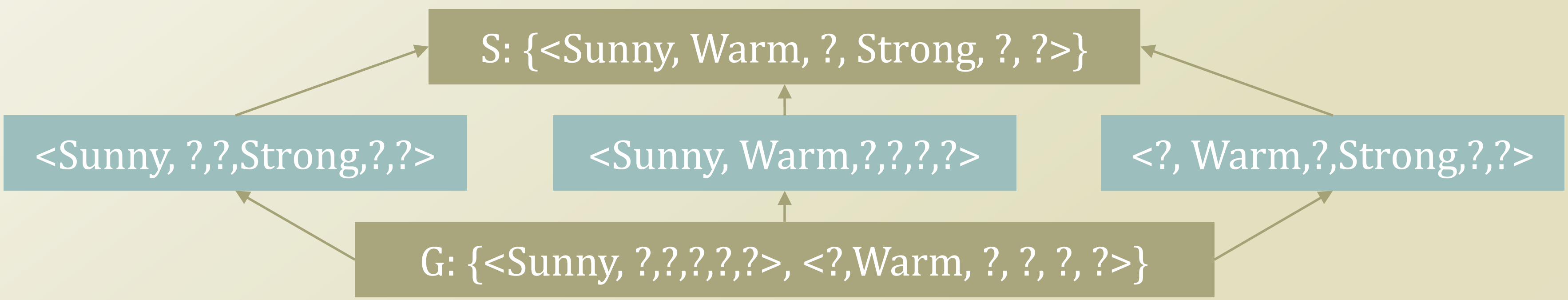
Find-S Algorithm

- Find-S Algorithm
  - Initialize h to the most specific in H
  - For instance x in D
    - if x is positive
      - For feature f in O
        - If f_i in h == f_i in x
          - Do nothing
        - Else
          - f_i in h = f_i in h ∪ f_i in x
  - Return h
- Instances
  - x_1: <Sunny, Warm, Normal, Strong, Warm, Same>
  - x_2: <Sunny, Warm, Normal, Light, Warm, Same>
  - x_4: <Sunny, Warm, Normal, Strong, Warm, Change>
- Hypotheses
  - h_0=<Ø, Ø, Ø, Ø, Ø, Ø>
  - h_1=<Sunny, Warm, Normal, Strong, Warm, Same>
  - $h_{1,2,3}$=<Sunny, Warm, Normal, ?, Warm, Same>
  - $h_{1,2,3,4}$=<Sunny, Warm, Normal, ?, Warm, ?>
- Any problems?
  - Many possible hs, and can't determine the converge



Version Space

- Many hypotheses possible, and No way to find the convergence
- Need to setup the perimeter of the possible hypothesis
- The set of the possible hypotheses == Version Space, ***VS***
  - General Boundary, ***G***
    - Is the set of the maximally general hypotheses of the version space
  - Specific Boundary, ***S***
    - Is the set of the maximally specific hypotheses of the version space
  - Every hypothesis, ***h***, satisifies
    - $\mathrm{VS}_{\mathrm{H,D}}=\{h \in H \mid \exists s \in S, \exists g \in G, g \geq h \geq s\}$
      $where\ x \geq y\ means\ x\ is\ more\ general\ or\ equal\ to\ y$



Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Candidate Elimination Algorithm

- Candidate Elimination Algorithm
  - Initialize S to maximally specific h in H
  - Initialize G to maximally general h in H
  - For instance x in D
    - If y of x is positive
      - Generalize S as much as needed to cover o in x
      - Remove any h in G, for which h(o)≠y
    - If y of x is negative
      - Specialize G as much as needed to exclude o in x
      - Remove any h in S, for which h(o)=y
  - Generate h that satisfies $\exists s \in S, \exists g \in G, g \geq h \geq s$

S0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

G0: {<?,?,?,?,?,?>}

Progress of Candidate Elimination Algorithm

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

S0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

↓

S1: {<Sunny, Warm, Normal, Strong, Warm, Same>}

↓

S2: {<Sunny, Warm, ?, Strong, Warm, Same>}

G0, G1, G2: {<?,?,?,?,?,?>}

Progress of Candidate Elimination Algorithm

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

S0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

S1: {<Sunny, Warm, Normal, Strong, Warm, Same>}

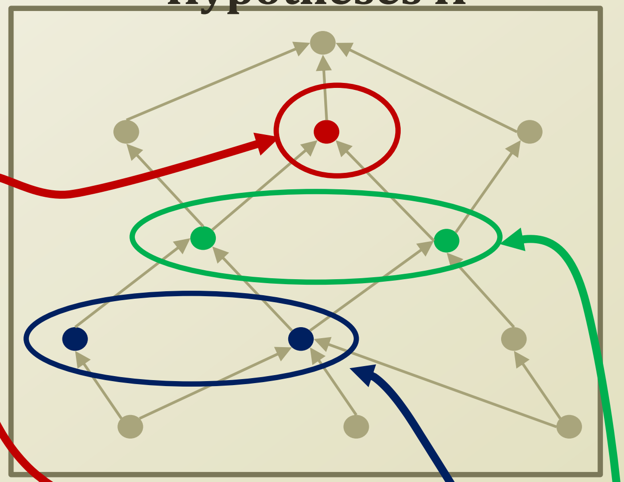
S2, S3: {<Sunny, Warm, ?, Strong, Warm, Same>}

G3: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>, <?,?,?,?,?,Same>}

G0, G1, G2: {<?,?,?,?,?,?>}

Progress of Candidate Elimination Algorithm

Hypotheses H



Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

S0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

S1: {<Sunny, Warm, Normal, Strong, Warm, Same>}

S2, S3: {<Sunny, Warm, ?, Strong, Warm, Same>}

S4: {<Sunny, Warm, ?, Strong, ?, ?>}

Still many ***h*s**

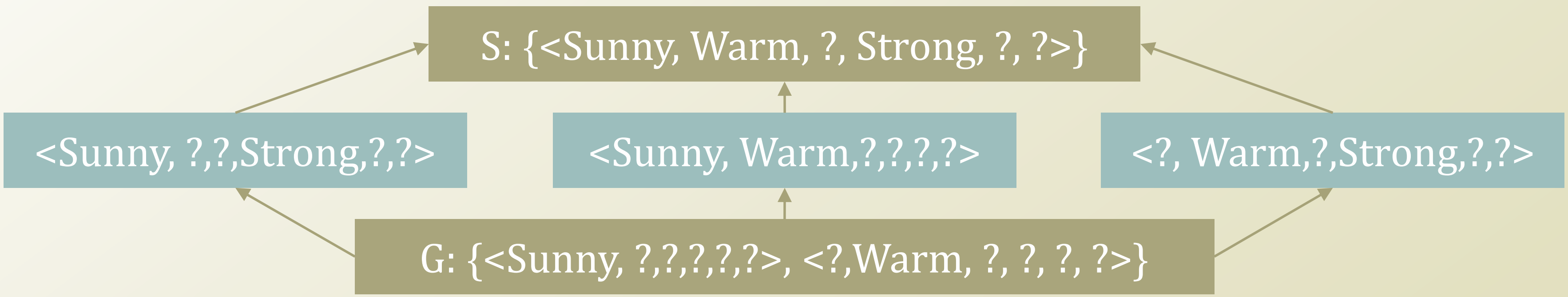
G4: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>}

G3: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>, <?,?,?,?,?,Same>}

G0, G1, G2: {<?,?,?,?,?,?>}

How to classify the next instance?

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes



- Somehow, we come up with the version space
  - A subset of ***H*** that satisfies the training data, ***D***
- Imagine a new instance kicks in
  - <Sunny, Warm, Normal, Strong, Cool, Change>
  - <Rainy, Cold, Normal, Light, Warm, Same>
  - <Sunny, Warm, Normal, Light, Warm, Same>
- How to classify these?
  - Which ***h*** to apply from the subset?
  - Or, a classification by all of ***h***s in the subset
  - How many are ***h***s satisfied?

Is this working?

- Will the candidate-elimination algorithm converge to the correct hypothesis?
  - Converge? → Able to select a hypothesis
  - Correct? → The hypothesis is true in the observed system
- Given the assumption, yes and yes
  - No observation errors, No inconsistent observations
  - No stochastic elements in the system we observe
  - Full information in the observations to regenerate the system
- However, we don't live in the perfect world
  - Any noise in ***o*** of ***x*** in ***D***
  - Decision factor other than ***o*** of ***x***
  - → **a correct *h* can be removed by the noise**
  - → **Cannot say yes and no**

Training data is error-free, noise-free

Target function is deterministic

Target function is contained in hypotheses set