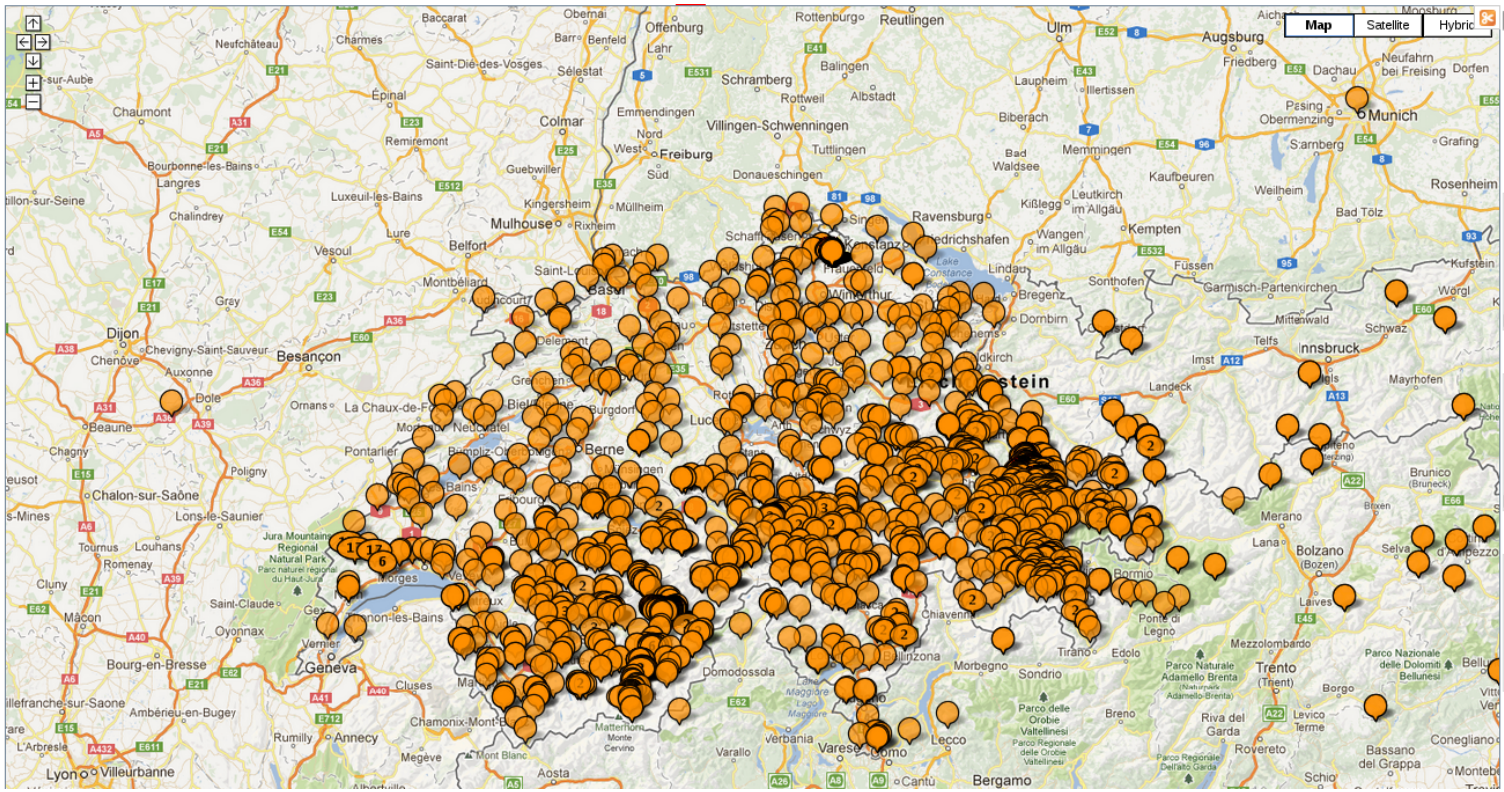
AFFINITY: Efficiently Querying Statistical Measures on Time-Series Data

Saket Sathe, Karl Aberer

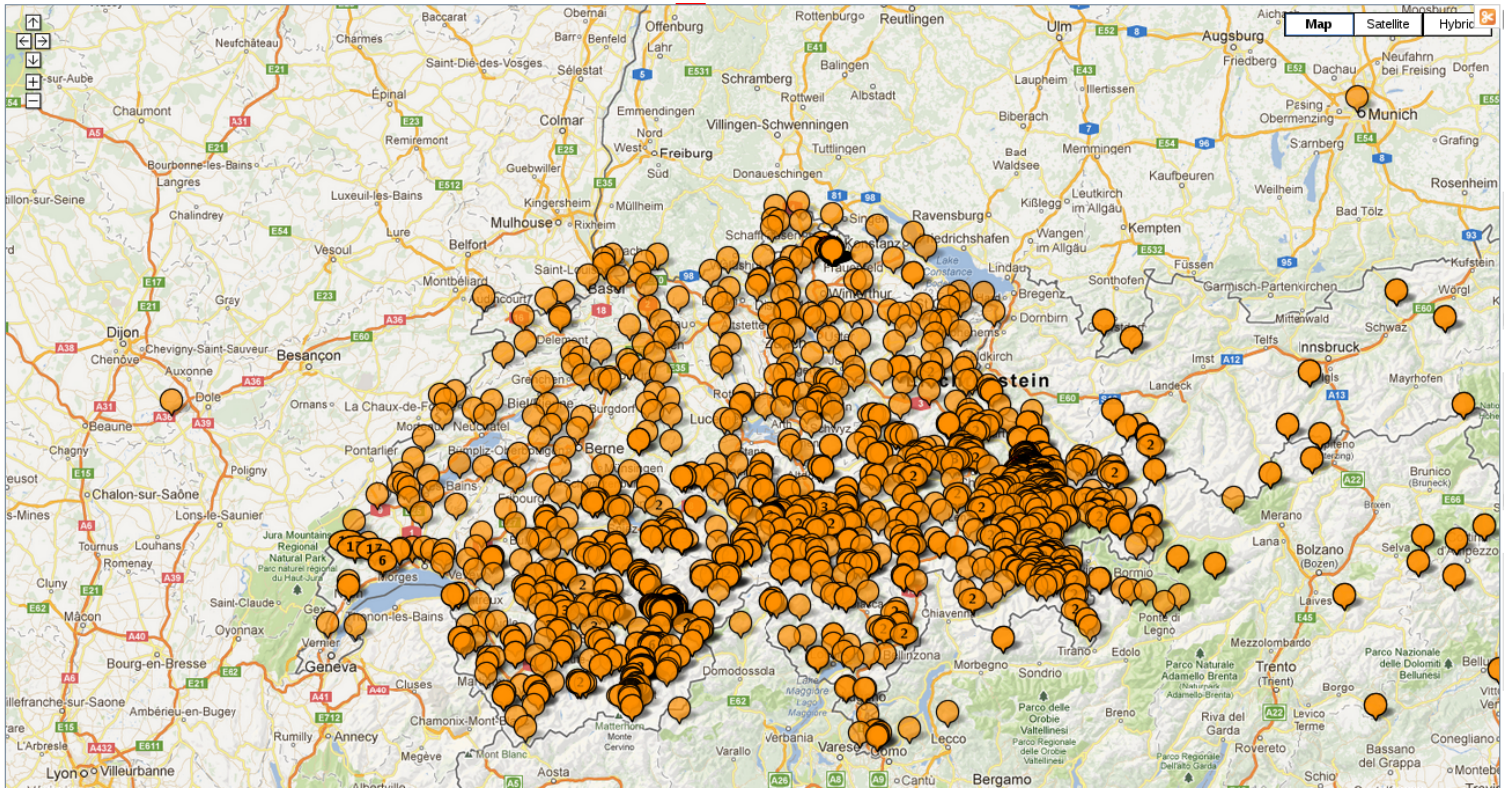
Distributed Information Systems Laboratory
EPFL, Switzerland

11th April, 2013

The SwissEx Project



The SwissEx Project – Challenge



- **Data Scale**
  - A single sensor can collect a large number of samples.
  - Smartphone accelerometers can produce huge datasets [Yan'12].
  - **Cost of sensors goes down, size of data goes up.**

Motivating Examples

Stock Markets

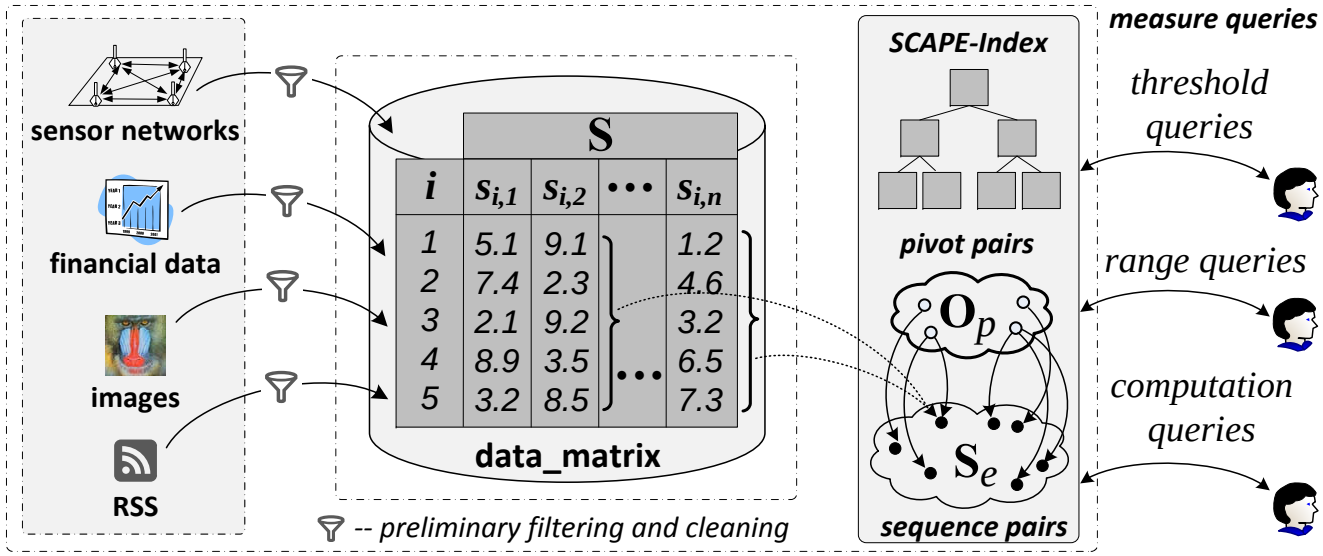
Given the intra-day stock quotes of n stocks obtained at a sampling interval Δt, return the correlation coefficients of the $\frac{n(n-1)}{2}$ pairs of stocks on a given day.

Data Centers

Find the computers in a data center whose number of TCP connections are correlated ≥ 0.9.

- Computing statistical measures is a fundamental primitive in time-series data processing.

Framework Overview



- **Given:** Data matrix $\mathbf{S}$ with n time-series (stocks, sensors, etc.).

- **Problems:**
  - There are $\frac{n(n-1)}{2}$ or $\mathcal{O}(n^2)$ pairs of time series.
  - **Threshold Queries:** Find the pairs of time series such that a statistical measure between them is $\geq \tau$.

Existing Work (Correlation Coefficient)

- Correlation coefficient as Euclidean distance [Zhu'02]:

$$corr(x, y) = 1 - \frac{1}{2} d^2(\hat{x}, \hat{y}),$$

$d^2(\hat{x}, \hat{y})$ – Euclidean distance, $\hat{x}$ and $\hat{y}$ are normalized x and y.

- Compute the DFT of x and y as $\hat{X}$ and $\hat{Y}$. $d^2(\hat{x}, \hat{y}) = d^2(\hat{X}, \hat{Y})$.

- **Threshold Query:** [Mueen'10]

$$corr(x, y) \geq \tau \implies d_k(\hat{X}, \hat{Y}) \leq \sqrt{2m(1 - \tau)},$$

where m is time series length and $d_k(\hat{X}, \hat{Y})$ is truncated DFT.

Drawbacks:

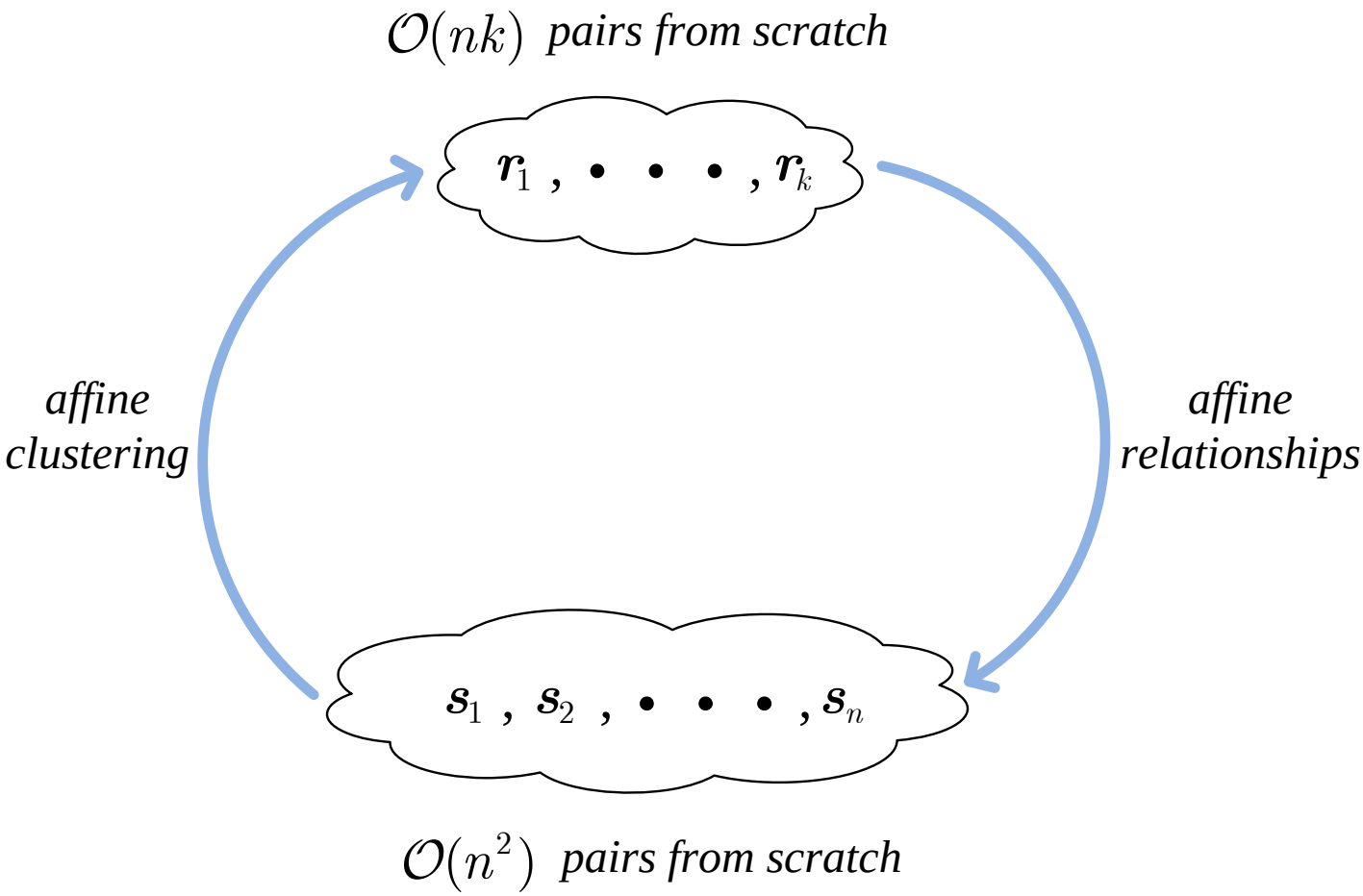
- Only one similarity measure (correlation coefficient) is handled.
- Internal structure of relationships among time series is not exploited.

Notational Setup

- **Given:** Data matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$ containing time series $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
- **Compute:**
  - Covariance matrix $\Sigma(\mathbf{S}) \in \mathbb{R}^{n \times n}$
    - $\Sigma_{uv}(\mathbf{S})$ is the covariance between time series $\boldsymbol{s}_u$ and $\boldsymbol{s}_v$.
  - Dot product matrix $\Pi(\mathbf{S}) \in \mathbb{R}^{n \times n}$
    - $\Pi_{uv}(\mathbf{S})$ is the dot product between time series $\boldsymbol{s}_u$ and $\boldsymbol{s}_v$.

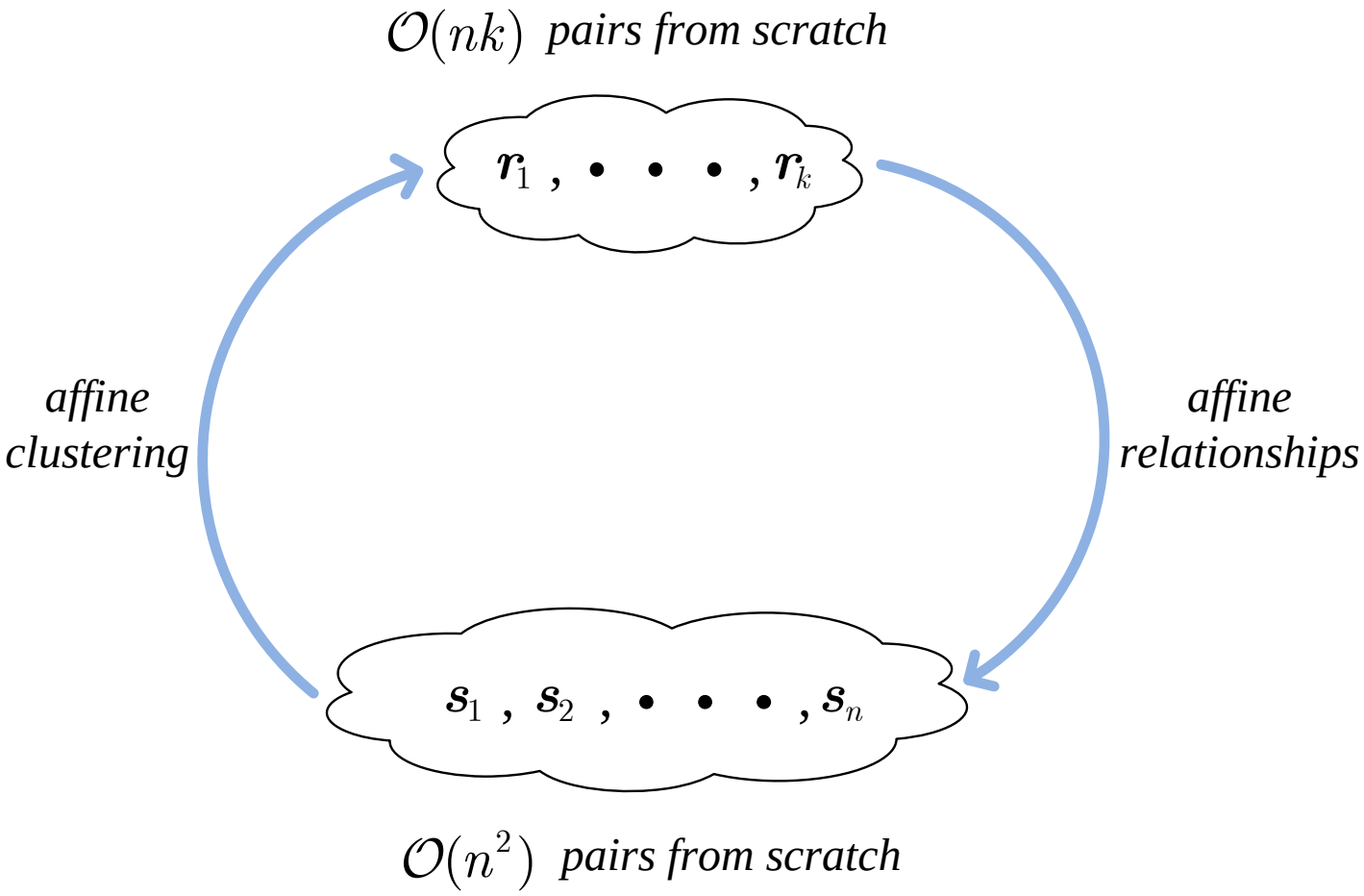
- The number of entries in $\Sigma(\mathbf{S})$ and $\Pi(\mathbf{S})$ are $\mathcal{O}(n^2)$.
- We choose covariance and dot product, since they are fundamental statistical measures and other measure can be derived from them.

Solution Overview



- In practice $k << n$, so we obtain **orders of magnitude** improvement.
- Affine relationships are used for computing **many** statistical measures.

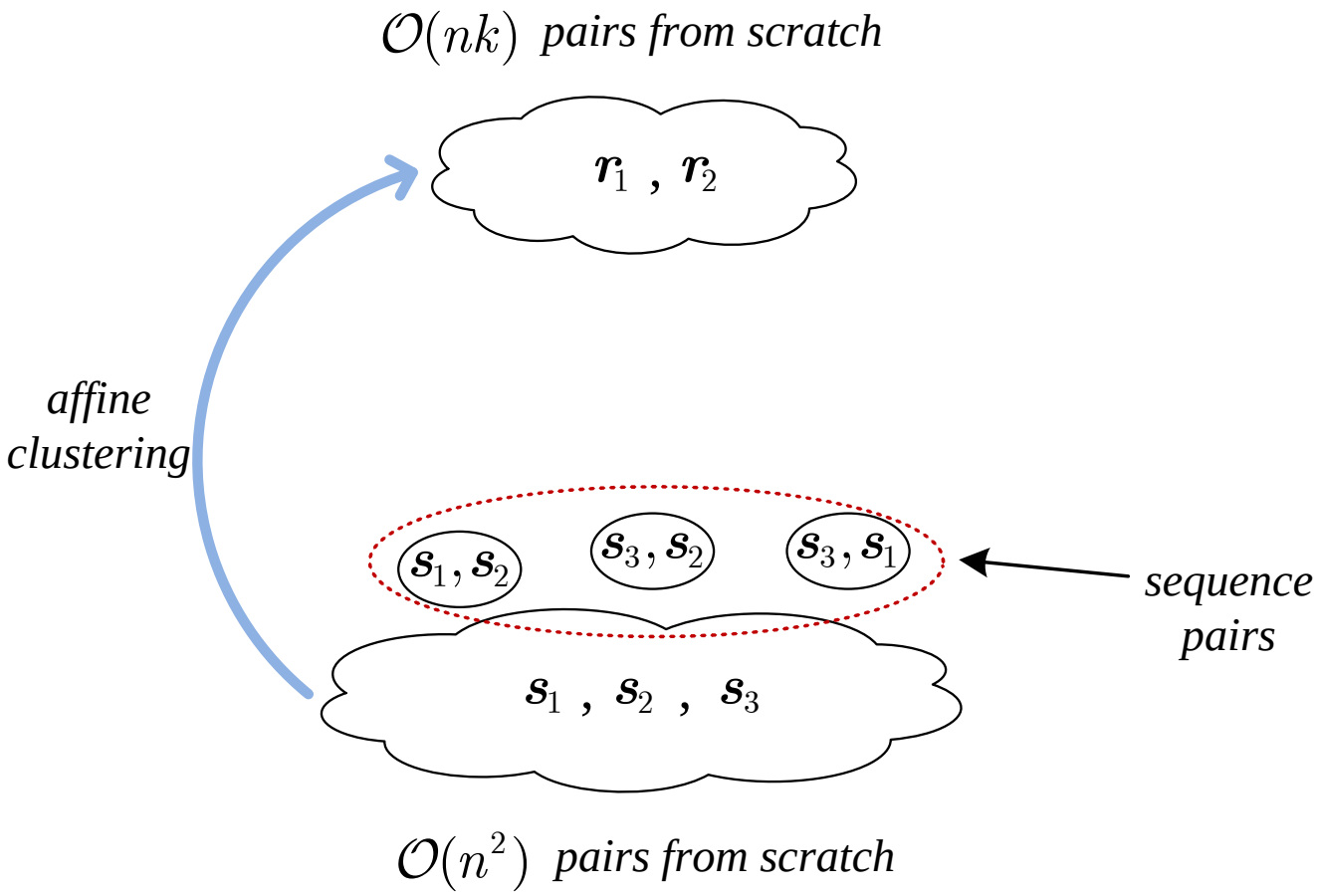
Solution Overview



Key Questions:

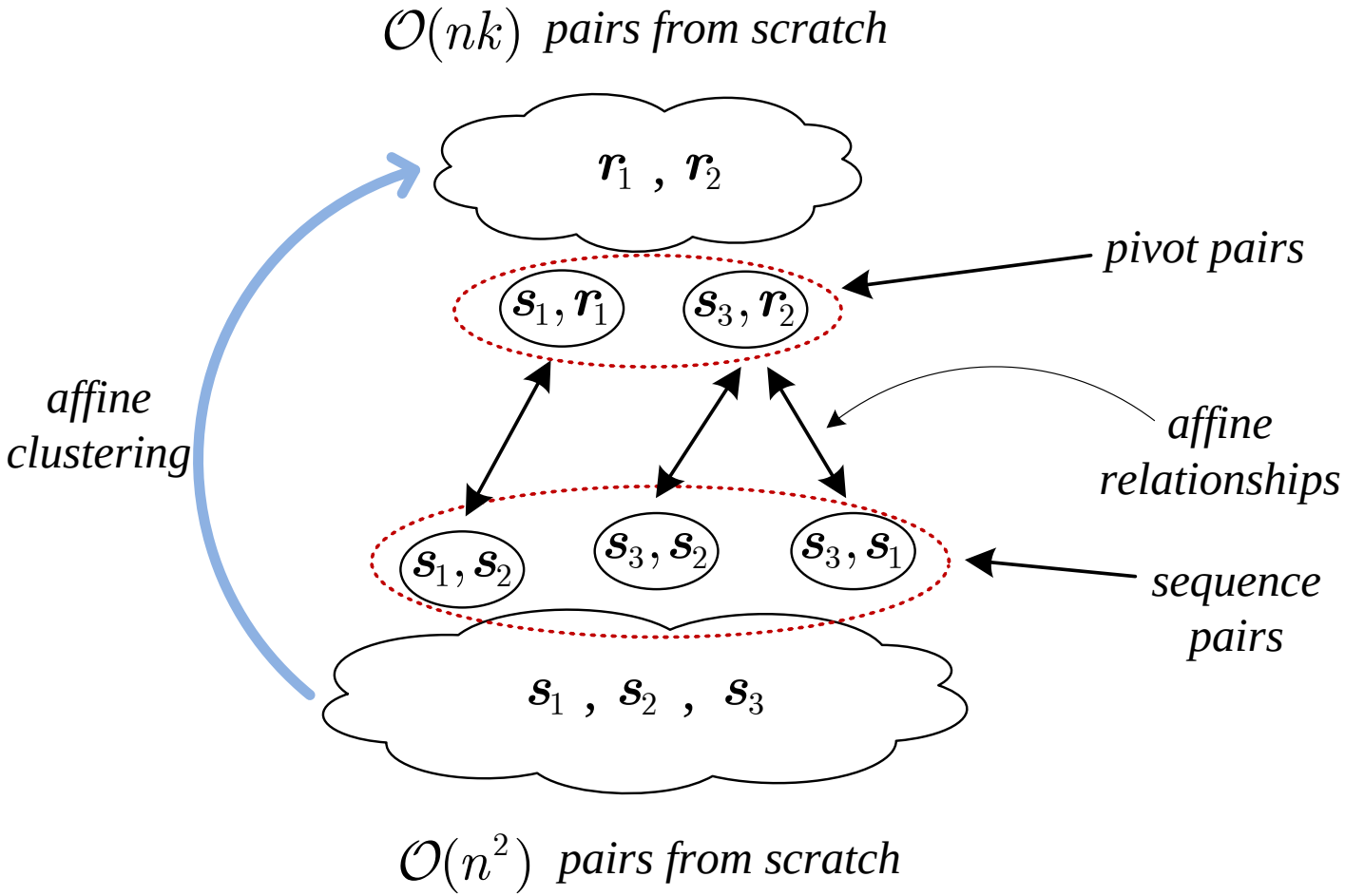
- How to choose or compute $r_1, r_2, \ldots, r_k$? $\longrightarrow$ **Affine Clustering**
- How to use $r_1, r_2, \ldots, r_k$ for computing $\Sigma(\mathbf{S})$ and $\Pi(\mathbf{S})$?

Computing Statistical Measures



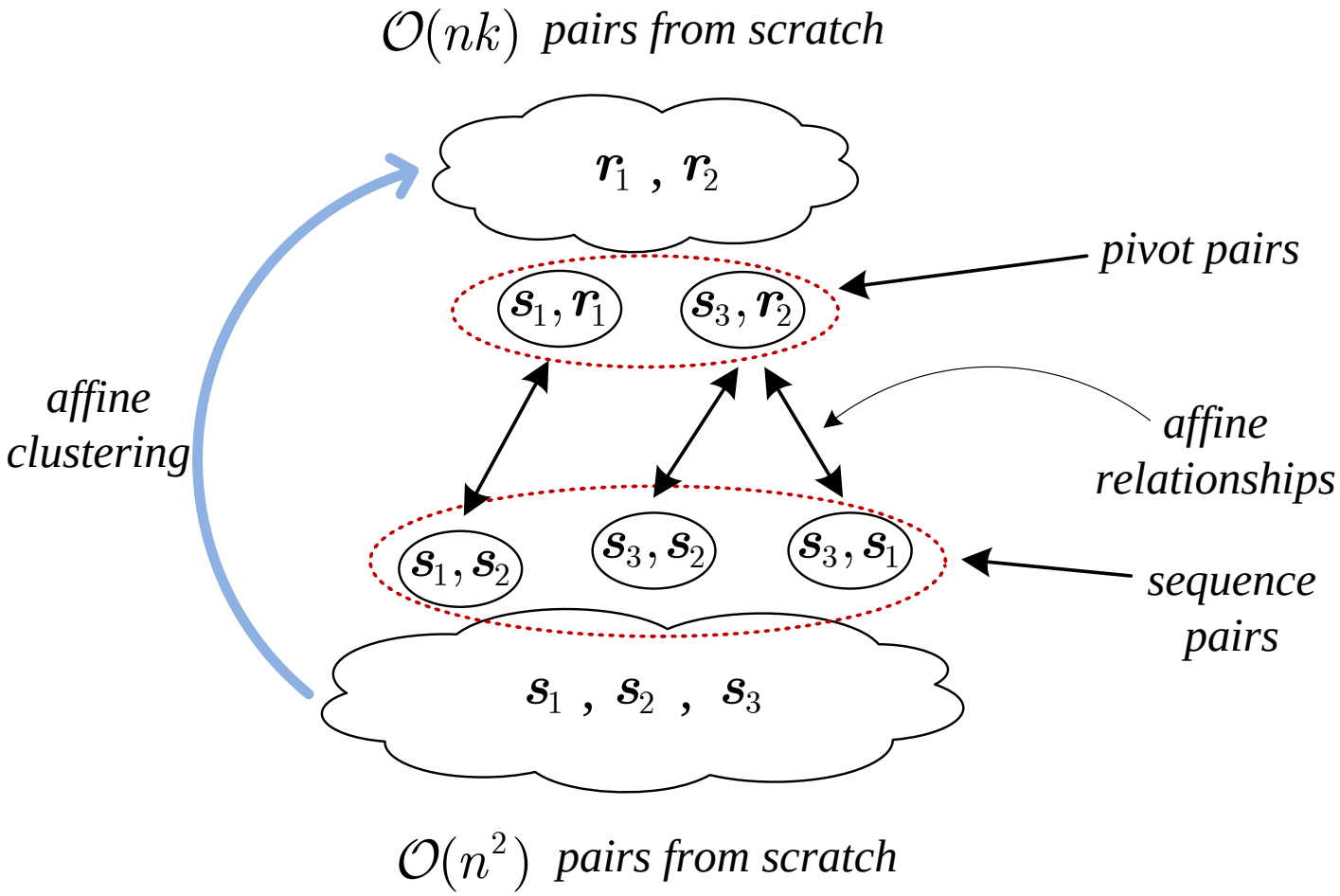
- Form the **sequence pairs** $[s_1, s_2]$, $[s_3, s_2]$, and $[s_3, s_1]$.

Computing Statistical Measures



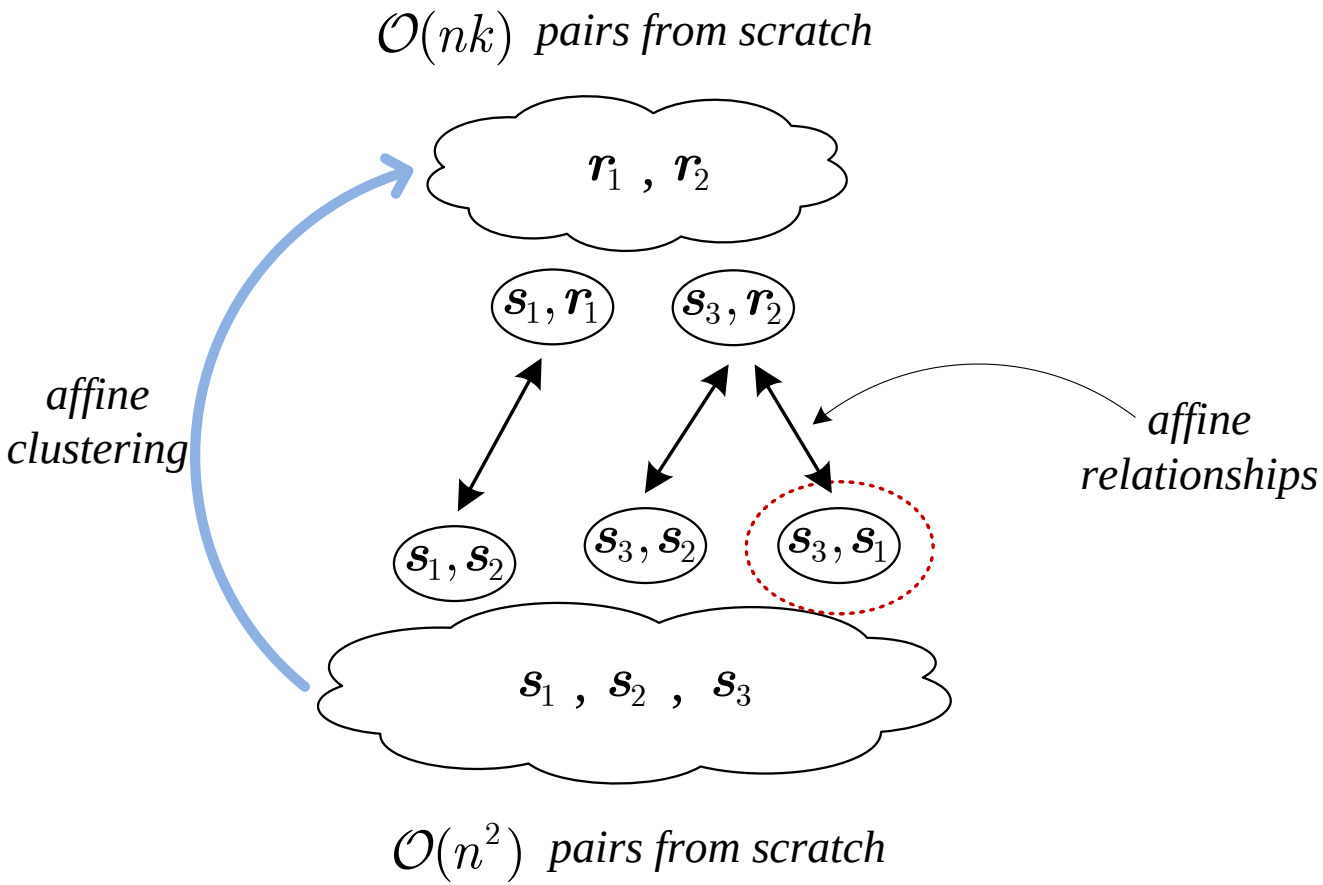
- Replace one time series from the sequence pair with its cluster center.
- The resulting pair is called the **pivot pair**.

Computing Statistical Measures



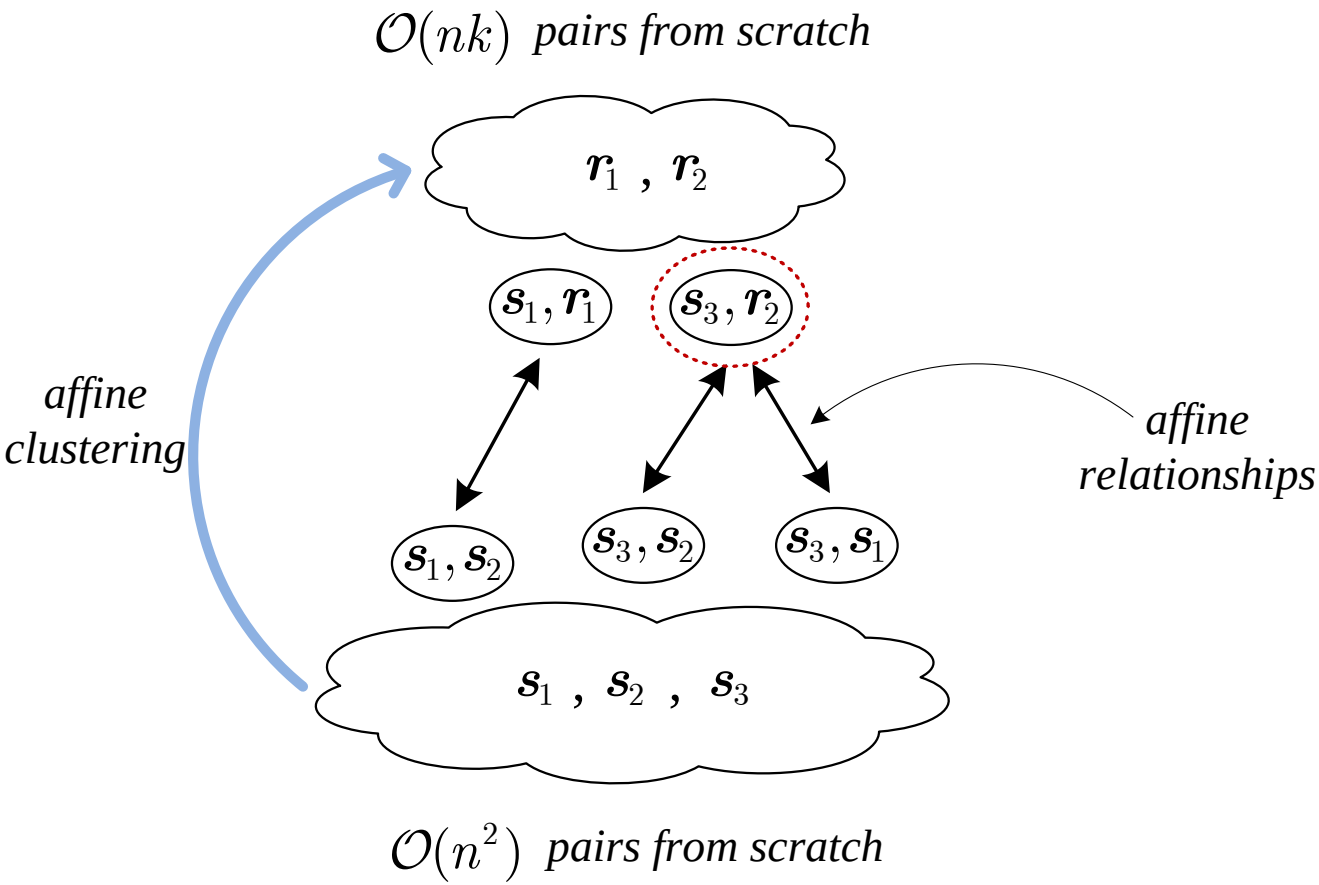
- **Observe:** There can only be $\mathcal{O}(nk)$ pivot pairs.
- **Real Example:** #time series 996, #pivot pairs 5.9K, #sequence pairs 495K

Computing Covariance



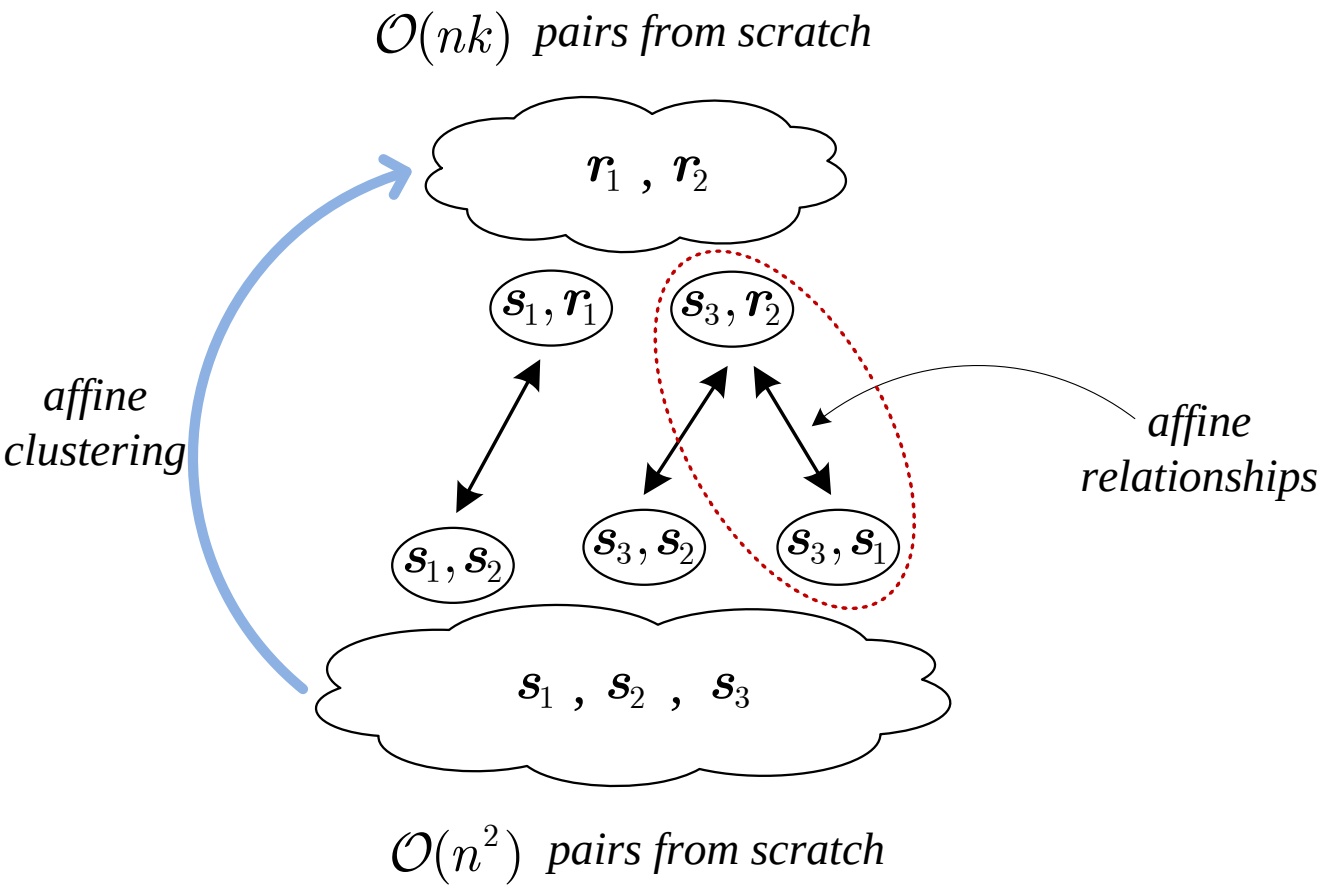
- **Task:** Compute the covariance of sequence pair $[s_3, s_1]$.

Computing Covariance



- **Compute:** Covariance of pivot pair $\Sigma([s_3, r_2])$ using brute force.

Computing Covariance

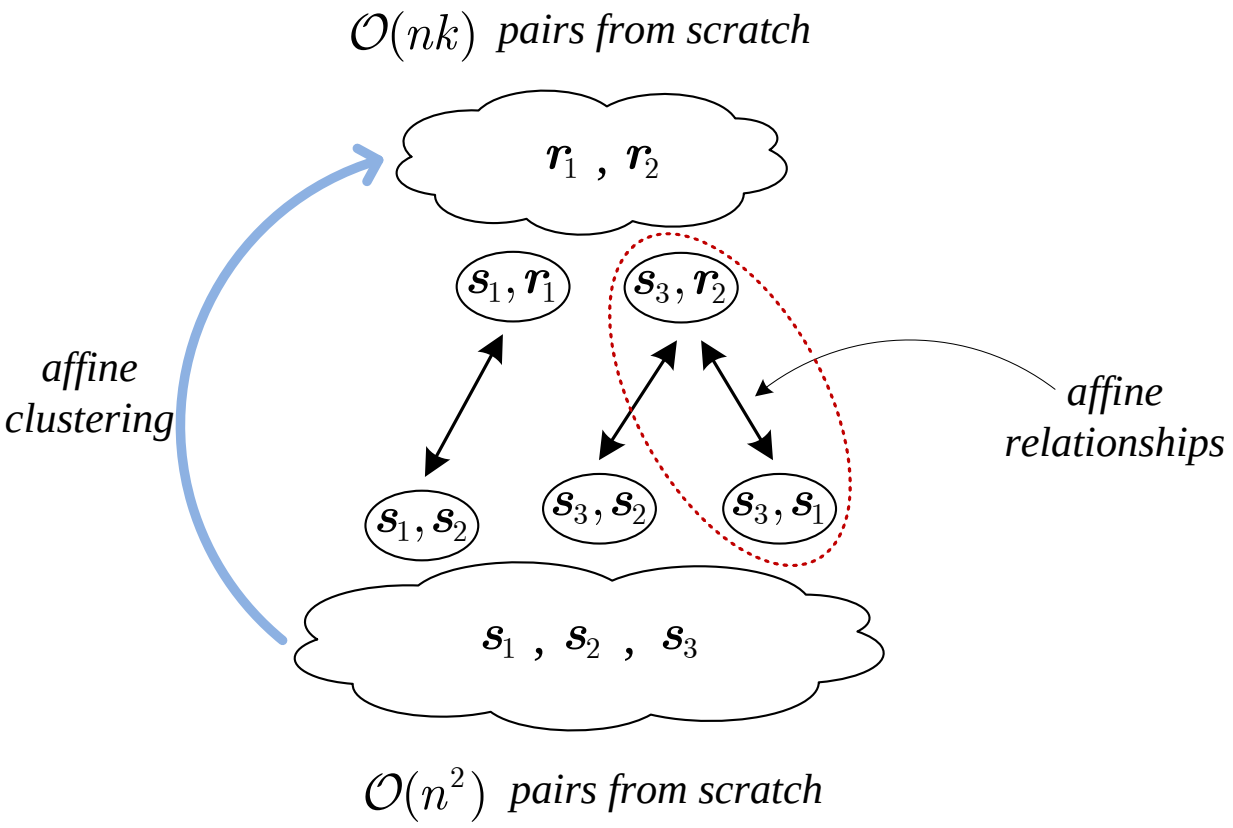


- **Approximate:** Covariance of sequence pair $[s_3, s_1]$ using:

$$\Sigma([s_3, s_1]) = \mathbf{A}^\top \Sigma([s_3, r_2])\mathbf{A},$$

where $\mathbf{A} \in \mathbb{R}^{2\times 2}$ is non-singular.

Computing Covariance



- $\mathbf{A}$ and $\boldsymbol{b}$ are estimated using least-squares method from the following affine transformation:

$$[\boldsymbol{s}_3, \boldsymbol{s}_1] = [\boldsymbol{s}_3, \boldsymbol{r}_2]\mathbf{A} + \mathbb{1}_m \boldsymbol{b}^\top$$

  where $\boldsymbol{b} \in \mathbb{R}^2$, $\mathbb{1}_m = (1, \ldots, 1)^\top \in \mathbb{R}^m$.

- Similarly compute covariance of all the other sequence pairs.

Computing Dot Product

- Procedure similar to covariance.
- **Compute:** Dot product of pivot pair $\Pi([\boldsymbol{s}_3, \boldsymbol{r}_2])$ using brute force.
- **Approximate:** Dot product of sequence pair using:

$$\Pi_{31}(\mathbf{S}) = \boldsymbol{a}_1^\top \cdot \Pi([\boldsymbol{s}_3, \boldsymbol{r}_2]) \cdot \boldsymbol{a}_2 + \boldsymbol{b}^\top \mathbf{A}^\top \begin{pmatrix} \mathsf{h}_3 \\ \mathsf{h}_1 \end{pmatrix},$$

  where $\mathsf{h}_3 = \sum_{i=1}^{m} s_{i3}$, $\mathsf{h}_1 = \sum_{i=1}^{m} s_{i1}$.

- Compute dot product of all the other sequence pairs.

Observe:

- For computing both, covariance and dot product, we used the same $\mathbf{A}$ and $\boldsymbol{b}$.
- Computation of a statistical measure using $\mathbf{A}$ and $\boldsymbol{b}$ is very efficient.

Approximation Error

Lemma – Dot Product

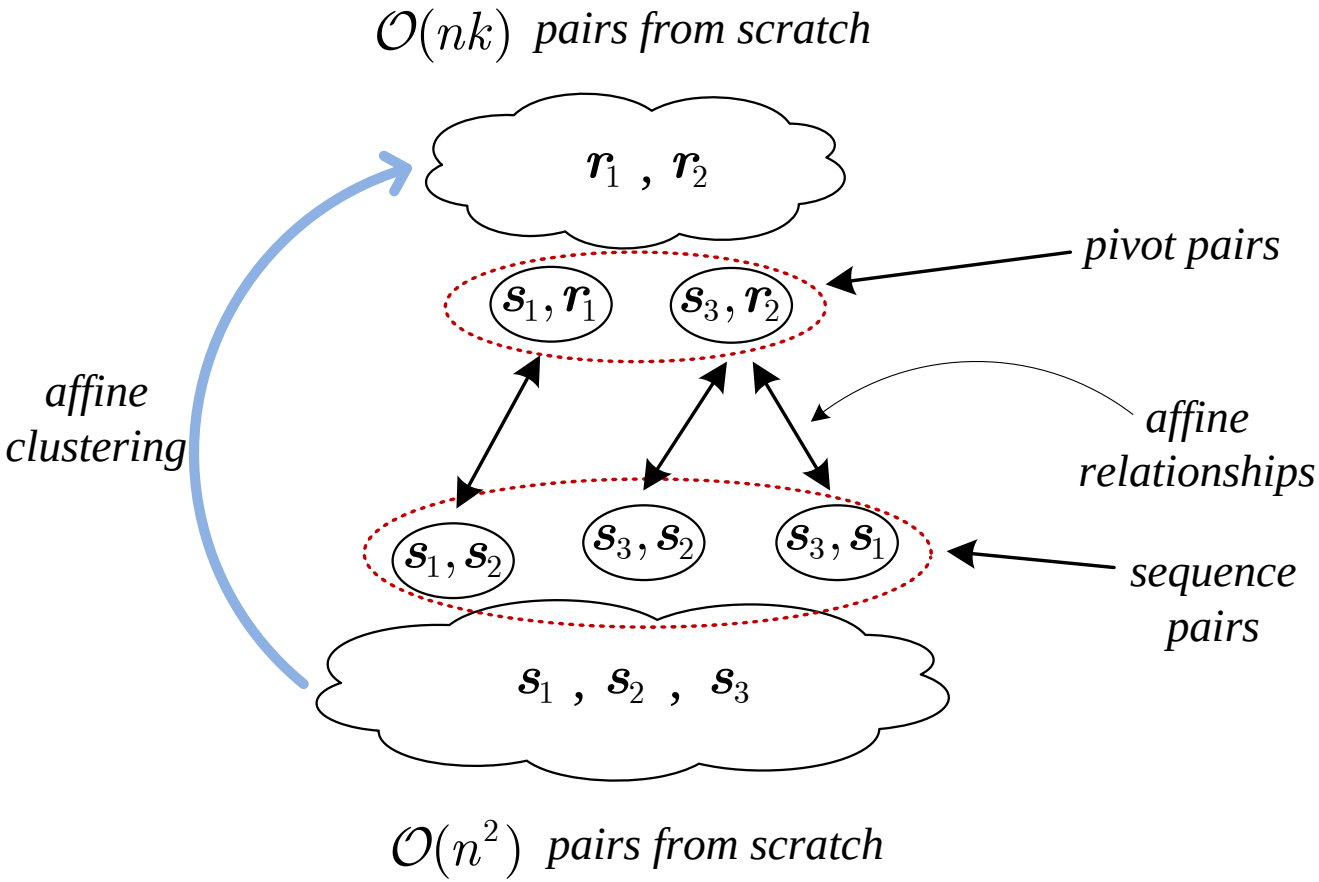
If there is a common time series between the sequence and pivot pairs, then the dot product can be computed with **zero error**.

Other Measures

Affine clustering ensures that if the second time series in the pivot pair is chosen as the cluster center, then **minimal approximation error** is produced.

Affine Relationships – Summary

- **Affine Relationship (Example):** $[\boldsymbol{s}_3, \boldsymbol{s}_1] = [\boldsymbol{s}_3, \boldsymbol{r}_2]\mathbf{A} + \mathbb{1}_m \boldsymbol{b}^\top$.



- Affine relationship is in between a pivot pair and a sequence pair characterized by $(\mathbf{A}, \boldsymbol{b})$.

- **Problems:**
  - ~~There are $\frac{n(n-1)}{2}$ or $\mathcal{O}(n^2)$ pairs of time series.~~
  - **Threshold Queries: Find the pairs of time series such that a statistical measure between them is $\geq \tau$.**

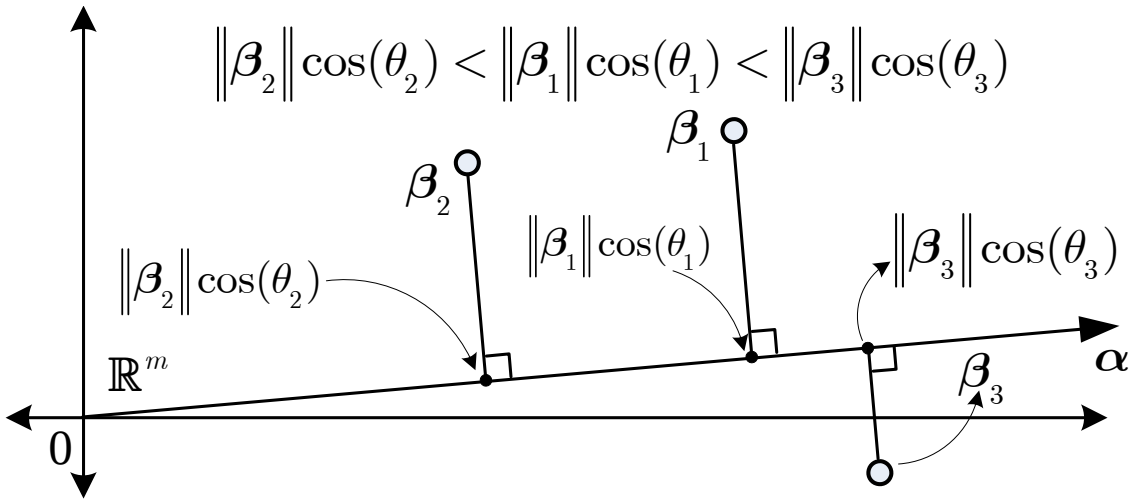
Scalar Projection (SCAPE) Index – Features

- A versatile index capable of indexing and querying many statistical measures.
- **Queries:** Find all the pairs of time series such that their covariance is $\geq \tau$ (**threshold**) or is $\geq \tau_l$ and $\leq \tau_u$ (**range**).
- Derived statistical measures need not be indexed separately
  - Correlation coefficient is derived by normalization of the covariance.
  - Correlation coefficient need not be indexed if the covariance is indexed.
  - Queries for the derived measures are processed using index-based pruning.

- **Idea:** Find a way to **order affine relationships** so that we can discard the unnecessary ones quickly.

Scalar Projection (SCAPE) Index – Intuition

- **Given:** Vectors $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_3$, and $\boldsymbol{\alpha}$.
- **Find (Threshold Query):** Vectors $\boldsymbol{\beta}_i$ such that $\boldsymbol{\alpha}^\top \boldsymbol{\beta}_i \geq \tau$.



- **Observe:** $\boldsymbol{\alpha}^\top \boldsymbol{\beta}_i \geq \tau \implies \|\boldsymbol{\beta}_i\| \cos(\theta_i) \geq \frac{\tau}{\|\boldsymbol{\alpha}\|}$, where θ_i is the angle between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_i$.
- It is **sufficient** to index the **scalar projections** $\|\boldsymbol{\beta}_i\| \cos(\theta_i)$ for processing the threshold query.

Example of α and β for Covariance

- **Consider Affine Relationship**: $[\boldsymbol{s}_3, \boldsymbol{s}_1] = [\boldsymbol{s}_3, \boldsymbol{r}_2]\mathbf{A} + \mathbb{1}_m \boldsymbol{b}^\top$

We know that: $\Sigma([\boldsymbol{s}_3, \boldsymbol{s}_1]) = \mathbf{A}^\top \Sigma([\boldsymbol{s}_3, \boldsymbol{r}_2])\mathbf{A}$

$$= \mathbf{A}^\top \begin{bmatrix} var(\boldsymbol{s}_3) & cov(\boldsymbol{s}_3, \boldsymbol{r}_2) \\ cov(\boldsymbol{s}_3, \boldsymbol{r}_2) & var(\boldsymbol{r}_2) \end{bmatrix} \mathbf{A}$$

- Let $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = [\boldsymbol{a}_1, \boldsymbol{a}_2]$. But, since $\boldsymbol{a}_1 = (1, 0)^\top$:

$$cov(\boldsymbol{s}_3, \boldsymbol{s}_1) = [var(\boldsymbol{s}_3), cov(\boldsymbol{s}_3, \boldsymbol{r}_2)] \times \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix}$$

$$cov(\boldsymbol{s}_3, \boldsymbol{s}_1) = \boldsymbol{\alpha}^\top \times \boldsymbol{\beta_i}$$

- Include pair $(\boldsymbol{s}_3, \boldsymbol{s}_1)$ in the query result if $\|\boldsymbol{\beta}_i\| \cos(\theta_i) \geq \frac{\tau}{\|\boldsymbol{\alpha}\|}$.

Experimental Evaluation

- **stock-data:** stocks from S&P and ETFs.
- **sensor-data:** 134 sensors monitoring ambient temperature, relative humidity, and surface temperature.

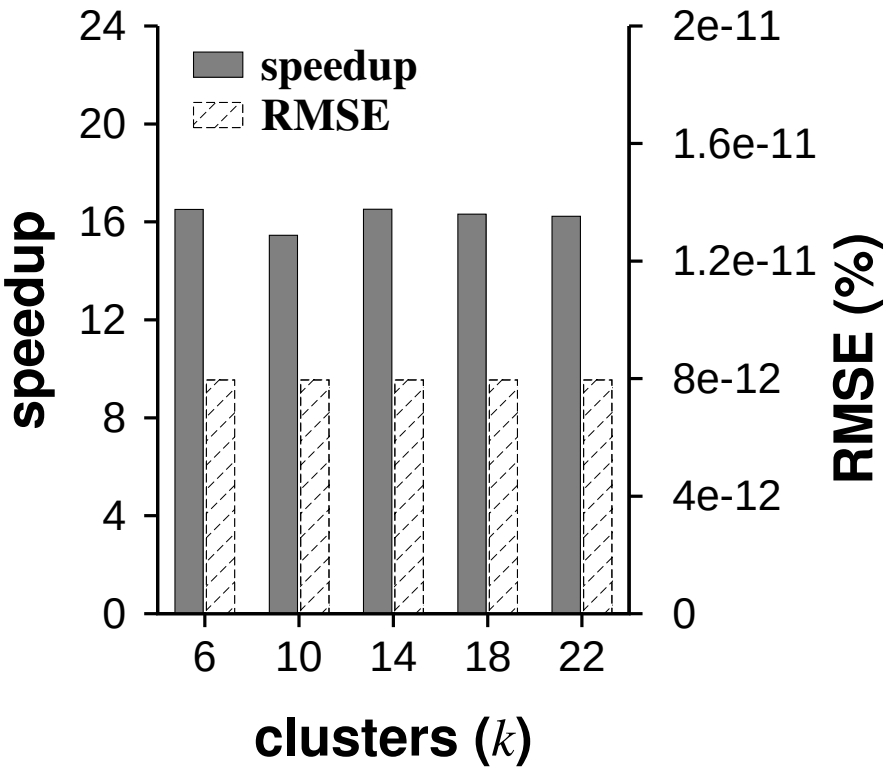
Table: Summary of the datasets.

	sensor-data	*stock-data*
sampling interval	2 min.	1 min.
#time series (n)	670	996
#samples per time series (m)	720	1,950
max. affine relationships	224,115	495,510

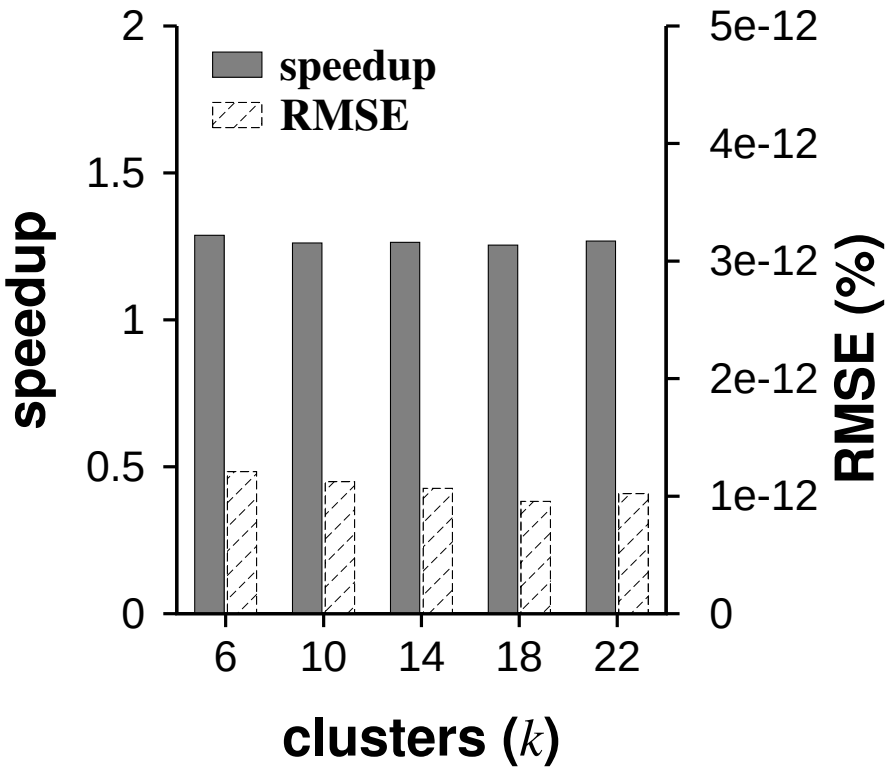
Various Methods

- W_N: a naïve or brute force method.
- W_A: using affine relationships.
- W_F: uses 5 largest DFT coefficients [Li'96, Zhu'02, Mueen'10].

Efficiency and Accuracy – Tradeoff



(a) covariance (stock-data)



(b) dot product (sensor-data)

Speedup varies from a factor 1.3 for simple measures (dot product) to factor 3500 for complex measures (mode).

Threshold Queries – SCAPE Index

Query type	Measure	Speedup		
		W_N	W_A	W_F
Threshold	correlation coefficient	59x	13.4x	32x
	covariance	160x	21x	×
	dot product	41x	35x	×
	median	5x	1.1x	×

- **Speedup using the SCAPE index when the largest result set is returned.**
- **Range query results are discussed in the paper.**

Conclusions and Future Work

Conclusions:

- Proposed methods for **improving efficiency** of computing many statistical measures.
- **Indexing mechanisms** for processing threshold and range queries.
- **Reducing the complexity** of computing pair-wise measures from $\mathcal{O}(n^2)$ to $\mathcal{O}(nk)$.

Conclusions and Future Work

Future Work:

- Pruning affine relationships
  - Can we prune affine relationships based on domain knowledge, query requirements, low correlation, etc?
- Dynamic affine relationships
  - How to handle evolving affine relationships?
- Distributed Query Processing
  - Many datasets cannot be stored on a single computing device.

Thank You.

Questions?

Saket Sathe
saket.sathe@epfl.ch

Karl Aberer
karl.aberer@epfl.ch