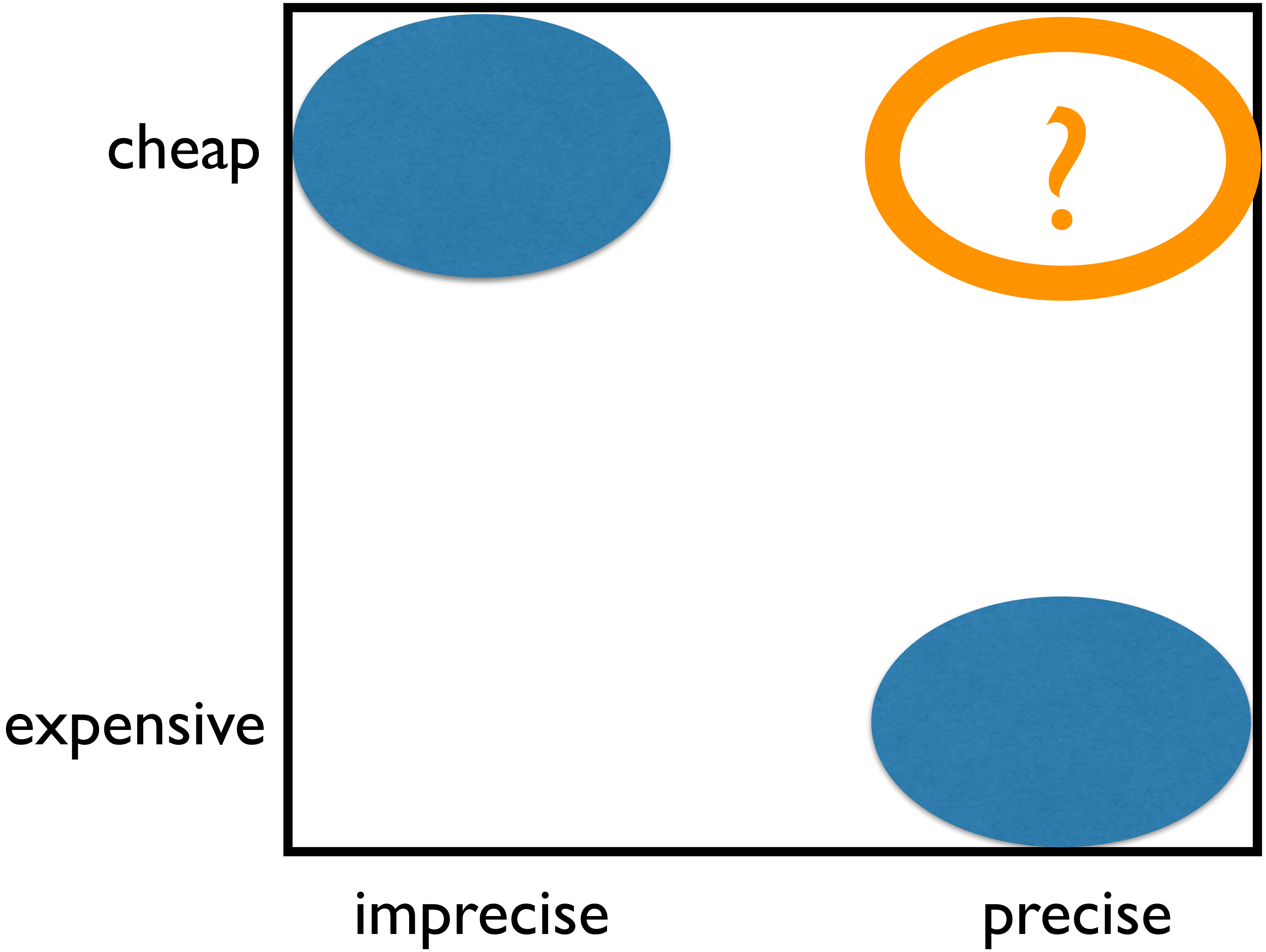
머신러닝을 이용한 선별적 정적 분석

오학주
서울대학교 프로그래밍 연구실

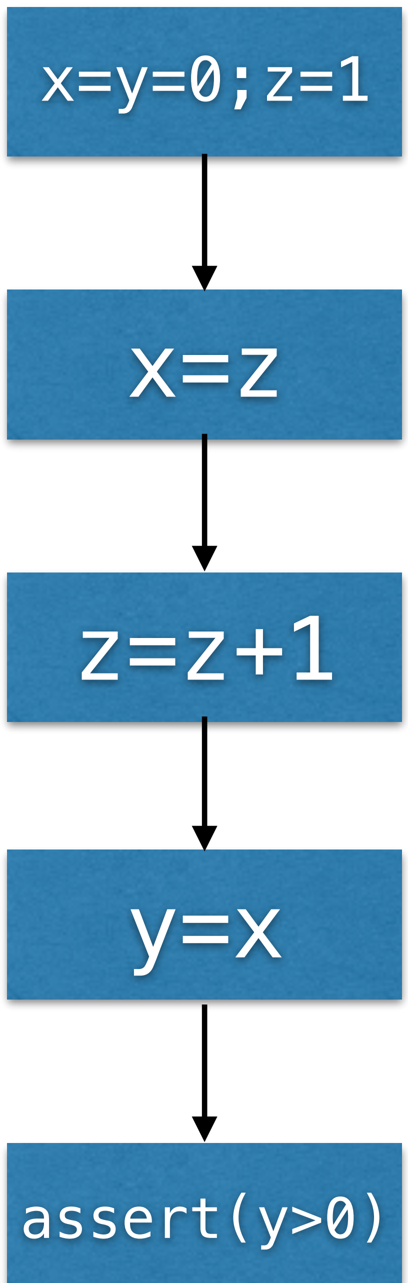
(with 양홍석)

Jan. 29, 2014 @ROSAEC Workshop

Challenge in Static Analysis



ex) Flow-Sensitivity



x	[0,0]
y	[0,0]
z	[1,1]

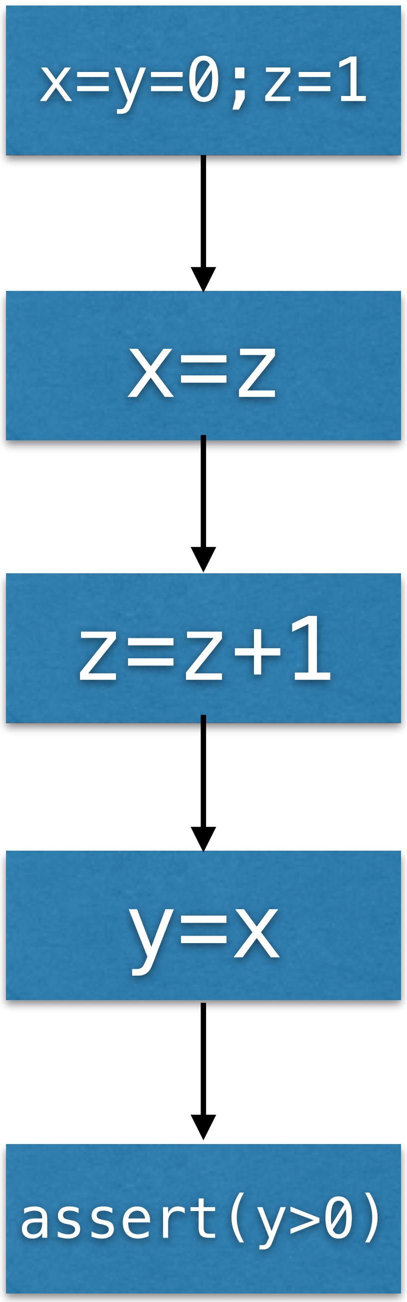
x	[1,1]
y	[0,0]
z	[1,1]

x	[1,1]
y	[0,0]
z	[2,2]

x	[1,1]
y	[1,1]
z	[2,2]

precise but costly

Flow-Insensitivity

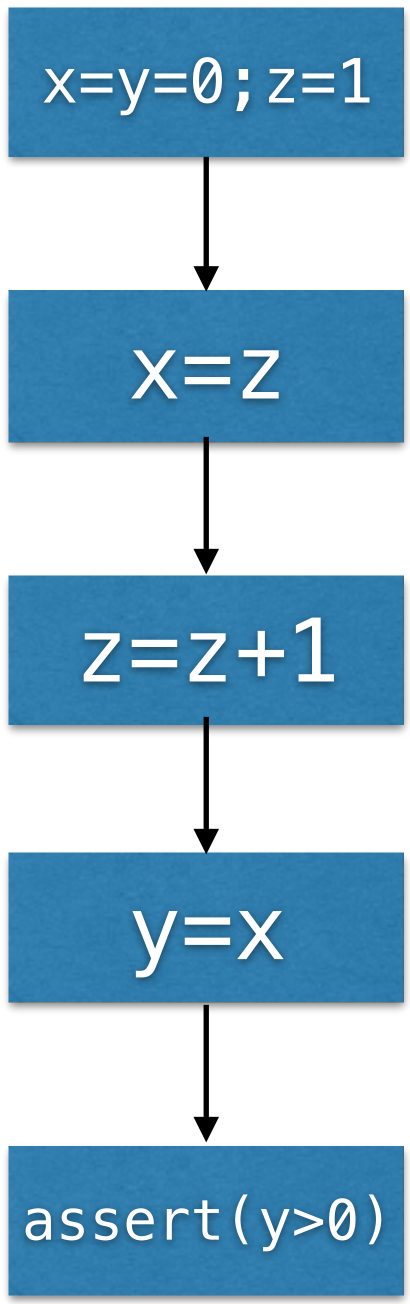


x	[0,+∞]
y	[0,+∞]
z	[1,+∞]

cheap but imprecise

FS : {x}

FI : {y,z}



x	[0,0]

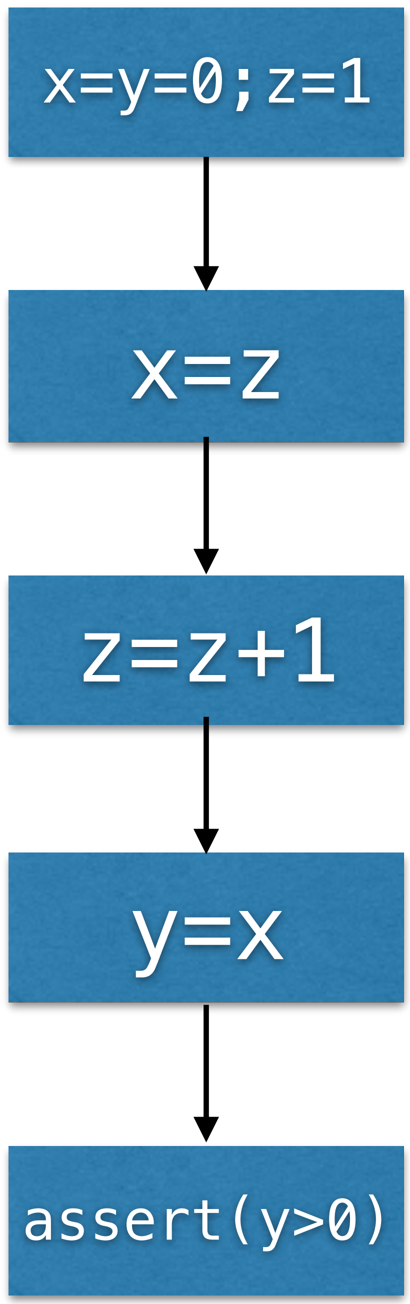
x	[1,+∞]

x	[1,+∞]

x	[1,+∞]

fail to prove

y	[0,+∞]
z	[1,+∞]



FS : {y}

y	[0,0]

y	[0,0]

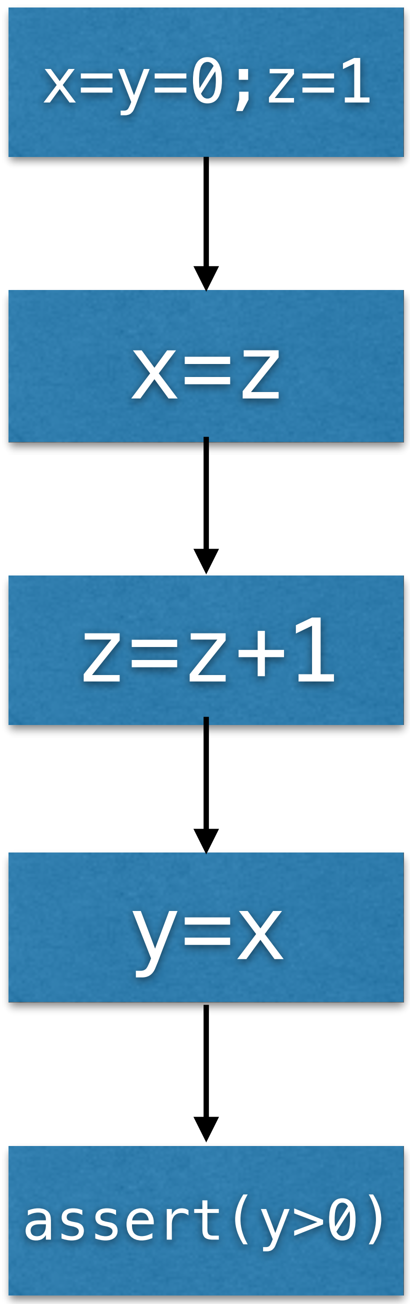
y	[0,0]

y	[0,+∞]

fail to prove

FI : {x,z}

x	[0,+∞]
z	[1,+∞]



FS : {z}

z	[1,1]

z	[1,1]

z	[2,2]

z	[2,2]

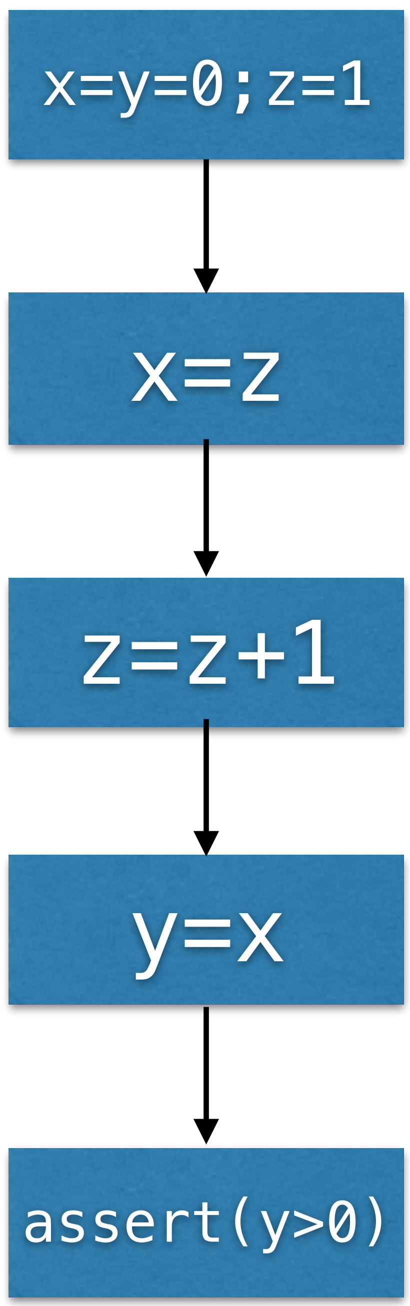
fail to prove

FI : {x,y}

x	[0,+∞]
y	[0,+∞]

FS : {y,z}

FI : {x}



y	[0,0]
z	[1,1]

y	[0,0]
z	[1,1]

y	[0,0]
z	[2,2]

y	[0,+∞]
z	[0,0]

fail to prove

x	[0,+∞]

FS : {x,y}

FI : {z}

```
x=y=0;z=1
   ↓
x=z
   ↓
z=z+1
   ↓
y=x
   ↓
assert(y>0)
```

x	[0,0]
y	[0,0]

x	[1,+∞]
y	[0,0]

x	[1,+∞]
y	[0,0]

x	[1,+∞]
y	[1,+∞]

Success!

z	[1,+∞]

Hard Search Problem

- Intractably large space, if not infinite
  - 2^N different parameters in FS
- Most of them are too imprecise or costly
  - $P(\{x,y,z\}) = \{\cancel{\varnothing}, \cancel{\{x\}}, \cancel{\{y\}}, \cancel{\{z\}}, \{x,y\}, \cancel{\{y,z\}}, \cancel{\{x,z\}}, \{x,y,z\}\}$

Past Approaches

- Abstraction refinement
- Search algorithms for parametric analyses
- Estimating the impacts of the precision
- …

Our Learning-based Approach

- Learn model M from dataset

$$\boxed{P_1, P_2, \ldots, P_m} \quad \Rightarrow \quad M : \text{pgm} \rightarrow 2^{\text{Var}}$$

Dataset

- For new program P, run static analysis with M(P)

Effectiveness
(flow-sensitivity)

trials	FI		FS			selective FS			
	prove	time	prove	time	cost	prove	time	quality	cost
1	3312	75.8	4602	372.8	4.9 x	4509	100.6	92.8 %	1.3 x
2	4558	56.3	5442	1014.2	18.0 x	5029	107.8	53.3 %	1.9 x
3	5261	109.4	6006	1586.5	14.5 x	5771	192.2	68.5 %	1.8 x
4	2415	35.1	2803	430.8	12.3 x	2725	65.2	79.9 %	1.9 x
5	4050	85.8	5313	1588.2	18.5 x	5124	134.7	85.0 %	1.6 x
total	19596	362.4	24166	4992.5	**13.8 x**	23158	600.5	**77.9 %**	**1.7 x**

$$M : \text{pgm} \rightarrow 2^{\text{Var}}$$

- 30 pgms randomly divided into 18 training and 12 test sets
- 10% selection of variables

Effectiveness
(flow-sensitivity + context-sensitivity)

FICI		FSCS			selective FSCS			
prove	time	prove	time	cost	prove	time	quality	cost
3312	75.8	5947	5960.0	78.6 x	5810	151.9	94.8 %	2.0 x
4558	56.3	6989	5150.8	91.5 x	6496	257.8	79.7 %	4.6 x
5260	109.4	7849	9511.1	86.9 x	7180	304.6	74.2 %	2.8 x
2415	35.1	3079	5013.7	142.8 x	2843	112.2	64.5 %	3.2 x
4050	85.8	5761	3528.6	41.1 x	5412	254.1	79.6 %	3.0 x
19595	362.4	29625	29164.2	**80.5 x**	27741	1080.6	**81.2 %**	**3.0 x**

$$M : \text{pgm} \rightarrow 2^{\text{Var}} \times 2^{\text{Procedure}}$$

- FSCS: context-sensitive for all procedures of size less than 80
- 10% selection of variables and procedures

Learning M from Dataset

1. Parameterize Static Analysis

number of
proved assertions

$$F(p, a) = n$$

abstraction
(e.g., a set of variables)

2. Choose Features

- Predicates (over variables)

$$f = \{f_1, f_2, \ldots, f_5\} \qquad (f_i : \mathrm{Var} \rightarrow \{0,1\})$$

- Represent each variable as a feature vector

$$f(x) = \langle f_1(x), f_2(x), f_3(x), f_4(x), f_5(x) \rangle$$

$$x : \langle 1,0,1,0,0 \rangle$$
$$y : \langle 1,0,1,0,1 \rangle$$
$$z : \langle 0,0,1,1,0 \rangle$$

3. Fix a Representation of *M*

$$M_{\mathrm{w}} : \mathrm{pgm} \rightarrow 2^{\mathrm{Var}}$$

- Parameterize *M* by a weight vector **w** $\in R^n$

$$\mathbf{w} = \langle 0.1, -0.5, 0.9, 0.3, -0.1 \rangle$$

- Compute *scores* of variables

$$x : \langle 1,0,1,0,0 \rangle \cdot \langle 0.1, -0.5, 0.9, 0.3, -0.1 \rangle = 1.0$$
$$y : \langle 1,0,1,0,1 \rangle \cdot \langle 0.1, -0.5, 0.9, 0.3, -0.1 \rangle = 0.9$$
$$z : \langle 0,0,1,1,0 \rangle \cdot \langle 0.1, -0.5, 0.9, 0.3, -0.1 \rangle = 1.2$$

- Select top k variables: {x,z} (k = 2)

4. Learn the best w from Dataset

$$\boxed{P_1, P_2, \ldots, P_m} \underset{\text{learn}}{\Longrightarrow} \mathbf{w}^*$$

Dataset

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathbb{R}^n}{\operatorname{argmax}} \sum_{P_i \in Dataset} F(P_i, M_{\mathbf{w}}(P_i))$$

Random Sampling

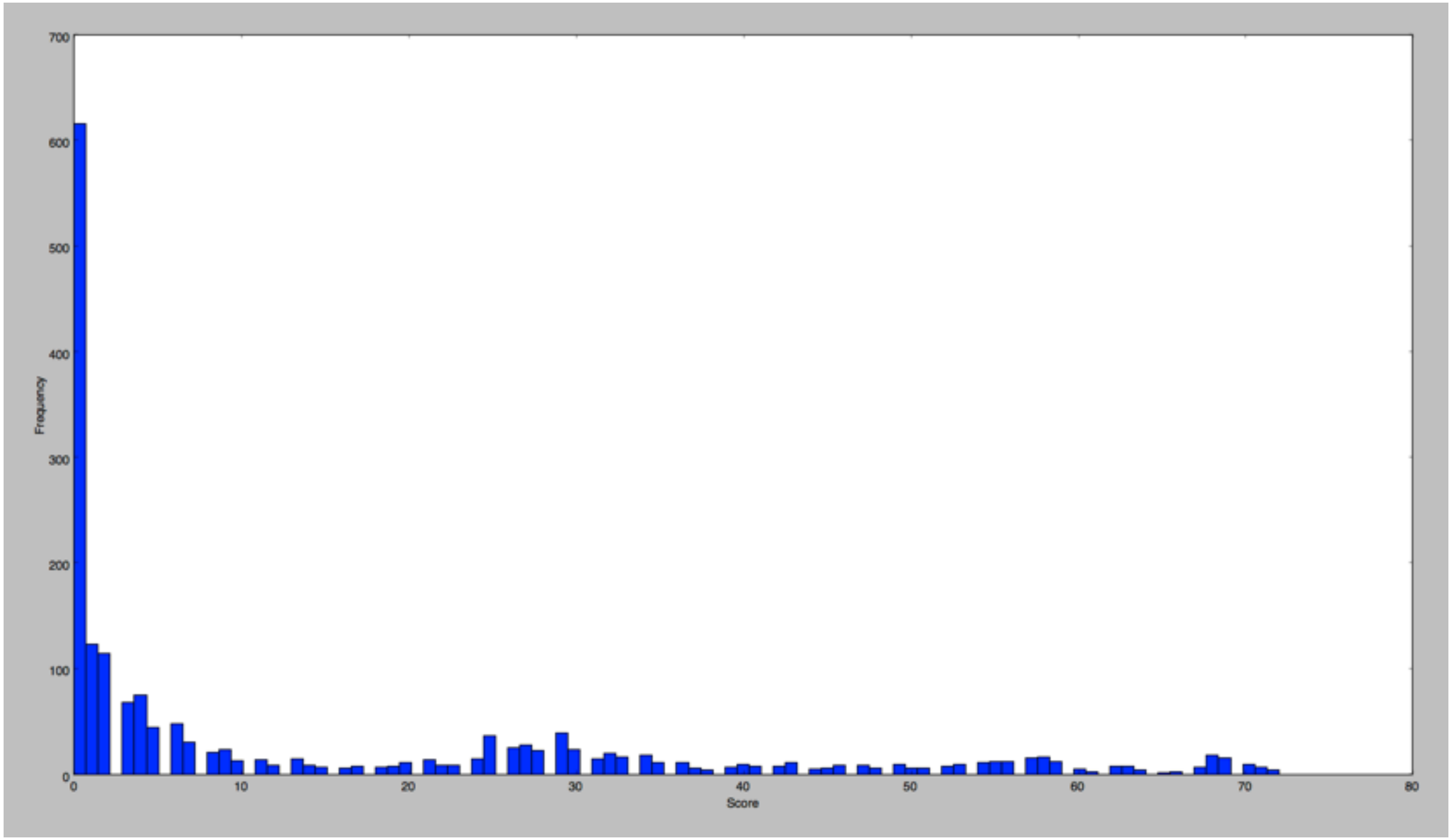
repeat N times

pick $\mathbf{w} \in \mathrm{R}^n$ randomly

evaluate $$\sum_{P_i \in Dataset} F(P_i, M_{\mathbf{w}}(P_i))$$

return best $\mathbf{w}$ found

Random Sampling



Model-based Search
(Bayesian Optimization)

repeat N times

Fit a probabilistic model to (w_k, y_k)

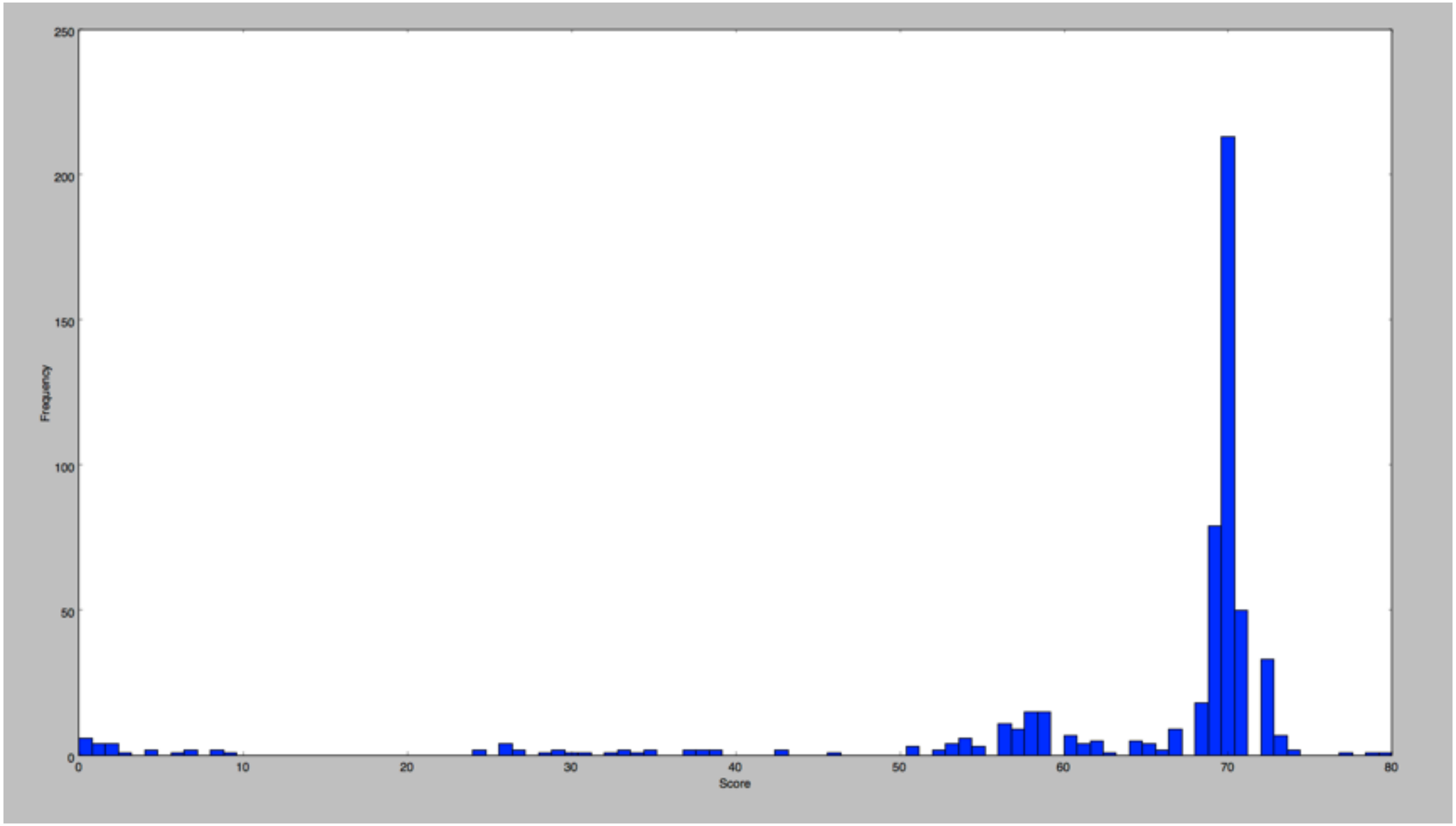
Select a promising w

$$y_i = \sum_{P_i \in Dataset} F(P_i, M_{\mathbf{w}}(P_i))$$

return best w found

- Probabilistic model: Gaussian processes
- Selection strategy: Expected improvement

Model-based Search



Summary

Our Learning-based Approach

- First learning-based approach
- Effective: 80% precision with 3x cost
- Generally applicable to any static analysis
- Fully automatic learning process

감사합니다