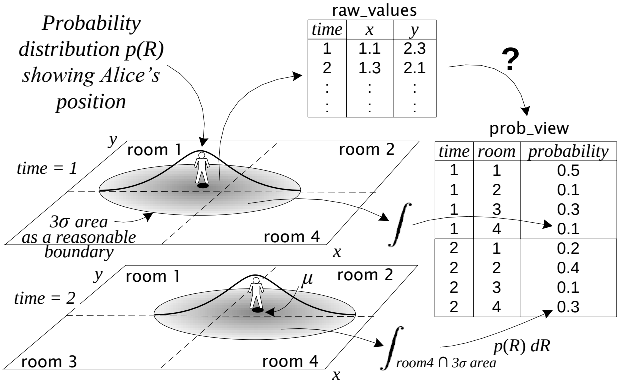
Creating Probabilistic Databases from Imprecise Time-Series Data

Saket Sathe, Hoyoung Jeung, Karl Aberer

EPFL, Switzerland

13th April, 2011

Outline



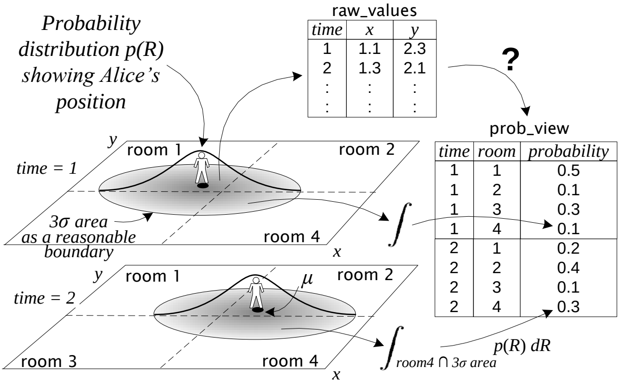
raw_values

time	x	y
1	1.1	2.3
2	1.3	2.1
⋮	⋮	⋮

prob_view

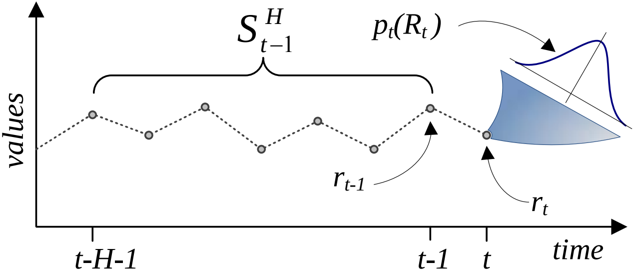
time	room	probability
1	1	0.5
1	2	0.1
1	3	0.3
1	4	0.1
2	1	0.2
2	2	0.4
2	3	0.1
2	4	0.3

Outline



- Dynamic Density Metrics
- Measure of Quality
- Efficiently creating probabilistic views
- Approximating Gaussian distributions using σ–cache
- Parameter setting under provable guarantees
- Experiments

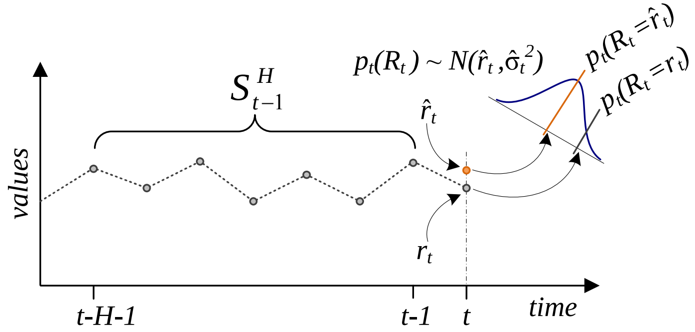
Problem Setting



Dynamic Density Metric

Given S_{t-1}^{H}, the dynamic density metric infers time-dependent probability distributions $p_t(R_t)$ at time t, where R_t is a random variable associated with r_t.

GARCH Metric



- $\hat{r}_t$ is modeled using an ARMA model
- $\hat{\sigma}_t^2$ is modeled using a GARCH model
- Thus $p_t(R_t)$ is a $\mathcal{N}(\hat{r}_t, \hat{\sigma}_t^2)$. We refer to this approach as ARMA-GARCH

Quality of Dynamic Density Metrics

	$\hat{\mathbf{r}}_{\mathbf{t}}$	$\hat{\sigma}_{\mathbf{t}}^{\mathbf{2}}$
ARMA-GARCH	ARMA	GARCH
Uniform Thresholding (UT)	ARMA	u (user-specified)
Variable Thresholding (VT)	ARMA	sample variance of S_{t-1}^{H}
Kalman-GARCH	Kalman Filter	GARCH

Quality of Dynamic Density Metrics

	$\hat{\mathbf{r}}_{\mathbf{t}}$	$\hat{\sigma}_{\mathbf{t}}^{\mathbf{2}}$
ARMA-GARCH	ARMA	GARCH
Uniform Thresholding (UT)	ARMA	u (user-specified)
Variable Thresholding (VT)	ARMA	sample variance of S_{t-1}^{H}
Kalman-GARCH	Kalman Filter	GARCH

Problem: The true density $\hat{p}_t(R_t)$ is not observable

Quality of Dynamic Density Metrics

	$\hat{\mathbf{r}}_\mathbf{t}$	$\hat{\sigma}_\mathbf{t}^\mathbf{2}$
ARMA-GARCH	ARMA	GARCH
Uniform Thresholding (UT)	ARMA	u (user-specified)
Variable Thresholding (VT)	ARMA	sample variance of S_{t-1}^H
Kalman-GARCH	Kalman Filter	GARCH

Indirect Method

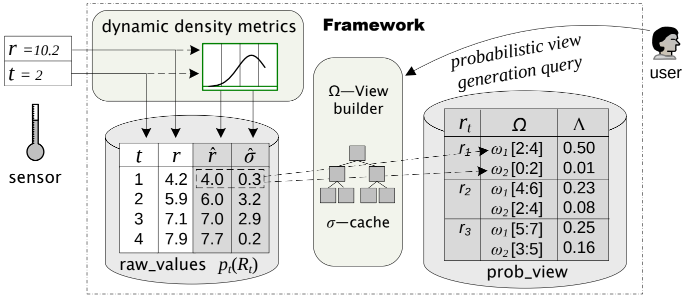
Suppose $p_1(R_1), \ldots, p_T(R_T)$ are the inferred densities and let $z_t = P(R_t \leq r_t)$ then z_t is uniformly distributed between $(0, 1)$ when $p_t(R_t) = \hat{p}_t(R_t)$ [Deibold et. al.].

$$d\{U_Z(z), Q_Z(z)\} = \sqrt{\sum_{x=0}^{1} (U_Z(x) - Q_Z(x))^2}, \tag{1}$$

where $U_Z(z)$ is the ideal uniform *cdf* between $(0, 1)$ and $Q_Z(z)$ is the observed *cdf* of z_t. We call $d\{U_Z(z), Q_Z(z)\}$ the density distance.

Probabilistic View Generation

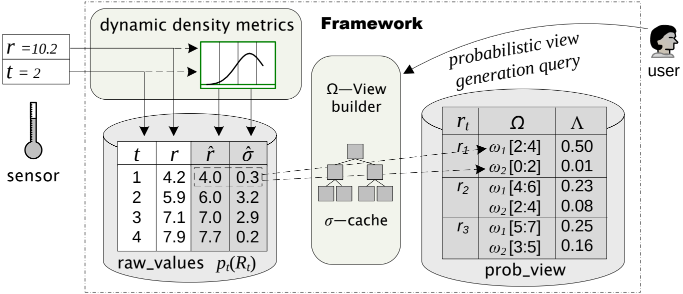
```
CREATE VIEW prob_view AS DENSITY r
OVER t OMEGA delta=2, n=2
FROM raw_values WHERE t >= 1 AND t <= 3
```



Probabilistic View Generation

```
CREATE VIEW prob_view AS DENSITY r
OVER t OMEGA delta=2, n=2
FROM raw_values WHERE t >= 1 AND t <= 3
```

Problem: Large computational cost when time interval and n are large and Δ is small (finer granularity)



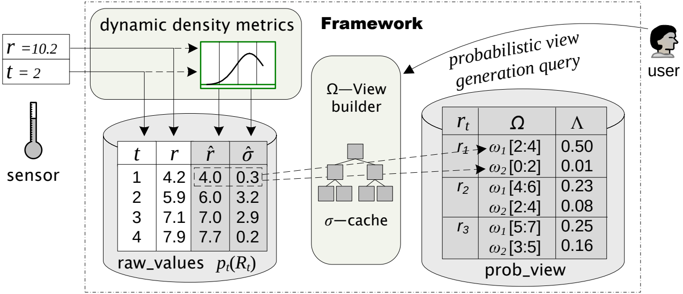
t	r	$\hat{r}$	$\hat{\sigma}$
1	4.2	4.0	0.3
2	5.9	6.0	3.2
3	7.1	7.0	2.9
4	7.9	7.7	0.2

r_t	Ω	Λ
r_1	ω_1 [2:4]	0.50
	ω_2 [0:2]	0.01
r_2	ω_1 [4:6]	0.23
	ω_2 [2:4]	0.08
r_3	ω_1 [5:7]	0.25
	ω_2 [3:5]	0.16

Probabilistic View Generation

```
CREATE VIEW prob_view AS DENSITY r
OVER t OMEGA delta=2, n=2
FROM raw_values WHERE t >= 1 AND t <= 3
```

Idea: Cache and reuse computation of probability values from earlier times



Constraint-Aware Caching

- Given: $p_t(R_t)$ and $p_{t'}(R_{t'})$ are Gaussian with $(\hat{r}_t, \hat{\sigma}_t^2)$ and $(\hat{r}_{t'}, \hat{\sigma}_{t'}^2)$
- Aim: Approximate values of $p_{t'}(R_{t'})$ by $p_t(R_t)$ when $t' > t$

Constraint-Aware Caching

- Given: $p_t(R_t)$ and $p_{t'}(R_{t'})$ are Gaussian with $(\hat{r}_t, \hat{\sigma}_t^2)$ and $(\hat{r}_{t'}, \hat{\sigma}_{t'}^2)$
- Aim: Approximate values of $p_{t'}(R_{t'})$ by $p_t(R_t)$ when $t' > t$

- Distance constraint guarantees that the maximum approximation error is upper bounded by the distance constraint when the cache is used
- Memory constraint guarantees that the cache does not use more memory than that specified by the memory constraint

Constraint-Aware Caching

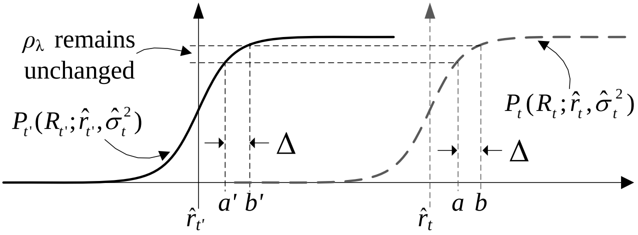
- Given: $p_t(R_t)$ and $p_{t'}(R_{t'})$ are Gaussian with $(\hat{r}_t, \hat{\sigma}_t^2)$ and $(\hat{r}_{t'}, \hat{\sigma}_{t'}^2)$
- Aim: Approximate values of $p_{t'}(R_{t'})$ by $p_t(R_t)$ when $t' > t$

- Distance constraint guarantees that the maximum approximation error is upper bounded by the distance constraint when the cache is used
- ~~Memory constraint guarantees that the cache does not use more memory than that specified by the memory constraint~~

Constraint-Aware Caching

- Given: $p_t(R_t)$ and $p_{t'}(R_{t'})$ are Gaussian with $(\hat{r}_t, \hat{\sigma}_t^2)$ and $(\hat{r}_{t'}, \hat{\sigma}_{t'}^2)$
- Aim: Approximate values of $p_{t'}(R_{t'})$ by $p_t(R_t)$ when $t' > t$

- Distance constraint guarantees that the maximum approximation error is upper bounded by the distance constraint when the cache is used
- ~~Memory constraint guarantees that the cache does not use more memory than that specified by the memory constraint~~



Guaranteeing Distance Constraint

- We use the Hellinger distance denoted $\mathcal{H}[\cdot,\cdot]$ as a distance measure. $0 \leq \mathcal{H} \leq 1$.

Theorem: Distance Constraint

Given a user-defined distance constraint $\mathcal{H}'$, we guarantee that $\mathcal{H}[p_t(R_t), p_{t'}(R_{t'})] \leq \mathcal{H}'$, if $\hat{\sigma}_{t'} \leq d_s \cdot \hat{\sigma}_t$ and $\hat{\sigma}_{t'} > \hat{\sigma}_t$ where the parameter d_s is chosen as any value satisfying,

$$d_s \leq \frac{1 + \sqrt{1 - \left(1 - \mathcal{H}'^2\right)^4}}{\left(1 - \mathcal{H}'^2\right)^2}.$$

We call d_s the ratio threshold.

Example

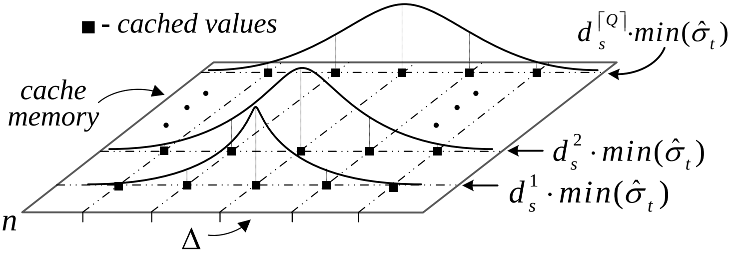
- Suppose $\mathcal{H}' = 0.2$, then $d_s \leq 1.5$
- Choose, say, $d_s = 1.4$ then if $\frac{\hat{\sigma}_{t'}}{\hat{\sigma}_t} \leq d_s$ then $\mathcal{H}\left[p_t(R_t), p_{t'}(R_{t'})\right] \leq 0.2$

Initializing the σ–cache

- Let $max(\hat{\sigma}_t)$ and $min(\hat{\sigma}_t)$ be the maximum and minimum standard deviations observed in a probabilistic view generation query
- Compute $\mathcal{Q}$, such that, $max(\hat{\sigma}_t) = d_s^{\mathcal{Q}} \cdot min(\hat{\sigma}_t)$
- $\lceil \mathcal{Q} \rceil$ gives us the maximum number of distributions that we should cache

Initializing the σ–cache

- Let $max(\hat{\sigma}_t)$ and $min(\hat{\sigma}_t)$ be the maximum and minimum standard deviations observed in a probabilistic view generation query
- Compute $\mathcal{Q}$, such that, $max(\hat{\sigma}_t) = d_s^{\mathcal{Q}} \cdot min(\hat{\sigma}_t)$
- $\lceil \mathcal{Q} \rceil$ gives us the maximum number of distributions that we should cache



- Find $d_s^q \cdot min(\hat{\sigma}_t)$ such that $d_s^q \cdot min(\hat{\sigma}_t) \leq \hat{\sigma}_{t'} < d_s^{q+1} \cdot min(\hat{\sigma}_t)$

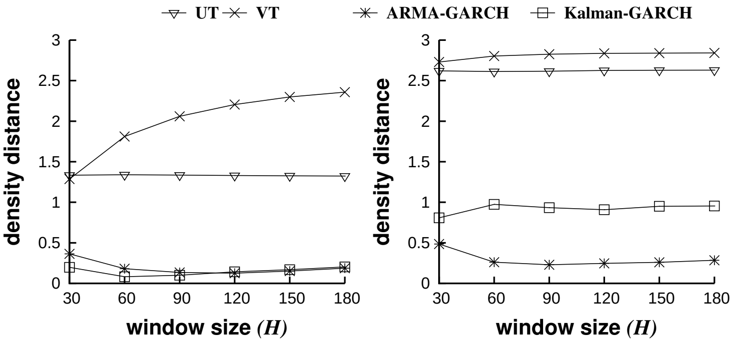
σ–cache: Features

```
CREATE VIEW prob_view AS DENSITY r
OVER t OMEGA delta=2, n=2
FROM raw_values WHERE t >= 1 AND t <= 3
```

- The rate at which the memory requirement grows is $\mathcal{O}\left(\log\left(\frac{max(\hat{\sigma}_t)}{min(\hat{\sigma}_t)}\right)\right)$
- The number of distributions cached **does not depend on**
  - number of tuples that match the `WHERE` clause
  - Δ or n

Experimental Evaluation

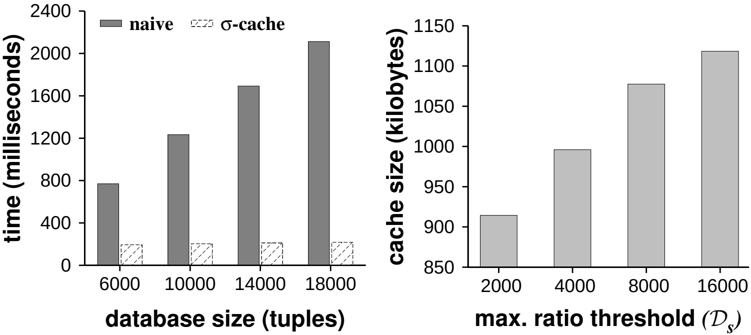
- campus-data: ambient temperature values for over sixty five hours
- car-data: more than one hour of GPS data



(a) campus-data (b) car-data

- ARMA-GARCH and Kalman-GARCH give upto 12 to 20 times lower density distances

Experimental Evaluation



(a) Efficiency (b) Scaling Characteristics

- Using $\Delta = 0.05, n = 300$ and Hellinger distance $\mathcal{H} = 0.01$
- An order of magnitude improvement in performance!

Conclusions

- Proposed time-series based models can be used for creating probabilistic databases
- Introduced the concept of density distance for measuring quality
- Proved useful and practical guarantees for using the σ-cache
- Caching and reusing distributions significantly increases the efficiency of creating probabilistic databases

Thank You.

Questions?

Saket Sathe
saket.sathe@epfl.ch

Hoyoung Jeung
hoyoung.jeung@epfl.ch

Karl Aberer
karl.aberer@epfl.ch