Version Space Learning

Refining the Hypothesis Space

The result of learning class membership can be represented by a Boolean concept.

c(X): {individuals} → {T, F}

We consider Boolean concepts which can be expressed as a Boolean function of a fixed set of predicates $p_i(X)$ which can be evaluated over individuals.

Example: $c(X) = (p_1(X) \wedge \neg p_2(X)) \vee \neg(p_5(X) \wedge p_8(X))$

The space of all c(X) is called hypothesis space.

The learning procedure has the task to search for a c(X) which correctly classifies given positive and negative examples.

Example for Concept Learning

Reading data of a person:

article	crime	academic	local	music	read
a1	T	F	F	T	T
a2	T	F	F	F	T
a3	F	T	F	F	F
a4	F	F	T	F	F
a5	T	T	F	F	T

Search for concept read(X)
based on predicates {crime(X), academic(X), local(X), music(X)}.

Example result:
read(X) = crime(X)

Version Space Learning

Hypothesis space can be partially ordered using the specialization relation:

h2 is more specific than h1 if h1(a) → h2(a) for all individuals A.

The version space is the subspace of the hypothesis space which is consistent with the training examples.

The general boundary G of a version space is the set of maximally general members of the version space.

The specific boundary S of a version space is the set of maximally specific members of the version space.

General and specific boundary completely determine the version space.

Candidate Elimination Algorithm

Let G = {T} and S = {F}

For each example e:

- If e is a positive example:
  - Remove the elements of G from G which classify e as negative
  - Remove each element s of S that classifies e as negative and replace it by the minimal generalizations of s that classify e positive and are less general or equal to some member of G
  - Nonmaximal hypotheses are removed from S
- If e is a negative example:
  - Remove the elements of S from S which classify e as positive
  - Remove each element g of G that classifies e as positive and replace it by the minimal specializations of g that classify e as negative and are more general or equal to some member of S
  - Nonminimal hypotheses are removed from G

Version Space for Example

Examples

0	**G0 = {T}** **S0 = {F}**
1	**G1 = {T}** **S1 = {crime ^ ¬academic ^ ¬local ^ music}**
2	**G2 = {T}** **S2 = {crime ^ ¬academic ^ ¬local}**
3	**G3 = {crime, ¬academic}** **S3 = {crime ^ ¬academic ^ ¬local}**
4	**G4 = {crime, ¬academic ^ ¬local}** **S4 = {crime ^ ¬academic ^ ¬local}**
5	**G5 = {crime}** **S3 = {crime ^ ¬local}**

Learning Poisenous Mushrooms (1)

Classifying mushrooms as either poisonous or not

Mushroom description:

Attribute	**Possible values**
Colour	{Red, Grey}
Size	{Small, Large}
Shape	{rOund, Elongated}
Environment	{Humid, Dry}
Height	{LoW, hIgh}
Texture	{sMooth, roUgh}
Class	{Poisenous, Non-poisenous}

Learning Poisenous Mushrooms (2)

Training examples:

Colour	Size	Shape	Environ.	Height	Texture	Class
({R},	{S},	{O},	{H},	{W},	{M}),	P
({R},	{S},	{E},	{H},	{W},	{M}),	P
({G},	{L},	{E},	{H},	{W},	{U}),	N
({R},	{S},	{E},	{H},	{I},	{U}),	P

Learning Poisenous Mushrooms (3)

Initial Version Space:

G: {({R, G}, {S, L}, {O, E}, {H, D}, {W, I}, {M, U})}

specialize

S: {({ }, { }, { }, { }, { }, { })}

generalize

Learning Poisenous Mushrooms (4)

Example 1: ({R}, {S}, {O}, {H}, {W}, {M}), P

G: {({R, G}, {S, L}, {O, E}, {H, D}, {W, I}, {M, U})}

specialize

{({R}, {S}, {O}, {H}, {W}, {M})}

S: {({ }, { }, { }, { }, { }, { })}

generalize

Learning Poisenous Mushrooms (5)

Example 2: ({R}, {S}, {E}, {H}, {W}, {M}), P

G: {({R, G}, {S, L}, {O, E}, {H, D}, {W, I}, {M, U})}

specialize

{({R}, {S}, {O, E}, {H}, {W}, {M})}

{({R}, {S}, {O}, {H}, {W}, {M})}

S: {({ }, { }, { }, { }, { }, { })}

generalize

Learning Poisenous Mushrooms (6)

Example 3: ({G}, {L}, {E}, {H}, {W}, {U}), N

G: {({R, G}, {S, L}, {O, E}, {H, D}, {W, I}, {M, U})}

specialize

{({R}, {S, L}, {O, E}, {H, D}, {W, I}, {M, U}),
({R, G}, {S}, {O, E}, {H, D}, {W, I}, {M, U}),
({R, G}, {S, L}, {O, E}, {H, D}, {W, I}, {M})}

{({R}, {S}, {O, E}, {H}, {W}, {M})}

{({R}, {S}, {O}, {H}, {W}, {M})}

S: {({ }, { }, { }, { }, { }, { })}

generalize

Learning Poisenous Mushrooms (7)

Example 4: ({R}, {S}, {E}, {H}, {I}, {U}), P

G: {({R, G}, {S, L}, {O, E}, {H, D}, {W, I}, {M, U})}

specialize

{({R}, {S, L}, {O, E}, {H, D}, {W, I}, {M, U}),
({R, G}, {S}, {O, E}, {H, D}, {W, I}, {M, U}),
({R, G}, {S, L}, {O, E}, {H, D}, {W, I}, {M})}

{({R}, {S}, {O, E}, {H}, {W, I}, {M, U})}
{({R}, {S}, {O, E}, {H}, {W}, {M})}
{({R}, {S}, {O}, {H}, {W}, {M})}

S: {({ }, { }, { }, { }, { }, { })}

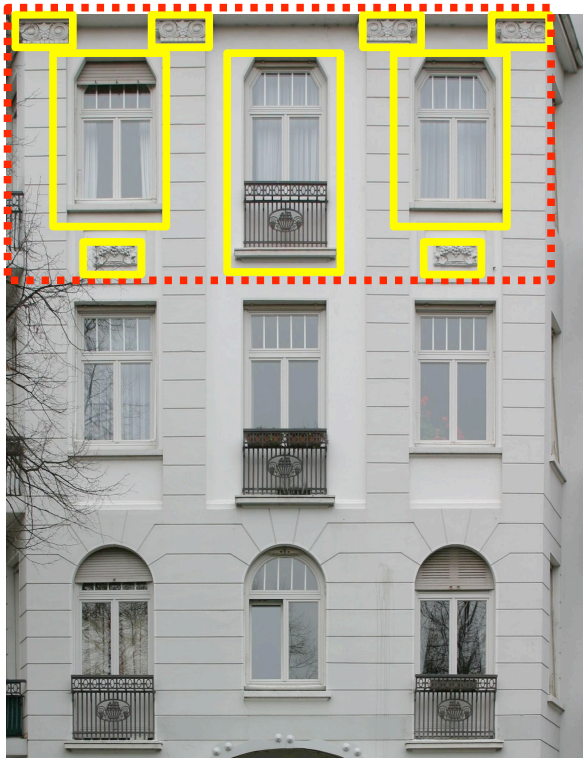
generalize

Extending Version Space Learning (1)

Incremental refinement of conceptual description of "roof"

positive example #1

positive example #2



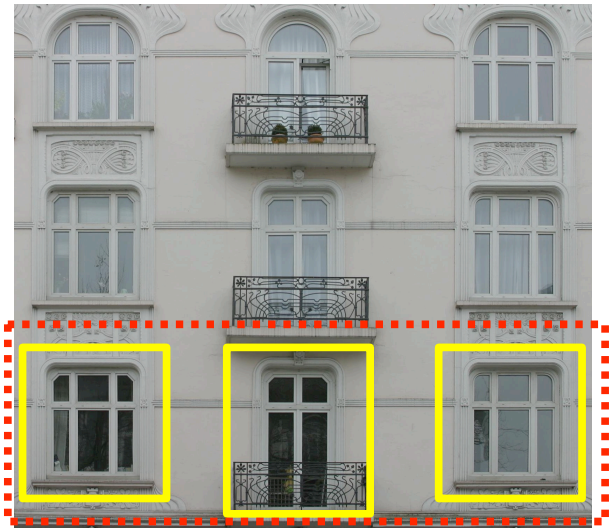
negative example #1

Extending Version Space Learning (2)

Incremental refinement of conceptual description of "window array"

positive example #1

positive example #2



negative example #1