Estimating Statistical Characteristics Under Interval Uncertainty and Constraints: Mean, Variance, Covariance, and Correlation

Ali Jalal-Kamali

Department of Computer Science
The University of Texas at El Paso
El Paso, TX 79968, USA

December 2011

1. Need for Estimating Statistical Characteristics

- Often, we have a sample of values $x_1, \ldots, x_n$ corresponding to objects of a certain type.
- A standard way to describe the population is to describe its mean, variance, and standard deviation:

$$E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i; \quad V = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E)^2; \quad \sigma = \sqrt{V}.$$

- When we measure two quantities x and y:
  - we describe the means E_x, E_y, variances V_x, V_y and standard deviations σ_x, σ_y of both;
  - we also estimate their covariance and correlation:

$$C_{x,y} = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x) \cdot (y_i - E_y); \quad \rho_{x,y} = \frac{C_{x,y}}{\sigma_x \cdot \sigma_y}.$$

2. Case of Interval Uncertainty

- The above formulas assume that we know the exact values of the characteristics $x_1, \ldots, x_n$.
- In practice, values usually come from measurements, and measurements are never absolutely exact.
- The measurement results $\widetilde{x}_i$ are, in general, different from the actual (unknown) values x_i: $\widetilde{x}_i \neq x_i$.
- Often, it is assumed that we know the probability distribution of the measurement errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$.
- However, often, the only information available is the upper bound on the measurement error: $|\Delta x_i| \leq \Delta_i$.
- In this case, the only information that we have about the actual value x_i is that $x_i \in \mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$, where
$$\underline{x}_i = \widetilde{x}_i - \Delta_i, \quad \overline{x}_i = \widetilde{x}_i + \Delta_i.$$

3. Need to Preserve Privacy in Statistical Databases

- In order to find relations between different quantities, we *collect* a large amount of *data*.
- *Example:* we collect *medical* data to try to find correlations between a disease and lifestyle factors.
- In some cases, we are looking for commonsense correlations, e.g., between smoking and lung diseases.
- For statistical databases to be most useful, we need to *allow researchers* to *ask* arbitrary *questions*.
- However, this may inadvertently *disclose* some *private information* about the individuals.
- Therefore, it is desirable to *preserve privacy* in statistical databases.

4. Intervals as a Way to Preserve Privacy in Statistical Databases

- One way to preserve privacy is to store *ranges* (intervals) rather than the exact data values.
- This makes sense from the viewpoint of a statistical database.
- In general, this is how data is often collected:
  - we set some *threshold* values $t_0, \ldots, t_N$ and
  - ask a person whether the actual value x_i is in the interval $[t_0, t_1]$, or $\ldots$, or in the interval $[t_{N-1}, t_N]$.
- As a result, for each quantity x and for each person i:
  - instead of the *exact* value x_i,
  - we store an *interval* $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ that contains x_i.
- Each of these intervals coincides with one of the given ranges $[t_0, t_1]$, $[t_1, t_2]$, $\ldots$, $[t_{N-1}, t_N]$.

5. Need to Estimate Statistical Characteristics $S(x_1, \ldots)$ Under Interval Uncertainty

- In both situations of measurement errors and privacy:
  - instead of the actual values x_i (and y_i),
  - we only know the intervals $\mathbf{x}_i$ (and $\mathbf{y}_i$) that contain the actual values.
- Different values of x_i (and y_i) from these intervals lead, in general, to different values of each characteristic.
- It is desirable to find the *range* of possible values of these characteristics when $x_i \in \mathbf{x}_i$ (and $y_i \in \mathbf{y}_i$):

$$\mathbf{S} = \{S(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\};$$

$$\mathbf{S} = \{S(x_1, \ldots, x_n, y_1, \ldots, y_n) :$$
$$x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n, y_1 \in \mathbf{y}_1, \ldots, y_n \in \mathbf{y}_n\}.$$

6. Estimating Statistical Characteristics under Interval Uncertainty: What is Known

- The mean $E = \frac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i$ is an increasing function of all its inputs $x_1, \ldots, x_n$.
- Hence, E is the smallest when all the inputs $x_i \in [\underline{x}_i, \overline{x}_i]$ are the smallest ($x_i = \underline{x}_i$): $\underline{E} = \frac{1}{n} \cdot \sum\limits_{i=1}^{n} \underline{x}_i$; $\overline{E} = \frac{1}{n} \cdot \sum\limits_{i=1}^{n} \overline{x}_i$.
- However, variance, covariance, and correlation are, in general, non-monotonic.
- It is known that computing the ranges of these characteristics under interval uncertainty is NP-hard.
- The problem gets even more complex because in practice, we often have additional constraints.

7. Formulation of the Problem and What We Did

- *Reminder:* under interval uncertainty,
  - in the absence of constraints, computing the range $\mathbf{E}$ of the mean E is feasible;
  - computing the ranges $\mathbf{V}$, $\mathbf{C}$, and $[\underline{\rho}, \overline{\rho}]$ is NP-hard.
- *Problem:* find practically useful cases when feasible algorithms are possible.
- *What is known:* for V, we can feasibly compute:
  - one of the endpoints ($\underline{V}$) – always; and
  - both endpoints – in the privacy case.
- *We designed:* feasible algorithms for computing:
  - the range $\mathbf{E}$ under constraints;
  - the range $\mathbf{C}$ in the privacy case; and
  - one of the endpoints $\underline{\rho}$ or $\overline{\rho}$.

8. Computing E under Variance Constraints

- In the previous expressions, we assumed only that x_i belongs to the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.
- In some cases, we have an additional *a priori* constraint on x_i: $V \le V_0$, for a given V_0.
- For example, we know that within a species, there can be ≤ 0.1 variation of a certain characteristic.
- Thus, we arrive at the following problem:
  - *given:* n intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ and a number $V_0 \ge 0$;
  - *compute:* the range

$$[\underline{E}, \overline{E}] = \{E(x_1, \ldots, x_n) \,:\, x_i \in \mathbf{x}_i \,\&\, V(x_1, \ldots, x_n) \le V_0\};$$

  - *under the assumption* that there exist values $x_i \in \mathbf{x}_i$ for which $V(x_1, \ldots, x_n) \le V_0$.
- This is a problem that we will solve in this thesis.

9. Cases Where This Problem Is (Relatively) Easy to Solve

- *First case:* V_0 is $\geq$ the largest possible value $\overline{V}$ of the variance corresponding to the given sample.
- In this case, the constraint $V \leq V_0$ is always satisfied.
- Thus, in this case, the desired range simply coincides with the range of all possible values of E.
- *Second case:* $V_0 = 0$.
- In this case, the constraint $V \leq V_0$ means that the variance V should be equal to 0, i.e., $x_1 = \ldots = x_n$.
- In this case, we know that this common value x_i belongs to each of n intervals $\mathbf{x}_i$.
- So, the set of all possible values E is the intersection:

$$E = \mathbf{x}_1 \cap \ldots \cap \mathbf{x}_n.$$

10. Main Result: A Feasible Algorithm that Computes $[\underline{E}, \overline{E}]$ under Interval Uncertainty and Variance Constraint

- In the general case, first, we compute the values

$$E^- \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^n \underline{x}_i \text{ and } V^- \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^n (\underline{x}_i - E^-)^2;$$

$$E^+ \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^n \overline{x}_i \text{ and } V^+ \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^n (\overline{x}_i - E^+)^2.$$

- If $V^- \le V_0$, then we return $\underline{E} = E^-$.
- If $V^+ \le V_0$, then we return $\overline{E} = E^+$.
- If $V_0 < V^-$ or $V_0 < V^+$, we sort the all $2n$ endpoints $\underline{x}_i$ and $\overline{x}_i$ into a non-decreasing sequence

$$z_1 \le z_2 \le \ldots \le z_{2n}$$

and consider $2n - 1$ *zones* $[z_k, z_{k+1}]$, $k = 1, \ldots, 2n - 1$.

11. Algorithm (cont-d)

- For each zone $[z_k, z_{k+1}]$, we take:
  - for every i for which $\overline{x}_i \le z_k$, we take $x_i = \overline{x}_i$;
  - for every i for which $z_{k+1} \le \underline{x}_i$, we take $x_i = \underline{x}_i$;
  - for every other i, we take $x_i = \alpha$; let us denote the number of such i's by n_k.
- The value α is determined from the condition that for the selected vector x, we have $V(x) = V_0$:

$$\frac{1}{n} \cdot \left(\sum_{i:\overline{x}_i \le z_k} (\overline{x}_i)^2 + \sum_{i:z_{k+1} \le \underline{x}_i} (\underline{x}_i)^2 + n_k \cdot \alpha^2 \right) -$$

$$\frac{1}{n^2} \cdot \left(\sum_{i:\overline{x}_i \le z_k} \overline{x}_i + \sum_{i:z_{k+1} \le \underline{x}_i} \underline{x}_i + n_k \cdot \alpha \right)^2 = V_0.$$

12. Algorithm: Last Part

- If none of the two roots of the above quadratic equation belongs to the zone, this zone is dismissed.
- If one or more roots belong to the zone, then for each of these roots α, we compute the value

$$E_k(\alpha)=\frac{1}{n}\cdot\left(\sum_{i:\overline{x}_i\le z_k}\overline{x}_i+\sum_{i:z_{k+1}\le \underline{x}_i}\underline{x}_i+n_k\cdot\alpha\right).$$

- After that:
  - if $V_0<V^-$, we return the smallest of the values $E_k(\alpha)$ as $\underline{E}$:
  $$\underline{E}=\min_{k,\alpha}E_k(\alpha);$$
  - if $V_0<V^+$, we return the largest of the values $E_k(\alpha)$ as $\overline{E}$:
  $$\overline{E}=\max_{k,\alpha}E_k(\alpha).$$

13. Computation Time of the Algorithm

- Sorting $2n$ numbers requires time $O(n \cdot \log(n))$.
- Once the values are sorted, we can then go zone-by-zone, and perform the corresponding computations:
  - for each of $2n - 1$ zones,
  - we compute several sums of n numbers.
- The sum for the first zone requires linear time.
- Once we have the sums for one zone, computing the sums for the next zone requires changing a few terms.
- Each value x_i changes status once, so overall, to compute all these sums, we need linear time $O(n)$.
- So, the total time is:

$$O(n \cdot \log(n)) + O(n) = O(n \cdot \log(n)).$$

14. Toy Example

- Case: $n = 2$, $\mathbf{x}_1 = [-1, 0]$, $\mathbf{x}_2 = [0, 1]$, $V_0 = 0.16$.
- In this case, according to the above algorithm, we compute the values

$$E^- = \frac{1}{2} \cdot (-1 + 0) = -0.5; \quad E^+ = \frac{1}{2} \cdot (0 + 1) = 0.5;$$

$$V^- = \frac{1}{2} \cdot (((-1) - (-0.5))^2 + (0 - (-0.5))^2) = 0.25;$$

$$V^+ = \frac{1}{2} \cdot ((0 - 0.5)^2 + (1 - 0.5)^2) = 0.25.$$

- Here, $V_0 < V^-$ and $V_0 < V^+$, so we consider zones.
- By sorting the 4 endpoints -1, 0, 0, and 1, we get

$$z_1 = -1 \le z_2 = 0 \le z_3 = 0 \le z_4 = 1.$$

- Thus, here, we have three zones:

$$[z_1, z_2] = [-1, 0], \quad [z_2, z_3] = [0, 0], \quad [z_3, z_4] = [0, 1].$$

15. Toy Example (cont-d)

- *For the first zone* $[z_1, z_2] = [-1, 0]$, according to the above algorithm, we select $x_2 = 0$ and $x_1 = \alpha$, where
$$\frac{1}{2} \cdot (0^2 + \alpha^2) - \frac{1}{4} \cdot (0 + \alpha)^2 = V_0 = 0.16.$$
- Here, $\alpha = -0.8$ and $\alpha = 0.8$, and only the first root belongs to the zone $[-1, 0]$.
- For this root, we compute the value
$$E_1 = \frac{1}{2} \cdot (0 + \alpha) = \frac{1}{2} \cdot (0 + (-0.8)) = -0.4.$$
- *For the second zone* $[z_2, z_3] = [0, 0]$, according to the above algorithm, we select $x_1 = x_2 = 0$.
- In this case, there is no need to compute α, so we directly compute
$$E_2 = \frac{1}{2} \cdot (0 + 0) = 0.$$

16. Toy Example (end)

- *For the third zone* $[z_3, z_4] = [0, 1]$, according to the above algorithm, we select $x_1 = 0$ and $x_2 = \alpha$, where

$$\frac{1}{2} \cdot (0^2 + \alpha^2) - \frac{1}{4} \cdot (0 + \alpha)^2 = V_0 = 0.16.$$

- Of the two roots $\alpha = -0.8$ and $\alpha = 0.8$, only the second root belongs to the zone $[0, 1]$.
- For this root, we compute the value

$$E_3 = \frac{1}{2} \cdot (0 + \alpha) = \frac{1}{2} \cdot (0 + 0.8) = 0.4.$$

- *As a result*, we get the values E_k for all three zones; so, we return

$$\underline{E} = \min(E_1, E_2, E_3) = -0.4;$$

$$\overline{E} = \max(E_1, E_2, E_3) = 0.4.$$

17. Estimating Covariance Range in Privacy Case: Formulation of the Problem

- *Given:*
  - x-thresholds $t^{(x)}_0$, $t^{(x)}_1$, $\ldots$, $t^{(x)}_{N_x}$;
  - y-thresholds $t^{(y)}_0$, $t^{(y)}_1$, $\ldots$, $t^{(y)}_{N_y}$;
  - n pairs of intervals $(\mathbf{x}_i, \mathbf{y}_i)$ in which:
    - each of $\mathbf{x}_i$ is one of the x-ranges $[t^{(x)}_k, t^{(x)}_{k+1}]$, and
    - each of $\mathbf{y}_i$ is one of the y-ranges $[t^{(y)}_\ell, t^{(y)}_{\ell+1}]$.
- *Compute:* the range $[\underline{C}_{x,y}, \overline{C}_{x,y}]$ of possible values of

$$C_{x,y} = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x) \cdot (y_i - E_y) = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y,$$

where

$$E_x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i, \quad E_y = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i.$$

18. Reducing Computing $\overline{C}_{x,y}$ to Computing $\underline{C}_{x,y}$

- We need to compute both the maximum $\overline{C}_{x,y}$ and the minimum $\underline{C}_{x,y}$.
- When we change the sign of y_i, the covariance changes sign as well: $C_{xy}(x_i, -y_i) = -C_{xy}(x_i, y_i)$.
- Thus, for the ranges, we get $\mathbf{C}_{xy}(\mathbf{x}_i, -\mathbf{y}_i) = -\mathbf{C}_{xy}(\mathbf{x}_i, \mathbf{y}_i)$.
- Since the function $z \to -z$ is decreasing:
  - its smallest value is attained when z is the largest;
  - its largest value is attained when z is the smallest.
- Thus, if z goes from $\underline{z}$ to $\overline{z}$, the range of $-z$ is $[-\overline{z}, -\underline{z}]$.
- Therefore, $\underline{C}_{xy}(\mathbf{x}_i, -\mathbf{y}_i) = -\overline{C}_{xy}(\mathbf{x}_i, \mathbf{y}_i)$.
- Thus, if we know how to compute $\underline{C}_{xy}(\mathbf{x}_i, \mathbf{y}_i)$, we can then compute $\overline{C}_{xy}(\mathbf{x}_i, \mathbf{y}_i)$ as $\overline{C}_{xy}(\mathbf{x}_i, \mathbf{y}_i) = -\underline{C}_{xy}(\mathbf{x}_i, -\mathbf{y}_i)$.
- So, we will now only talk about computing $\underline{C}_{x,y}$.

19. Algorithm for Computing $\underline{C}_{xy}$: Main Idea

- We have N_x possible x-ranges $[t_k^{(x)}, t_{k+1}^{(x)}]$.
- We also have N_y possible y-ranges $[t_\ell^{(y)}, t_{\ell+1}^{(y)}]$.
- So, totally, we have $N_x \cdot N_y$ cells $[t_k^{(x)}, t_{k+1}^{(x)}] \times [t_\ell^{(y)}, t_{\ell+1}^{(y)}]$.
- In this algorithm, we analyze these cells c one by one.
- For each c, we assume that the pair (E_x, E_y) corresponding to the minimizing set (x_i, y_i) is contained in c.
- We then find the values (x_i, y_i) where, under this assumption, the minimum of C_{xy} is attained.
- Based on these values x_i and y_i, we compute E_x, E_y.
- If $(E_x, E_y) \in c$, we compute the value C_{xy}.
- The smallest of the corresponding values C_{xy} is the desired minimum $\underline{C}_{xy}$.

20. Possible Position of Intervals $\mathbf{x}_i$ and $\mathbf{y}_i$ in Relation to the Cell

- For each cell $[t_k^{(x)}, t_{k+1}^{(x)}] \times [t_\ell^{(y)}, t_{\ell+1}^{(y)}]$ and for each i, there are three possible positions for $\mathbf{x}_i$:

  X^0: $\mathbf{x}_i$ coincides with the cell's x-range;
  X^-: $\mathbf{x}_i$ is to the left of the x-range;
  X^+: $\mathbf{x}_i$ is to the right of the x-range.

- Similarly, there are three possible positions for $\mathbf{y}_i$:

  Y^0: $\mathbf{y}_i$ coincides with the cell's y-range;
  Y^-: $\mathbf{y}_i$ is below of the y-range;
  Y^+: $\mathbf{y}_i$ is above the y-range.

- So, we have $3 \cdot 3 = 9$ pairs of options.

21. Selecting x_i and y_i at Which C_{xy} Attains its Minimum

For each cell c and for each i, the minimum of $\underline{C}_{xy}$ under the assumption $(E_x, E_y) \in c$ is attained:

- in case (X^+, Y^+): for $x_i = \underline{x}_i$ and $y_i = \underline{y}_i$;
- in case (X^+, Y^0): for $x_i = \overline{x}_i$ and $y_i = \underline{y}_i$;
- in case (X^+, Y^-): for $x_i = \overline{x}_i$ and $y_i = \underline{y}_i$;
- in case (X^-, Y^+): for $x_i = \underline{x}_i$ and $y_i = \overline{y}_i$;
- in case (X^-, Y^0): for $x_i = \underline{x}_i$ and $y_i = \overline{y}_i$;
- in case (X^-, Y^-): for $x_i = \overline{x}_i$ and $y_i = \overline{y}_i$;
- in case (X^0, Y^+): for $x_i = \underline{x}_i$ and $y_i = \overline{y}_i$;
- in case (X^0, Y^-): for $x_i = \overline{x}_i$ and $y_i = \underline{y}_i$;
- in case (X^0, Y^0): for $(x_i, y_i) = (\underline{x}_i, \underline{y}_i)$ or for $(x_i, y_i) = (\overline{x}_i, \overline{y}_i)$.

22. Implementation Details

- For those i for which $\mathbf{x}_i \times \mathbf{y}_i \neq c$, we directly compute the minimizing values x_i and y_i.
- For each i for which $\mathbf{x}_i \times \mathbf{y}_i = c$, we have two different options: $(x_i, y_i) = (\underline{x}_i, \underline{y}_i)$ and $(x_i, y_i) = (\overline{x}_i, \overline{y}_i)$.
- A naive implementation would require testing all 2^M combinations, where M is the number of such cells.
- Luckily, the value C_{xy} does not change if we swap pairs (x_i, y_i).
- So, the value C_{xy} only depends on the number of i's to which we assign $(x_i, y_i) = (\underline{x}_i, \underline{y}_i)$.
- Thus, we can make computations efficient if, for each integer $m = 0, 1, 2, \ldots, M$, we assign:
  - to m i's, the values $x_i = \underline{x}_i$ and $y_i = \underline{y}_i$, and
  - to the rest, the values $x_i = \overline{x}_i$ and $y_i = \overline{y}_i$.

23. Resulting Computation Time of Our Algorithm

- For each cell, we perform $M+1 \le n$ computations C_{xy}, one for each option m.
- In general, computing $E_x = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i$, $E_y = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} y_i$, and $C_{x,y} = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (x_i - E_x) \cdot (y_i - E_y)$ takes time $O(n)$.
- However, each new computation differs from the previous one
  - by a single change in $\sum x_i \cdot y_i$ and
  - a single change in estimating $E_x \sim \sum x_i$ and $E_y \sim \sum y_i$.
- Thus, each new computation requires $O(1)$, and so, for each cell, the total computation time is $O(n)$.
- So, for all $N_x \cdot N_y$ cells, we need time $O(N_x \cdot N_y \cdot n)$.

24. Computation Time: Discussion

- *Reminder:* this algorithm takes time $O(N_x \cdot N_y \cdot n)$.
- Usually, the number N_x of x-ranges and the number N_y of y-ranges are fixed.
- In this case, what we have is a *linear-time* algorithm.
- Clearly, it is not possible to compute covariance faster than in linear time:
  - we need to take into account all n pairs $(\mathbf{x}_i, \mathbf{y}_i)$, and
  - processing each data point requires at least one computation.
- So, our algorithm is *(asymptotically) optimal* – it requires the smallest possible order of computation time $O(n)$.
- *Comment:* for general (non-privacy) intervals, the problem is NP-hard.

25. Computing $\overline{C}_{xy}$: A Reminder

- We use the fact that $\overline{C}_{xy} = -\underline{C}_{xz}$ where $z = -y$.
- We form N_y threshold values for z:

$$t_0^{(z)} = -t_{N_y}^{(y)}, t_1^{(z)} = -t_{N_y-1}^{(y)}, \ldots, t_{N_y}^{(z)} = -t_0^{(y)}.$$

- We then form N_y z-ranges:

$$[t_0^{(z)}, t_1^{(z)}], [t_1^{(z)}, t_2^{(z)}], \ldots, [t_{N_y-1}^{(z)}, t_{N_y}^{(z)}].$$

- Based on the intervals $\mathbf{y}_i = [\underline{y}_i, \overline{y}_i]$, we form intervals $\mathbf{z}_i = -\mathbf{y}_i = [-\overline{y}_i, -\underline{y}_i]$.
- We apply the above algorithm for computing the lower bound to compute the value $\underline{C}_{xz}$.
- Finally, we compute $\overline{C}_{xy}$ as $\overline{C}_{xy} = -\underline{C}_{xz}$.

26. Estimating Correlation: Main Result

- There exists a polynomial-time algorithm that:
  - given n pairs of intervals $[\underline{x}_i, \overline{x}_i]$ and $[\underline{y}_i, \overline{y}_i]$,
  - computes (at least) one of the endpoint of the interval $[\underline{\rho}, \overline{\rho}]$ of possible values of the correlation ρ.
- Specifically, in the case of a non-degenerate interval $[\underline{\rho}, \overline{\rho}]$:
  - when $\overline{\rho} \le 0$, we compute the lower endpoint $\underline{\rho}$;
  - when $0 \le \underline{\rho}$, we compute the upper endpoint $\overline{\rho}$;
  - in all remaining cases, we compute both endpoints $\underline{\rho}$ and $\overline{\rho}$.

27. Reducing Minimum to Maximum

- When we change the sign of y_i, the correlation changes sign as well:

$$\rho(x_1,\ldots,x_n,-y_1,\ldots,-y_n)=-\rho(x_1,\ldots,x_n,y_1,\ldots,y_n).$$

- If z goes from $\underline{z}$ to $\overline{z}$, the range of $-z$ is $[-\overline{z},-\underline{z}]$.
- So, for the endpoints of the ranges, we get

$$\overline{\rho}([\underline{x}_1,\overline{x}_1],\ldots,[\underline{x}_n,\overline{x}_n],-[\underline{y}_1,\overline{y}_1],\ldots,-[\underline{y}_n,\overline{y}_n])=$$

$$-\underline{\rho}([\underline{x}_1,\overline{x}_1],\ldots,[\underline{x}_n,\overline{x}_n],[\underline{y}_1,\overline{y}_1],\ldots,[\underline{y}_n,\overline{y}_n]),$$

where $-[\underline{y}_i,\overline{y}_i]=\{-y_i : y_i\in[\underline{y}_i,\overline{y}_i]\}=[-\overline{y}_i,-\underline{y}_i]$.

- If we know how to compute $\overline{\rho}$, we can compute $\underline{\rho}$ as

$$\underline{\rho}([\underline{x}_1,\overline{x}_1],\ldots,[\underline{x}_n,\overline{x}_n],[\underline{y}_1,\overline{y}_1],\ldots,[\underline{y}_n,\overline{y}_n])=$$

$$-\overline{\rho}([\underline{x}_1,\overline{x}_1],\ldots,[\underline{x}_n,\overline{x}_n],[-\overline{y}_1,-\underline{y}_1],\ldots,[-\overline{y}_n,-\underline{y}_n]).$$

- Thus, we can concentrate on computing $\overline{\rho}$.

28. Algorithm

- For each i from 1 to n, the box $[\underline{x}_i, \overline{x}_i] \times [\underline{y}_i, \overline{y}_i]$ has four vertices: $(\underline{x}_i, \underline{y}_i)$, $(\underline{x}_i, \overline{y}_i)$, $(\overline{x}_i, \underline{y}_i)$, and $(\overline{x}_i, \overline{y}_i)$.
- Let's consider 4-tuples consisting of two vertices and two signs $(-,-)$, $(-,0)$, $\ldots$, $(+,+)$.
- For the first vertex, we:
  - slightly increase x if the first sign is $+$ and
  - slightly decrease x if the first sign is $-$.
- We similarly move the second vertex depending on the second sign.
- We form a straight line through the resulting points.
- We select two 4-tuples, and form two lines: *representative x-line* and *representative y-line*.

29. Algorithm (cont-d)

- We have an actual x-line $y = E_y + k_x \cdot (x - E_x)$ and an actual y-line $x = E_x + k_y \cdot (y - E_y)$.
- Here, E_x, E_y, k_x, k_y are to-be-determined.
- For each box, based on its location in comparison to the representative lines, we select the values x_i and y_i:
- If the box is above the repr. x-line, take $x_i = \overline{x}_i$; then, select y_i s.t. $(\overline{x}_i, y_i)$ is the closest to the actual y-line.
- If the box is below the x-line, we take $x_i = \underline{x}_i$.
- If the box is to the right of the y-line, take $y_i = \underline{y}_i$; select x_i s.t. $(x_i, \underline{y}_i)$ is the closest to the actual x-line.
- If the box is to the left of the repr. y-line, take $y_i = \overline{y}_i$.
- When the box contains the intersection point (E_x, E_y) of x- and y-lines, take $x_i = E_x$ and $y_i = E_y$.

30. Algorithm (cont-d)

- For each i, we get explicit expressions for x_i and y_i in terms of the four unknowns E_x, E_y, k_x and k_y.
- By substituting these expressions into the following formulas, we get a system of 4 equations with 4 unknowns:

$$E_x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i; \quad E_y = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i;$$

$$\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y = k_x \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x)^2 \right);$$

$$\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y = k_y \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - E_y)^2 \right).$$

- For each of the solutions E_x, E_y, k_x and k_y, we compute x_i and y_i ($i = 1, \ldots, n$), and then the correlation ρ.
- The largest of these values ρ is returned as $\overline{\rho}$.

31. Computation Time

- We have $4n$ possible vertices, so we have $O(n^2)$ possible pairs of vertices – and thus, $O(n^2)$ possible 4-tuples.
- Thus, we have $O(n^2)$ possible representative x-lines, and we also have $O(n^2)$ representative y-lines.
- In our algorithms, we consider pairs consisting of a representative x-line and a representative y-line.
- We have $O(n^2) \cdot O(n^2) = O(n^4)$ possible pairs of lines.
- For each pair of lines, we need:
  - $O(n)$ steps to select x_i and y_i for each of n boxes;
  - $O(n)$ steps to compute ρ;
  - to the total of $O(n) + O(n) = O(n)$.
- Thus, the total computation time is $O(n^4) \times O(n) = O(n^5)$, which is polynomial (feasible).

32. Proof of the First Result: Main Lemmas

- For $x'_i = -x_i$, we have $E' = -E$ and $V' = V$.
- Thus $\underline{E} = -\overline{E'}$; so, it is sufficient to consider $\overline{E}$.
- Let x be an optimizing vector, i.e., $E(x) = \overline{E}$.
- *Lemma 1:* if $x_i < E$, then $x_i = \overline{x}_i$.
- *Proof:* else, by adding $\Delta x_i > 0$ to x_i, we could increase E without increasing V.
- *Lemma 2:* if $\underline{x}_i < x_i < \overline{x}_i$, then:
  - for every j for which $E \le x_j < x_i$, we have $x_j = \overline{x}_j$;
  - for every k for which $x_k > x_i$, we have $x_k = \underline{x}_k$.
- *Proof:* similar.
- *Lemma 3:* if for all $x_i \ge E$, we have either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$, then $x_i = \overline{x}_i$ and $x_j = \underline{x}_j$ imply $x_i \le x_j$.

33. Proof of the First Result (cont-d)

- *Lemma 1:* if $x_i < E$, then $x_i = \overline{x}_i$.
- *Lemma 2:* if $\underline{x}_i < x_i < \overline{x}_i$, then:
  - for every j for which $E \le x_j < x_i$, we have $x_j = \overline{x}_j$;
  - for every k for which $x_k > x_i$, we have $x_k = \underline{x}_k$.
- *Lemma 3:* if for all $x_i \ge E$, we have either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$, then $x_i = \overline{x}_i$ and $x_j = \underline{x}_j$ imply $x_i \le x_j$.
- Thus, there exists a threshold value α such that
  - for all j for which $x_j < \alpha$, we have $x_j = \overline{x}_j$;
  - for all k for which $x_k > \alpha$, we have $x_k = \underline{x}_k$.
- Once we know to which zone α belongs, we can uniquely determine all x_j of the corresponding vector x.
- Then $\overline{E}$ is the largest of the values $E(x)$ corresponding to different zones.

34. Toward Justification of our Second Algorithm: Known Facts from Calculus

- A function $f(x)$ defined on an interval $[\underline{x}, \overline{x}]$ attains its minimum:
  - either an internal point $x \in (\underline{x}, \overline{x})$,
  - or at one of its endpoints $x = \underline{x}$ or $x = \overline{x}$.
- If the minimum of $f(x)$ is attained at an internal point, then
$$\frac{df}{dx} = 0.$$
- If the minimum is attained for $x = \underline{x}$, then
$$\frac{df}{dx} \geq 0.$$
- If the minimum is attained for $x = \overline{x}$, then
$$\frac{df}{dx} \leq 0.$$

35. Let Us Apply These Facts to Our Problem

- In general, for the point $(x_1, \ldots, x_n)$ at which a function $f(x_1, \ldots, x_n)$ attains its minimum, we have:
  - if $x_i = \underline{x}_i$, then $\dfrac{\partial f}{\partial x_i} \geq 0$;
  - if $x_i = \overline{x}_i$, then $\dfrac{\partial f}{\partial x_i} \leq 0$;
  - if $\underline{x}_i < x_i < \overline{x}_i$, then $\dfrac{\partial f}{\partial x_i} = 0$.
- For covariance C_{xy}, we have $\dfrac{\partial C_{xy}}{\partial x_i} = \dfrac{1}{n} \cdot (y_i - E_y)$.
- Thus, for the point $(x_1, \ldots, x_n, y_1, \ldots, y_n)$ at which C_{xy} attains its minimum, we have:
  - if $x_i = \underline{x}_i$, then $y_i \geq E_y$.
  - if $x_i = \overline{x}_i$, then $y_i \leq E_y$.
  - if $\underline{x}_i < x_i < \overline{x}_i$, then $y_i = E_y$.

36. Case of $\overline{y}_i < E_y$

- *Case:* $\overline{y}_i < E_y$.
- *Reminder:*
  - if $x_i = \underline{x}_i$, then $y_i \ge E_y$.
  - if $x_i = \overline{x}_i$, then $y_i \le E_y$.
  - if $\underline{x}_i < x_i < \overline{x}_i$, then $y_i = E_y$.
- Since $\overline{y}_i < E_y$ and $y_i \le \overline{y}_i$, we have $y_i < E_y$.
- Thus, in this case:
  - we cannot have $x_i = \underline{x}_i$, because then we would have $y_i \ge E_y$
  - we cannot have $\underline{x}_i < x_i < \overline{x}_i$, because then we would have $y_i = E_y$.
- So, if $\overline{y}_i < E_y$, the only remaining option is $x_i = \overline{x}_i$.

37. Case of $E_y < \underline{y}_i$

- *Case:* $E_y < \underline{y}_i$.
- *Reminder:*
  - if $x_i = \underline{x}_i$, then $y_i \geq E_y$.
  - if $x_i = \overline{x}_i$, then $y_i \leq E_y$.
  - if $\underline{x}_i < x_i < \overline{x}_i$, then $y_i = E_y$.
- Since $E_y < \underline{y}_i$ and $\underline{y}_i \leq y_i$, we have $E_y < y_i$.
- Thus, in this case:
  - we cannot have $x_i = \overline{x}_i$, because then we would have $y_i \leq E_y$
  - we cannot have $\underline{x}_i < x_i < \overline{x}_i$, because then we would have $y_i = E_y$.
- So, if $E_y < \underline{y}_i$, the only remaining option is $x_i = \underline{x}_i$.

38. Cases of $\overline{x}_i < E_x$ and $E_x < \underline{x}_i$

- We have shown that:
  - if $\overline{y}_i < E_y$, then $x_i = \overline{x}_i$;
  - if $E_y < \underline{y}_i$, then $x_i = \underline{x}_i$.
- We can similarly conclude that:
  - if $\overline{x}_i < E_x$, then $y_i = \overline{y}_i$;
  - if $E_x < \underline{x}_i$, then $y_i = \underline{y}_i$.
- So, we can tell exactly where the min is attained if:
  - the interval $\mathbf{x}_i$ is either completely to the left or to the right of E_x, and
  - the interval $\mathbf{y}_i$ is either completely to the left or to the right of E_y,
- E.g., if $\overline{x}_i < E_x$ ($\mathbf{x}_i$ to the left of E_x) and $E_y < \underline{y}_i$ ($\mathbf{y}_i$ to the right), then min is attained for $x_i = \underline{x}_i$ and $y_i = \overline{y}_i$.

39. Case When One of the Intervals Contains E_x or E_y Inside

- What if one of the intervals, e.g., $\mathbf{x}_i$, is fully to the left or fully to the right of E_x, but $\mathbf{y}_i$ contains E_y inside?
- For example, if $\overline{x}_i < E_x$, this means that $y_i = \overline{y}_i$.
- Since E_y in inside the interval $[\underline{y}_i, \overline{y}_i]$, this means that $\underline{y}_i \le E_y \le \overline{y}_i$ and thus, $E_y \le y_i$.
- If $E_y < y_i$, then, as we have shown earlier, we get $x_i = \underline{x}_i$.
- One can show that the same conclusion holds when $y_i = E_y$.
- So, in this case, we also have a single pair (x_i, y_i) where the minimum can be attained: $x_i = \underline{x}_i$ and $y_i = \overline{y}_i$.

40. Case When $(E_x, E_y) \in c$

- Where is the point (x_i, y_i) at which the minimum is attained?
- Calculus shows that (x_i, y_i) is in the union U_1 of the following three linear segments:
  - a segment where $x_i = \underline{x}_i$ and $y_i \geq E_y$;
  - a segment where $x_i = \overline{x}_i$ and $y_i \leq E_y$; and
  - a segment where $\underline{x}_i < x_i < \overline{x}_i$ and $y_i = E_y$.
- Similarly, (x_i, y_i) is in the union U_2 of the following three linear segments:
  - a segment where $y_i = \underline{y}_i$ and $x_i \geq E_x$;
  - a segment where $y_i = \overline{y}_i$ and $x_i \leq E_x$; and
  - a segment where $\underline{y}_i < y_i < \overline{y}_i$ and $x_i = E_x$.
- So, $(x_i, y_i) \in U_1 \cap U_2 = \{(\underline{x}_i, \underline{y}_i), (\overline{x}_i, \overline{y}_i), (E_x, E_y)\}$.

41. Case when $(E_x, E_y) \in c$ (cont-d)

- We showed that in this case, the minimum of C_{xy} is attained at $(\underline{x}_i, \underline{y}_i)$, $(\overline{x}_i, \overline{y}_i)$, or at (E_x, E_y).
- Let us show that it cannot be attained at (E_x, E_y).
- Indeed, let us then take a small Δ and replace $x_i = E_x$ with $x_i + \Delta$ and $y_i = E_y$ with $y_i - \Delta$. Then:

$$E'_x = E_x + \frac{\Delta}{n},\ \ E'_y = E_y - \frac{\Delta}{n},\ \ C'_{xy} = C_{xy} - \frac{\Delta^2}{n} \cdot \left(1 - \frac{1}{n}\right).$$

- These equalities are easy to prove if we shift all the values of x_j by $-E_x$ and all the values of y_j by $-E_y$.
- Indeed, such a shift does not change C_{xy}.
- The new value C'_{xy} is smaller than C_{xy}, while we assumed that C_{xy} is minimal: a contradiction.
- Thus, in the case when $(E_x, E_y) \in c$, the minimum can be only attained at $(\underline{x}_i, \underline{y}_i)$ or $(\overline{x}_i, \overline{y}_i)$.

42. Proof of Correctness: Final Step

- We know that for minimizing vector $(x_1, \ldots, x_n, y_1, \ldots, y_n)$, the pair (E_x, E_y) must be contained in one of the $N_x \cdot N_y$ cells.
- We have already shown that for each cell:
  - if the pair (E_x, E_y) is contained in this cell,
  - then the corresponding minimizing values x_i and y_i will be as above.
- Thus, the actual minimizing value will be obtained when we analyze the corresponding cell.
- So, the desired value $\underline{C}_{xy}$ will be among the values computed by the above algorithm.
- Thus, the smallest of the computed values will be exactly $\underline{C}_{xy}$.

43. Towards Proving the Third Result: Reminder

- A function $f(x)$ defined on an interval $[\underline{x}, \overline{x}]$ attains its minimum:
  - either an internal point $x \in (\underline{x}, \overline{x})$,
  - or at one of its endpoints $x = \underline{x}$ or $x = \overline{x}$.
- If the minimum of $f(x)$ is attained at an internal point, then
$$\frac{df}{dx} = 0.$$
- If the minimum is attained for $x = \underline{x}$, then
$$\frac{df}{dx} \geq 0.$$
- If the minimum is attained for $x = \overline{x}$, then
$$\frac{df}{dx} \leq 0.$$

44. Proof of the Third Result

- $\dfrac{\partial \rho}{\partial x_i} = \dfrac{1}{\sigma_x \cdot \sigma_y \cdot n} \cdot [(y_i - E_y) - k_x \cdot (x_i - E_x)]$, w/ $k_x = \dfrac{C}{V_x}$.
- Thus, the sign of the derivative coincides with the sign of the expression $(y_i - E_y) - k_x \cdot (x_i - E_x)$.
- So, the sign depends on whether we are above or below the actual x-line $y_i = E_y + k_x \cdot (x_i - E_x)$.
- The sign of $\dfrac{\partial \rho}{\partial y_i}$ depends on where we are w.r.t. the actual y-line $x_i = E_x + k_y \cdot (y_i - E_y)$, with $k_y = \dfrac{C}{V_y}$.
- Now, the selection of x_i and y_i follows from calculus.
- All possible locations of lines w.r.t. vertices are covered:
  - each line can be moved and rotated
  - until it almost touches two points – i.e., becomes one of our representative lines.

45. Acknowledgments

I want to sincerely thank everyone who helped me:

- members of my committees, Drs. Vladik Kreinovich, Luc Longpré, and Peter Moscoupoulos;
- all other faculty and staff from UTEP Computer Science Department, especially Dr. Eric Freudenthal;
- all my friends here and in Iran;
- last but not the least, my amazing family for their non-stop love and support all through my life.

There are no words to fully express all my feelings, I can only say THANK YOU to everyone!