Processing Forecasting Queries

Songyun Duan, Shivnath Babu

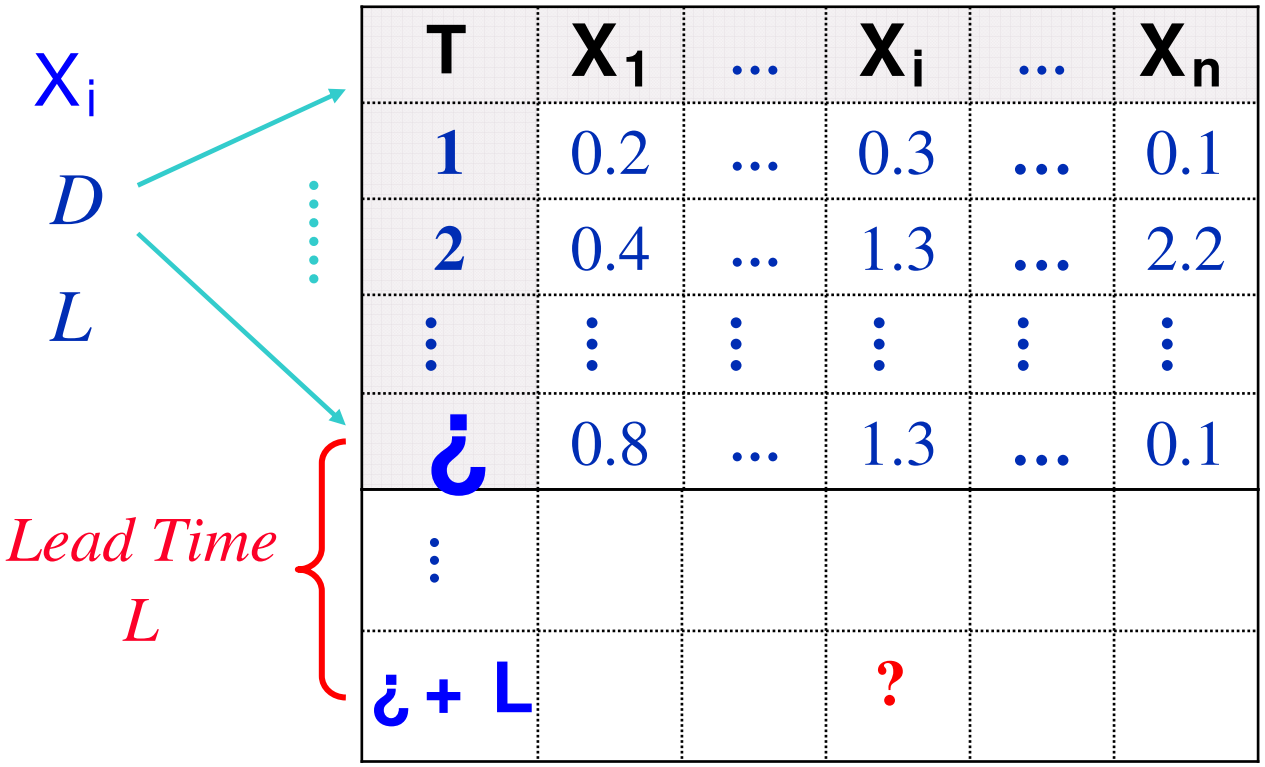
Duke University

Motivation

- Real-time forecasting of future events based on historical data is useful in many domains
  - Proactive system management
    - If a performance problem is forecast, take corrective actions in advance to avoid it
  - Adaptive query processing
  - Inventory planning
  - Environmental monitoring
  - And many others
- Need a framework to process forecasting queries automatically and efficiently

Forecasting Queries

- Select X_i
  From D
  Forecast L



- $D(T; X_1; X_2; : : : ; X_n)$: historical time-series data up to timestamp ¿
- Denoted as $Forecast(D, X_i; L)$

An Example Forecasting Query

Day	A	B	C
9	35	25	17
10	35	46	68
11	13	46	16
12	13	46	68
13	35	46	68
14	36	46	16
15	35	25	16
16	13	47	68
17	12	25	16
18			?

Table: Usage

Select *C*

From Usage

Forecast 1 day

Lead time

Example Query Processing
-- a Naïve Approach

Day	A	B	C	C_1
9	35	25	17	17
10	35	46	68	68
11	13	46	16	16
12	13	46	68	68
13	35	46	68	68
14	36	46	16	16
15	35	25	16	16
16	13	47	68	68
17	12	25	16	16
18			?	?

Class attribute

26.6

$C_1 = 0.47^* A + 1.18^* B - 0.53^* C$

Example Query Processing

Day	A	B	C
9	35	25	17
10	35	46	68
11	13	46	16
12	13	46	68
13	35	46	68
14	36	46	16
15	35	25	16
16	13	47	68
17	12	25	16
18			?

Day	A_{i-2}	B_{i-1}	C_1
9	35	25	68
10	35	46	16
11	13	46	68
12	13	46	68
13	35	46	16
14	36	46	16
15	35	25	68
16	13	47	16
17	12	25	? → 68

Previous prediction=26.6

$C_1 = 1.24 * A_{i-2} + 0.3 * B_{i-1}$

Bayesian network

Challenges

To process real-time forecasting queries,

- Challenge 1: generate a good processing strategy automatically and efficiently
  - Apply appropriate transformations to the data
    - E.g., shift, discretization, normalization, aggregation
  - Pick the right type of statistical model
    - E.g., multivariate linear regression (MLR), classification and regression tree (CART), Bayesian network (BN)
- Challenge 2: for continuous forecasting over streaming data, adapt processing strategies when necessary

Outline

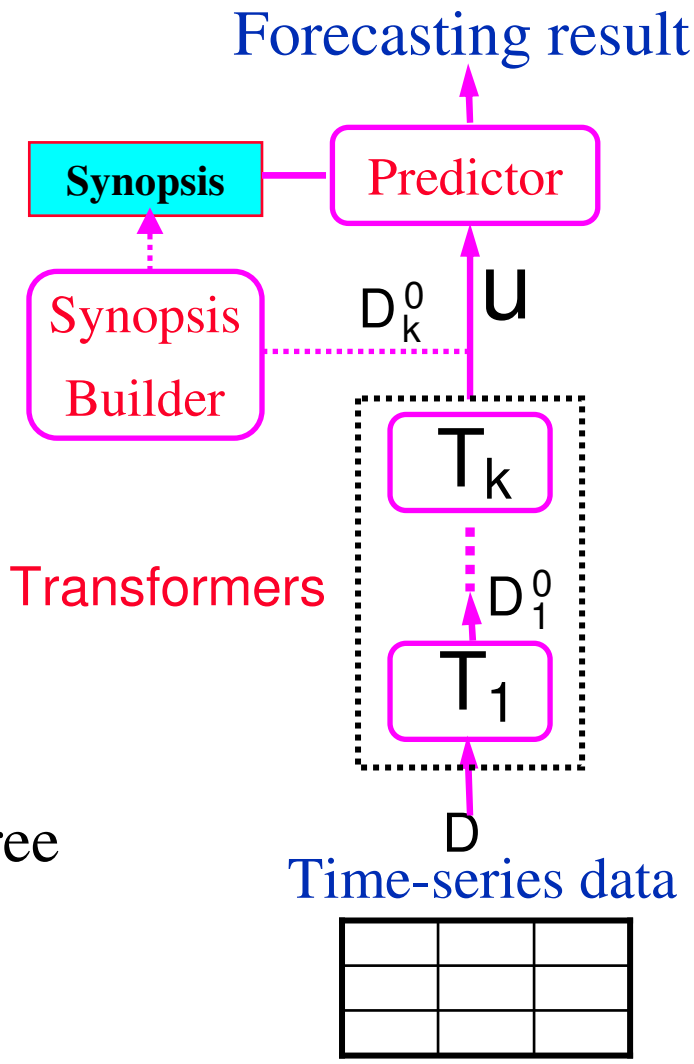
- Space of execution plans
- Plan Search Algorithm
- Processing continuous forecasting query
- Experimental evaluation
- Related work
- Summary

Execution Plan

Query: Forecast(D, X_i; L)

- Logical operators
  - Transformer: D) D^0
    - E.g., Shift (X, δ)
  - Synopsis builder:

    B(D;Z)) Syn(f Y_1;¢¢¢;Y_ng ! Z)
  - Predictor: P(Syn;u)) u:Z
- Synopsis Syn(f Y_1;¢¢¢;Y_ng ! Z)
  - E.g., linear regression, regression tree



Sample Execution Plan

Select	*C*
From	Usage
Forecast	1 day

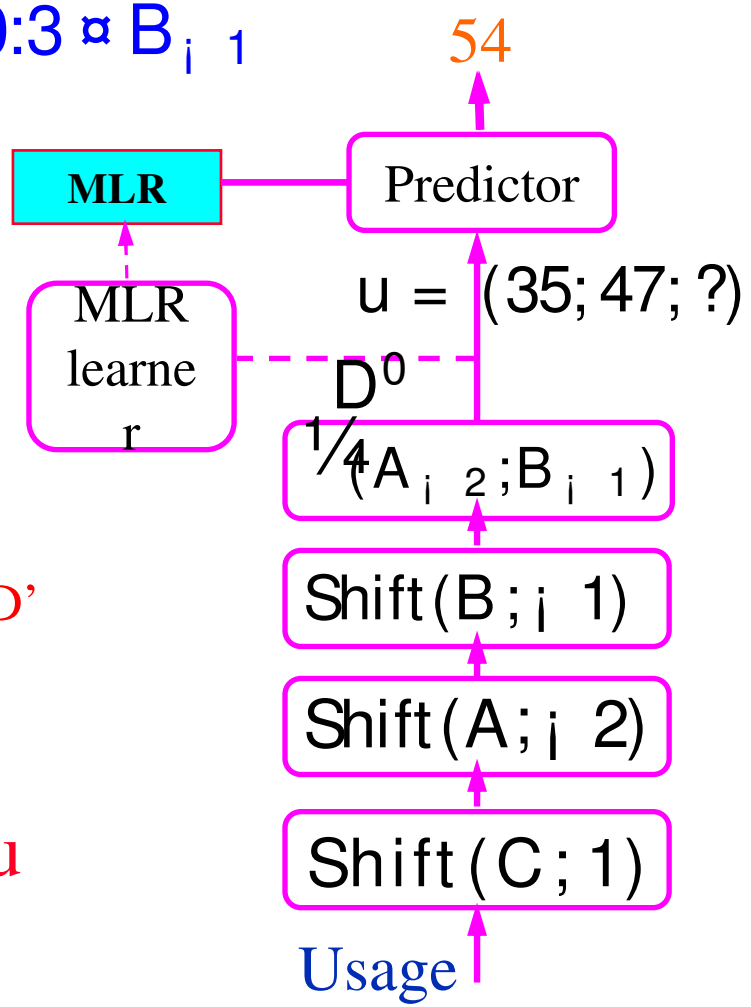
Synopsis = multivariate linear regression (MLR)

$C_1 = 1.24 * A_{i-2} + 0.3 * B_{i-1}$

Day				A_{i-2}	B_{i-1}	C_1
			… … …			
13				…	…	16
14				…	46	16
15				35	46	68
16				36	25	16
17				35	47	?
18			?	12	25	?

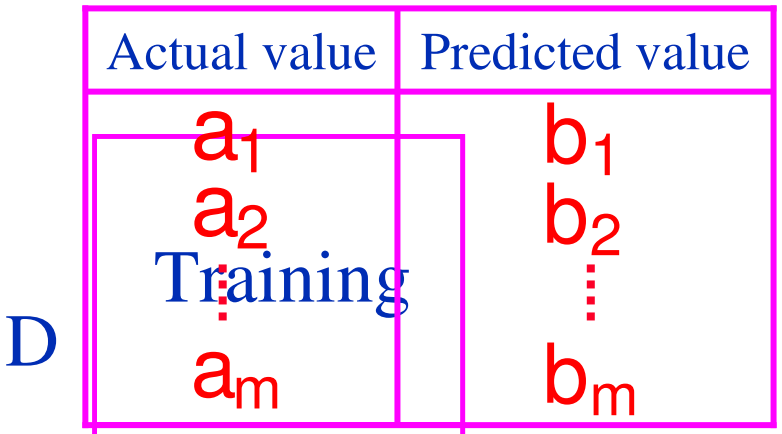
D’

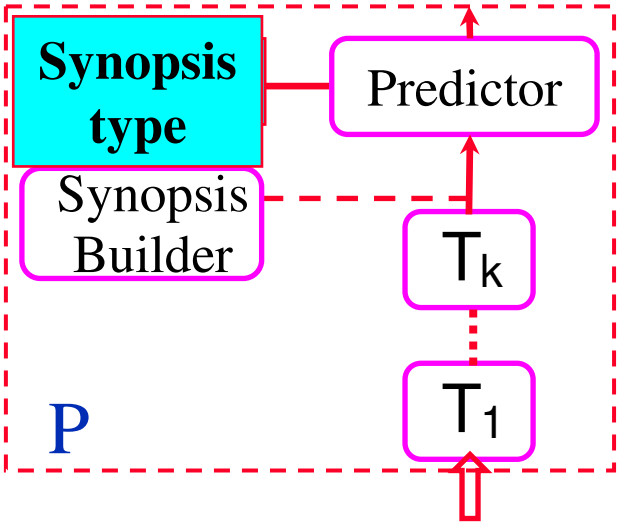
u



Estimating Accuracy of a Plan

- Accuracy
  - How close are forecasting results to "real values"
- Given a dataset D and a plan P





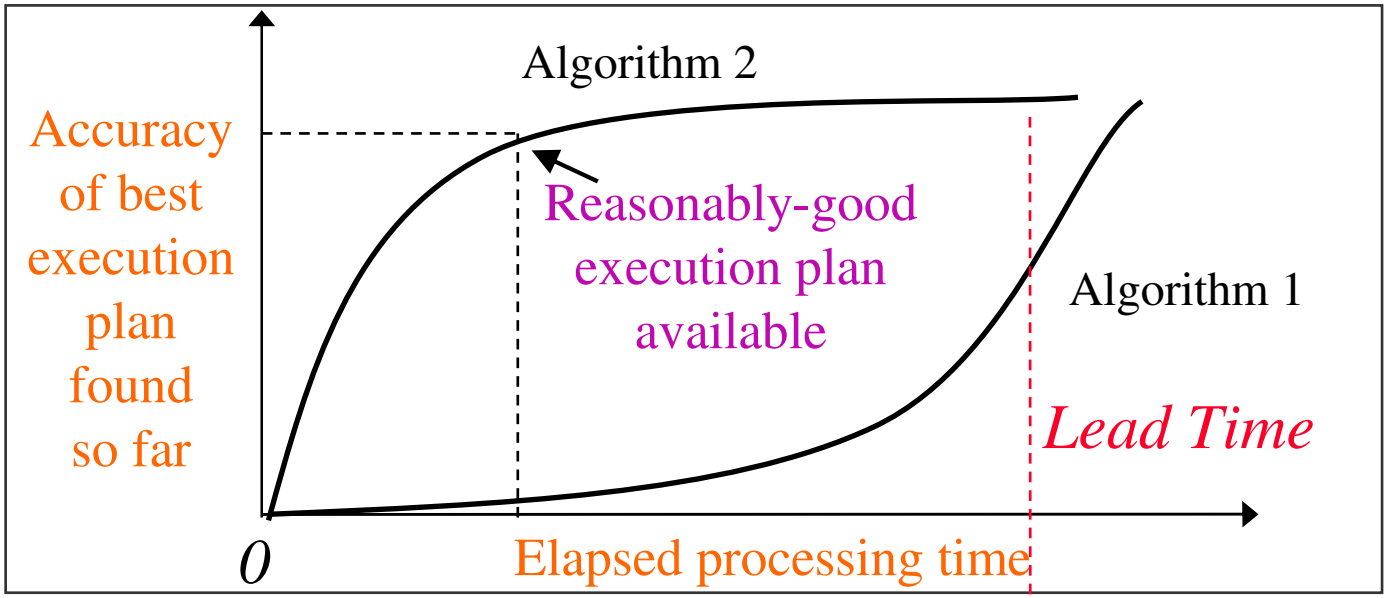
Example accuracy metric:

$$RMSE = \sqrt{\frac{\sum_i (a_i - b_i)^2}{m}}$$

K-fold cross validation to get unbiased estimation

Find a Good Plan Quickly

- Optimization challenge: minimize the number of plans executed before finding a plan with high accuracy
  - Efficient plan search to balance accuracy Vs. running time



- Simplified plan space to describe our algorithms
  - Two types of transformers: Shift and Project
  - One synopsis: Bayesian Network (BN)

Fa's Plan Search (FPS) Algorithm

Query: Forecast(Usage, C, 1)

Dataset (n = 3)

Shift(X_i, δ) (-Δ<=δ<0) (Δ=2)

A	B	C	A_{i-1}	B_{i-1}	C_{i-1}	A_{i-2}	B_{i-2}	C_{i-2}
1	4	7	1	4	7	1	4	7
2	5	8	2	5	8	2	5	8
3	6	9	3	6	9	3	6	9
…	…	…	…	…	…	…	…	…

C_1
8
9
…
?

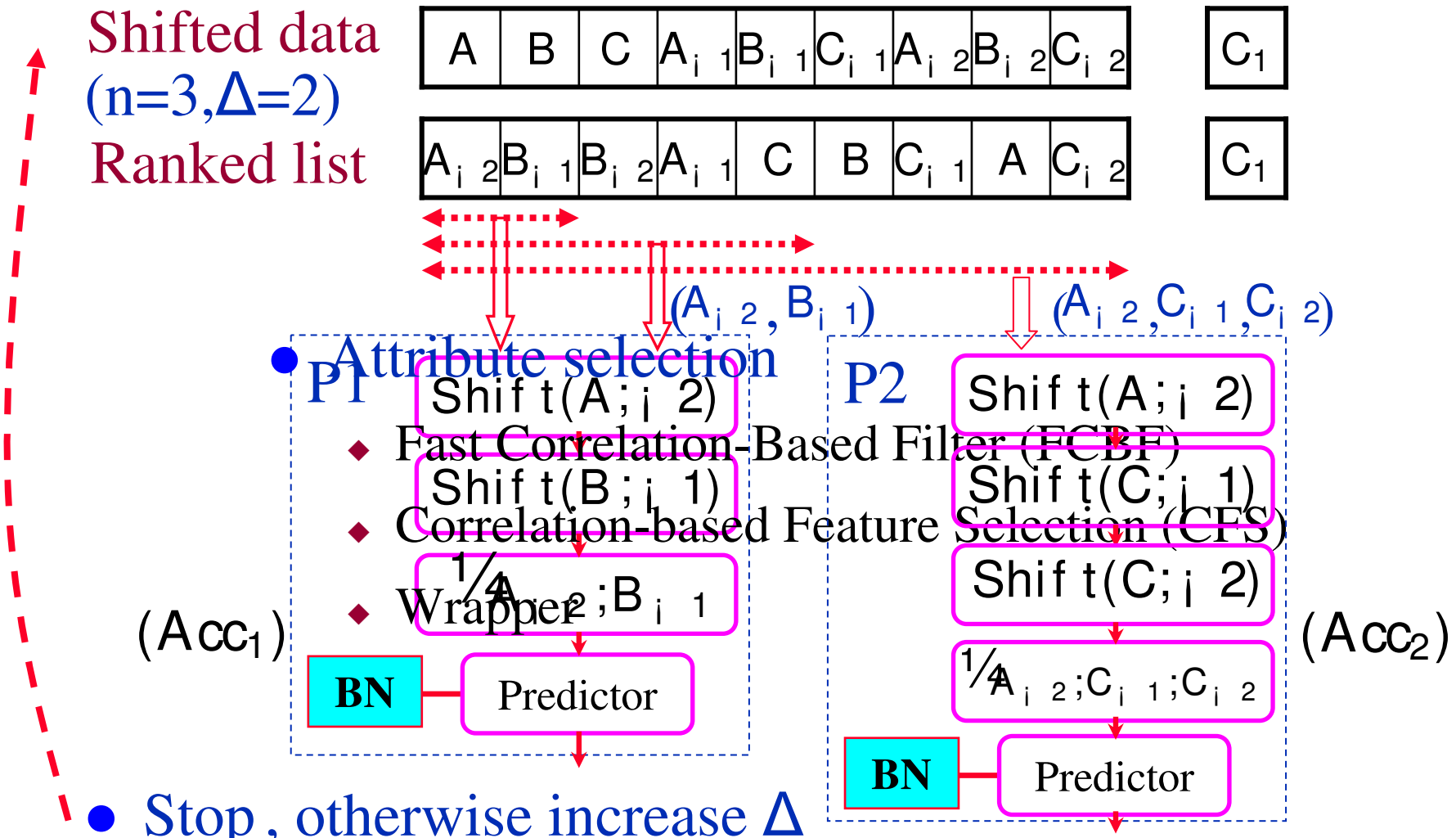
Class attribute

Ranked list

A_{i-2}	B_{i-1}	B_{i-2}	A_{i-1}	C_{i-1}	B_{i-1}	C_{i-1}	A_{i-2}	C_{i-2}	C

- Learn a synopsis and generate the plan?
- Imagine: Attribute Ranking
  - Extended data has 9000+ attributes
  - Takes too much time to get a plan
- Attribute-Ranking
  - Linear-correlation-based
  - Entropy-based, e.g., information gain

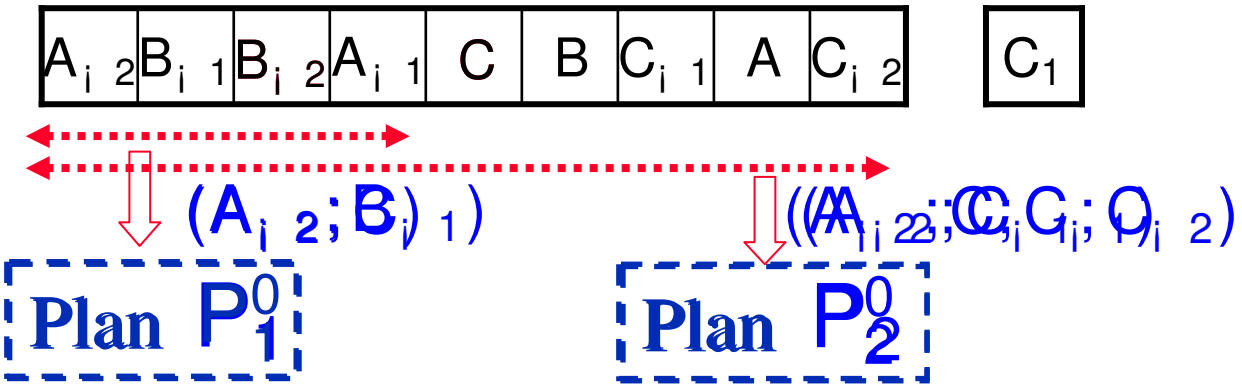
Fa's Plan Search (FPS) Algorithm



Adaptive Fa's Plan Search (FPS-A)

Continuous query: Forecast(S[W], X_i; L)

- The ranked list and plans for S[W]



Outline

- Space of execution plans
- Plan Search Algorithm
- Processing continuous forecasting query
- Experimental evaluation
- Related work
- Summary

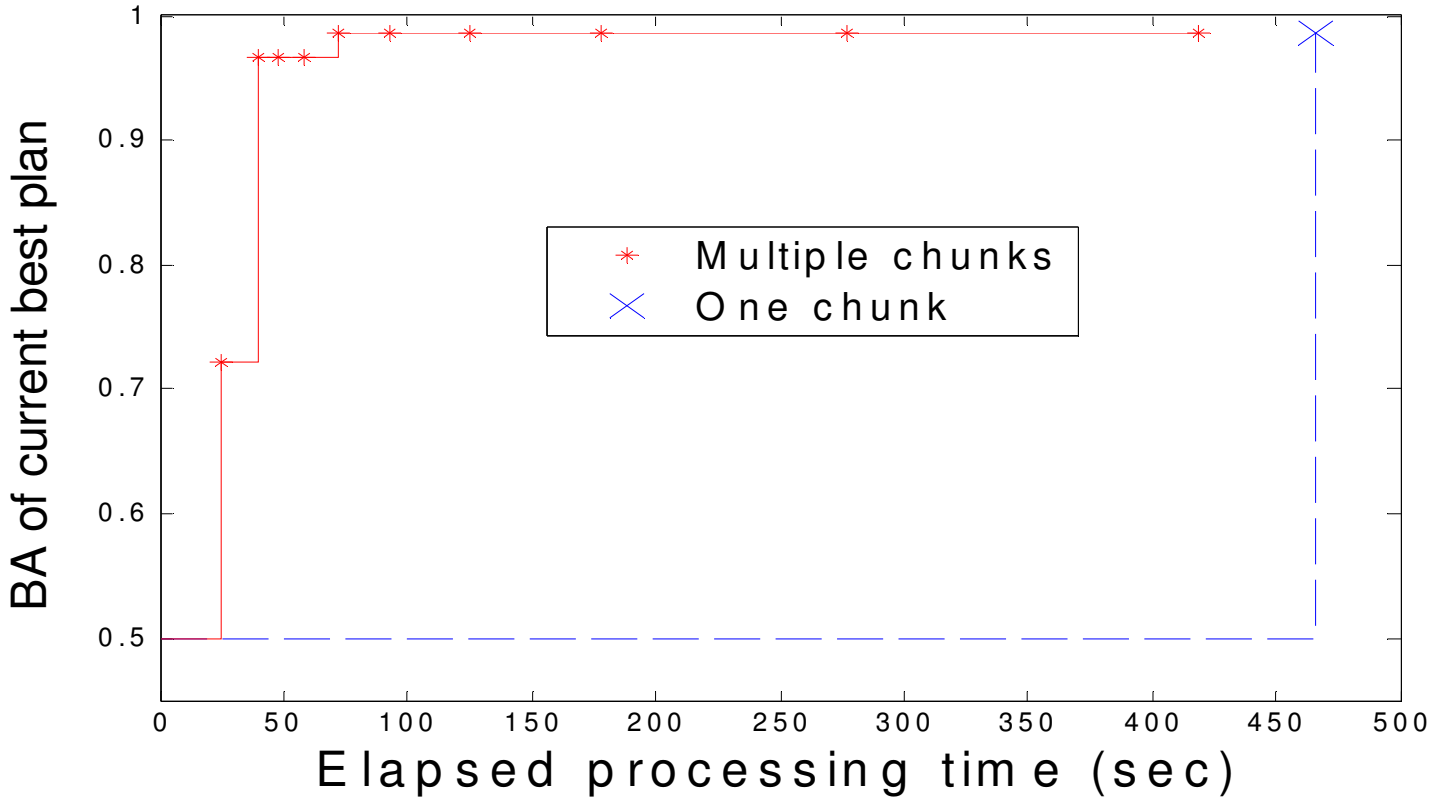
Experimental Setting

- Target domain: system and database monitoring
- Datasets (#attributes 3~250, #instances 700~15000)
  - Aging dataset from a departmental cluster
    - Aging behavior: progressive degradation in performance
  - A real dataset parsed from logs of 98' World Cup web-site
    - Periodic segments – characteristic of most popular web-sites
  - 5 testbed datasets
    - Our testbed runs OLTP applications using MySQL
    - Simulated periodic workloads, aging behavior, and multiple resource contentions
  - 2 synthetic datasets: simulated complicated patterns to study the robustness of our algorithms

Multiple Chunks Vs. One Chunk

- Accuracy metric = balanced accuracy

$$BA = 1 - 0.5 * \left(\frac{\#\,false\_positives}{\#\,negatives} + \frac{\#\,false\_negatives}{\#\,positives} \right)$$



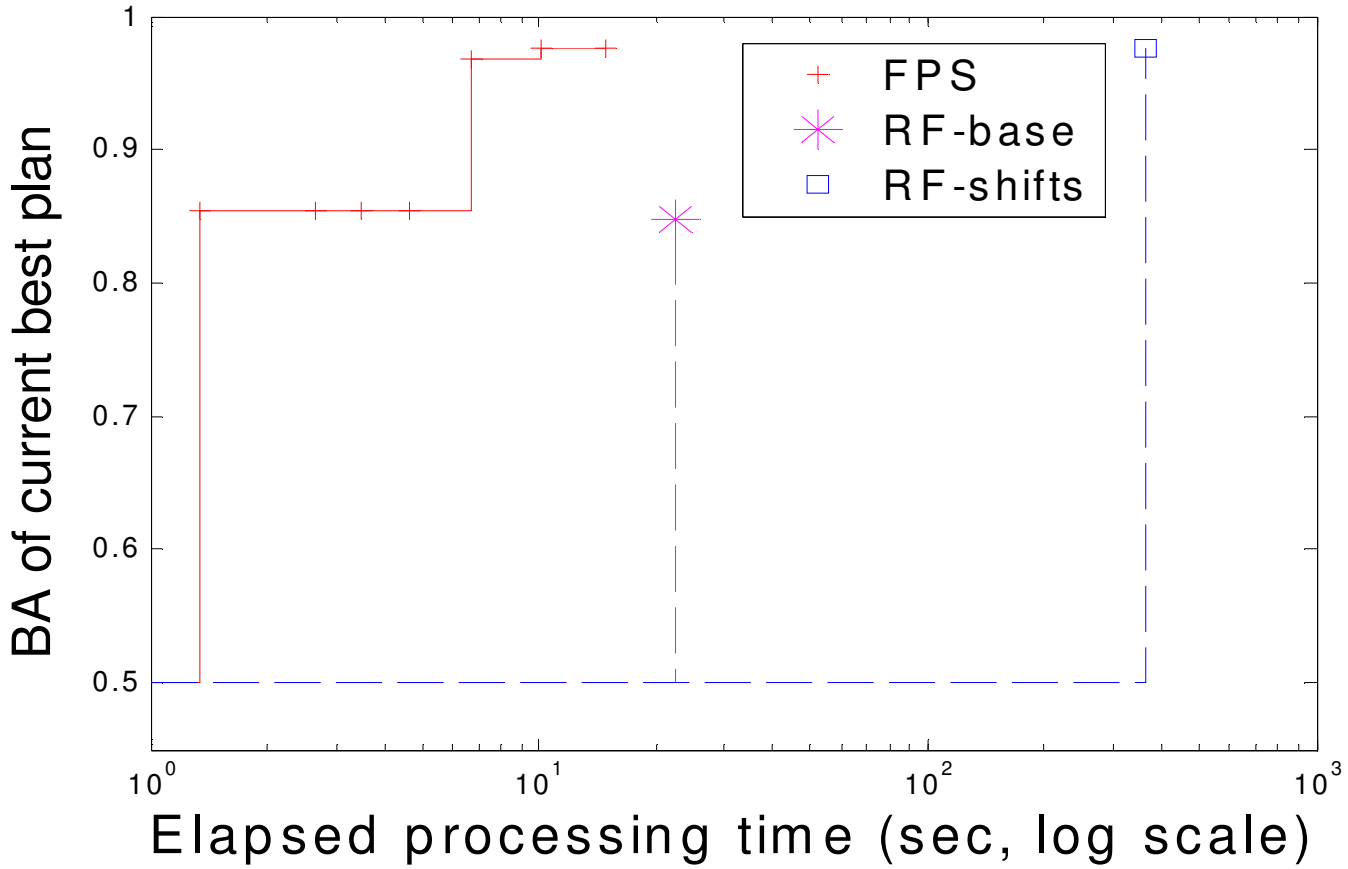
Testbed dataset, Lead time = 25, n=50+,Δ= 30

Synopsis Comparison

	FPS(BN)		FPS(CART)		FPS(MLR)		FPS(SVM)		FPS(RF)	
Dataset	BA	Time	BA	Time	BA	Time	BA	Time	BA	Time
Aging-real	.71	62	.71	135	.64	36	0.51	1948		
FIFA-real	.87	29	.85	37	.84	201				
Periodic-small-tb	.84	45	.85	249	.80	130			.86	22339
Multi-small-tb	.91	53	.91	50	.85	19			.91	933
Aging-variant-tb	.82	14	.81	109	.80	24	.86	482	.85	3200

- FPS using BN or CART can achieve accuracy comparable to more sophisticated synopses
  - Lesson: More important to find right transformations
- FPS using BN or CART has lower running time
- Thus: we use BN as the default synopsis

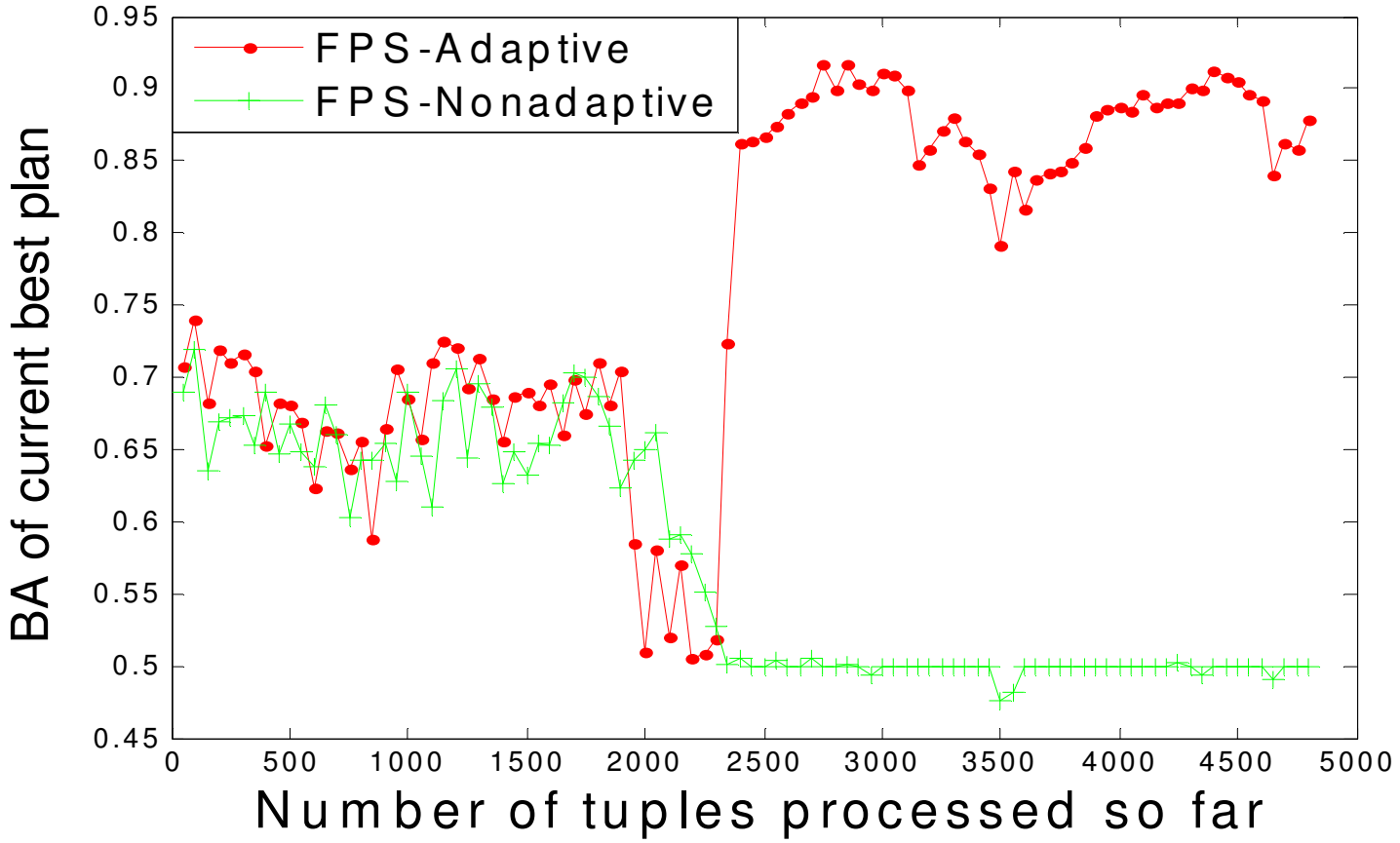
FPS Vs. State-of-the-Art Synopsis



Synthetic dataset, Lead time = 25, n = 3, Δ= 90

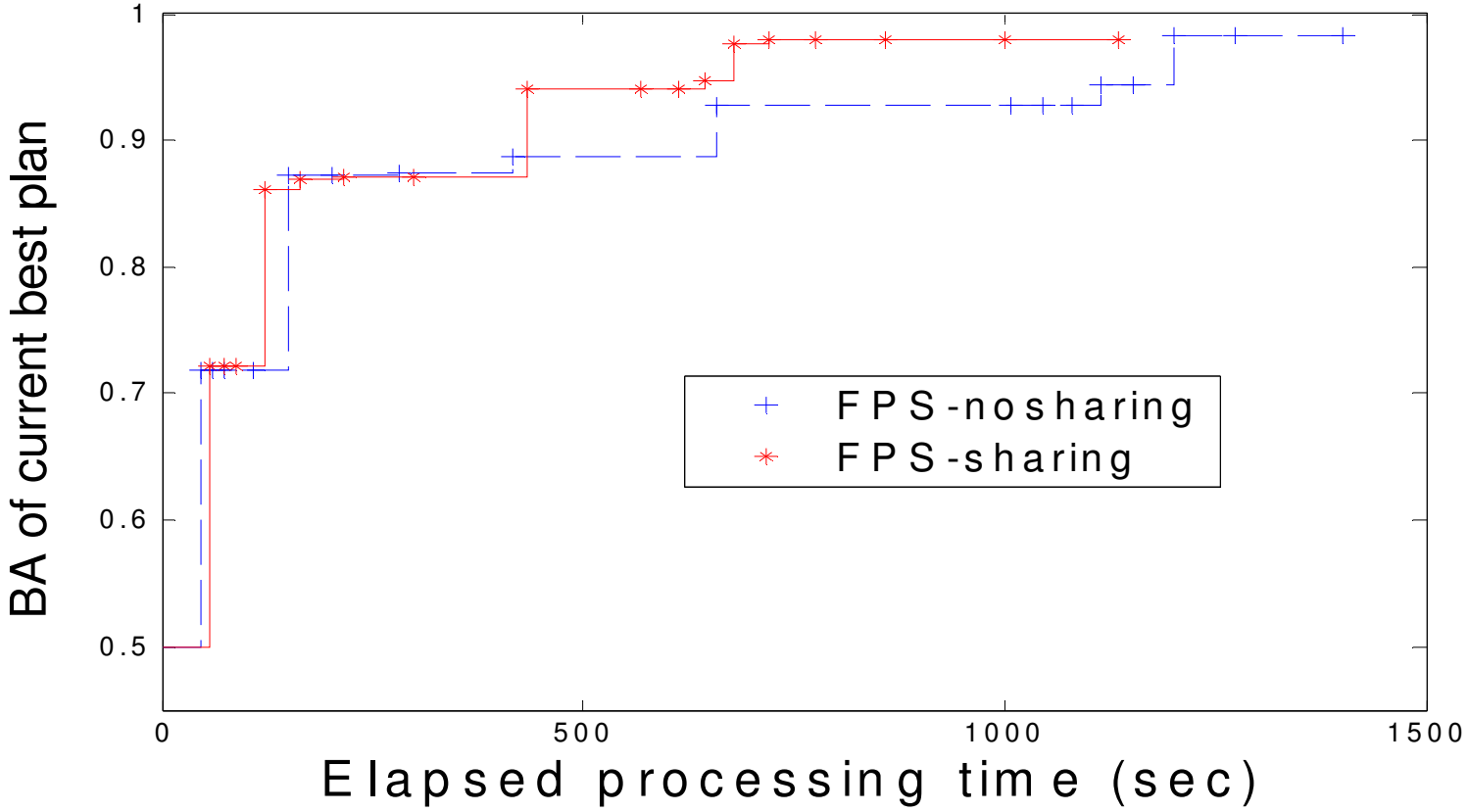
FPS-adaptive Vs. FPS-nonadaptive

- Runtime overhead < 2%
- Adaptability and Convergence (Lead time = 25, Δ= 90)



Sharing Computation in FPS and FPS-A

- FPS and FPS-A aggressively share computation as multiple plans are explored during plan search



Lead time = 25, n = 50+, Δ= 90

Related Work

- Time-series forecasting, performance problem forecasting, and self-tuning
  - Difference: our automated framework balances accuracy against running time
- Machine learning techniques on data transformation and modeling
  - Can be incorporated as operators in our framework
- Integration of synopses and data mining algorithms with DBMS and DSMS
  - Used for processing conventional SQL/XML queries
  - Difference: our framework automatically chooses good combinations of transformations and synopses

Summary

- Defined declarative one-time and continuous forecasting queries
- Proposed an automatic plan search algorithm for processing one-time forecasting queries
- Proposed an adaptive algorithm for continuous forecasting over streaming data
- Extensive experimental evaluation of both algorithms

Thanks!

Execution Plans for Example Query

Query: Forecast (Usage, C, 1)