Weighted Model Integration
with Orthogonal Transformations

David Merrell, Aws Albarghouthi, Loris D'Antoni

Department of Computer Sciences

University of Wisconsin – Madison

dmerrell@cs.wisc.edu

Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

- **Inference method** for probabilistic models;
  - e.g., Bayesian networks.
  - More generally: **probabilistic programs**.
  - Restrict ourselves to linear real arithmetic.

```
x = Normal(0,9);
y = Normal(0,9);
z = x + y;

Pr{z < 1}
```

Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

- **Inference method** for probabilistic models;
  - e.g., Bayesian networks.
  - More generally: **probabilistic programs**.
  - Restrict ourselves to linear real arithmetic.

```
x = Normal(0,9);
y = Normal(0,9);
z = x + y;

Pr{z < 1}
```

$$\varphi \equiv \begin{array}{c} \exists z.\ z = x + y \\ \wedge\ z < 1 \end{array}$$

$$x, y \sim N(0, 9)$$

Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

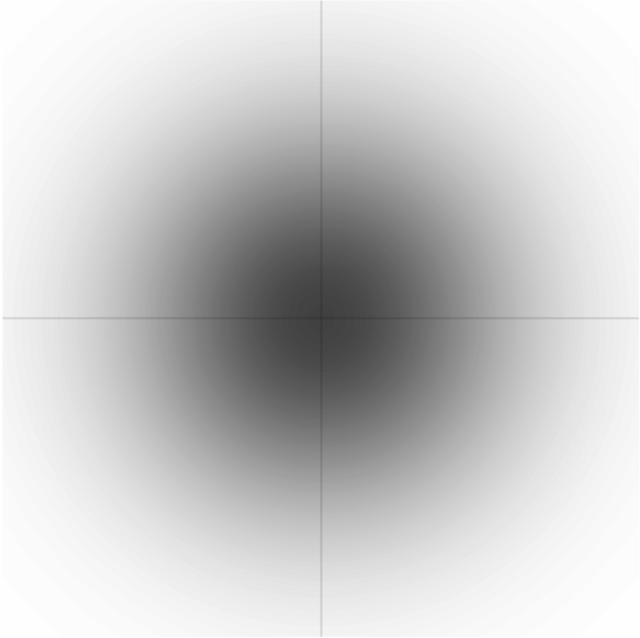
- **Inference method** for probabilistic models;
  - e.g., Bayesian networks.
  - More generally: **probabilistic programs**.
  - Restrict ourselves to linear real arithmetic.

```
x = Normal(0,9);
y = Normal(0,9);
z = x + y;

Pr{z < 1}
```

$$\varphi \equiv \begin{array}{c} \exists z.\ z = x + y \\ \wedge\ z < 1 \end{array}$$

$$x, y \sim N(0, 9)$$



Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

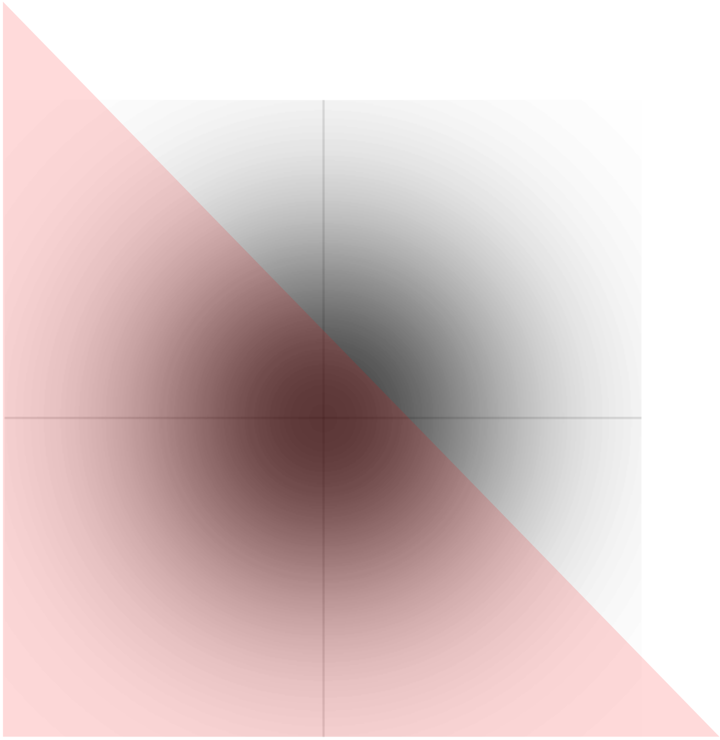
- **Inference method** for probabilistic models;
  - e.g., Bayesian networks.
  - More generally: **probabilistic programs**.
  - Restrict ourselves to linear real arithmetic.

```
x = Normal(0,9);
y = Normal(0,9);
z = x + y;

Pr{z < 1}
```

$$\varphi \equiv \begin{array}{c} \exists z.\ z = x + y \\ \wedge\ z < 1 \end{array}$$

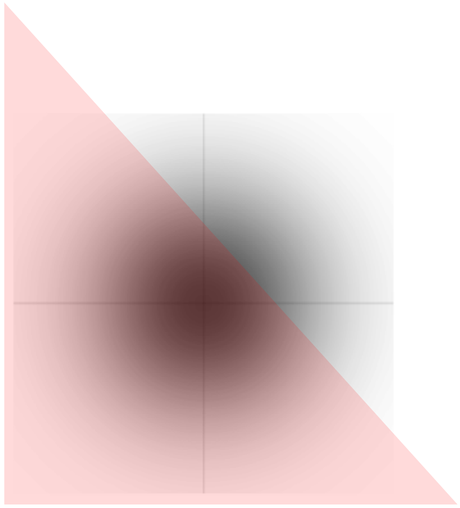
$$x, y \sim N(0, 9)$$



Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

- Rectangular Decomposition Methods
  [Sankaranarayanan 2013; Albarghouthi 2017,2016]

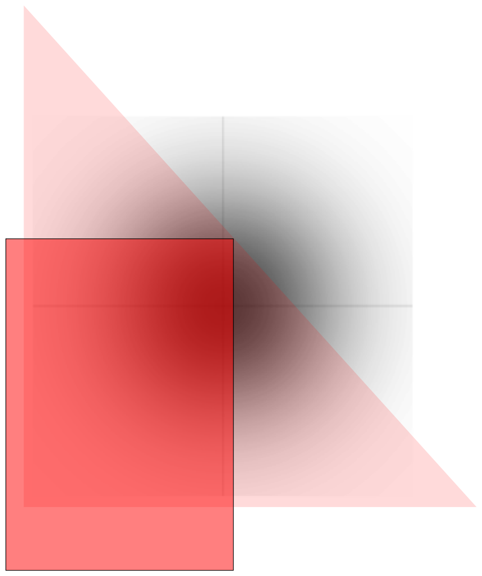


Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

- Rectangular Decomposition Methods
  [Sankaranarayanan 2013; Albarghouthi 2017,2016]

$$H_\varphi \equiv \left(\bigwedge_{x \in \mathcal{X}_\varphi} l_x \leq u_x \right) \wedge \forall \mathcal{X}_\varphi . \left(\bigwedge_{x \in \mathcal{X}_\varphi} l_x \leq x \leq u_x \right) \Rightarrow \varphi$$

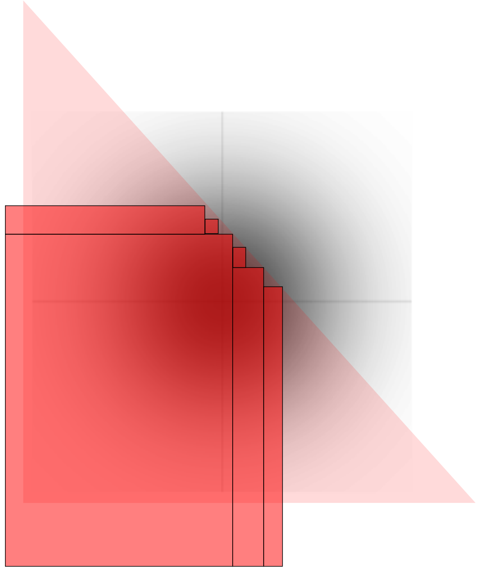


Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

- Rectangular Decomposition Methods
  [Sankaranarayanan 2013; Albarghouthi 2017,2016]

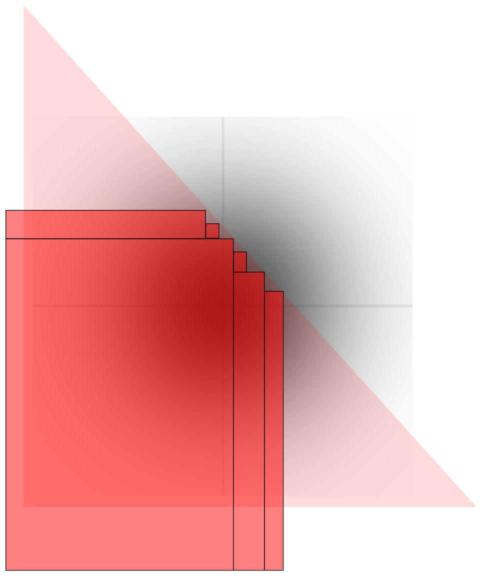
$$H_\varphi \equiv \Big(\bigwedge_{x\in\mathcal{X}_\varphi} l_x \leq u_x\Big) \wedge \forall \mathcal{X}_\varphi . \Big(\bigwedge_{x\in\mathcal{X}_\varphi} l_x \leq x \leq u_x\Big) \Rightarrow \varphi$$



Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

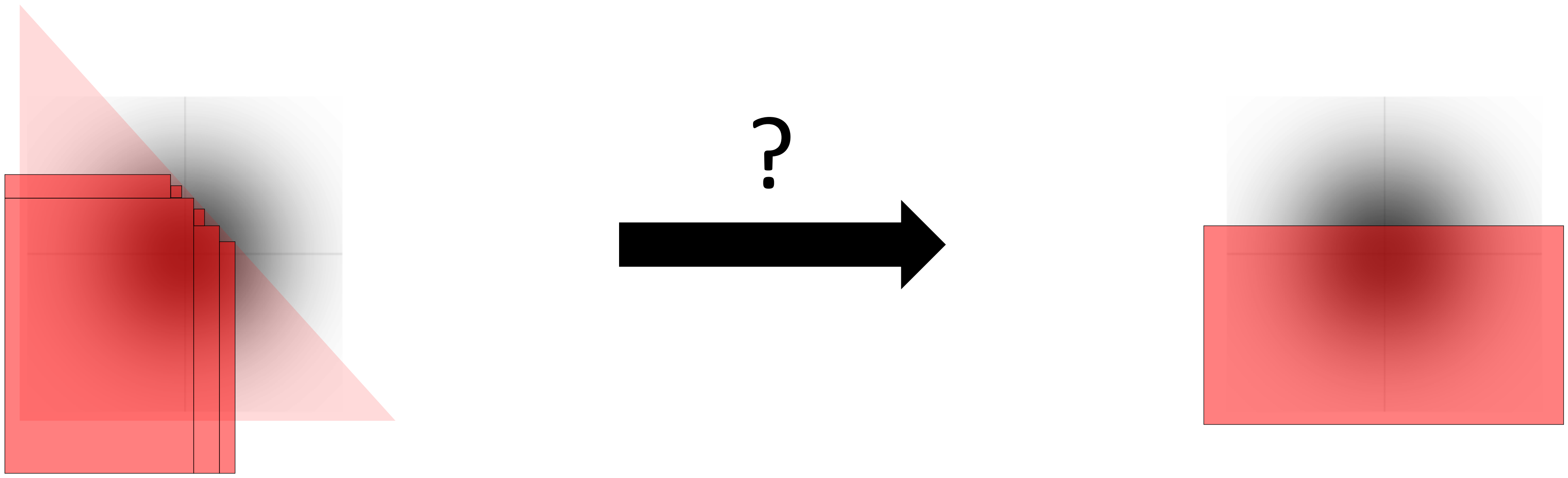
- Rectangular Decomposition Methods
  [Sankaranarayanan 2013; Albarghouthi 2017,2016]
- Problem: **inefficient sampling.**



Background: Weighted Model Integration

[Belle et al 2015, 2016; Chistikov et al 2015, 2014]

- Rectangular Decomposition Methods
  [Sankaranarayanan 2013; Albarghouthi 2017,2016]
- Problem: **inefficient sampling.**
- Can we improve efficiency by **rotating** the formula**?**



Using Rotation Matrices

Using Rotation Matrices

- How do transformations **affect volume?**

Using Rotation Matrices

- **How do transformations affect volume?**

- Integration by Substitution of Variables:

$$\int_{T\varphi} f(\mathcal{X})d\mathcal{X} = \int_{\varphi} f(T\mathcal{X})|\det T|d\mathcal{X}$$

Using Rotation Matrices

- How do transformations **affect volume?**

- Integration by Substitution of Variables:

$$\int_{T\varphi} f(\mathcal{X})d\mathcal{X} = \int_{\varphi} f(T\mathcal{X})|\det T|d\mathcal{X}$$

Region of integration transformed by matrix T

Using Rotation Matrices

- How do transformations **affect volume?**

- Integration by Substitution of Variables:

$$\int_{T\varphi} f(\mathcal{X})d\mathcal{X} = \int_{\varphi} f(T\mathcal{X})|\det T|d\mathcal{X}$$

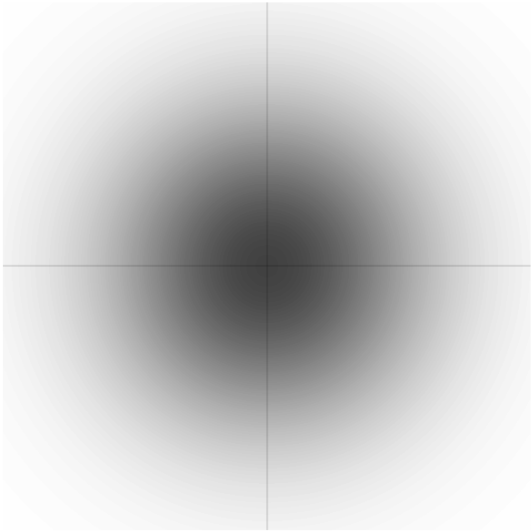
Region of integration transformed by matrix T

Determinant is 1
for orthogonal transformations.

Using Rotation Matrices

- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)

Standard normal joint PDF

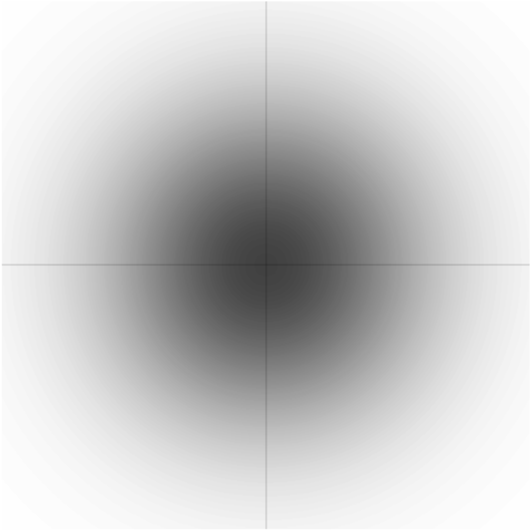


Using Rotation Matrices

- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)

Standard normal joint PDF

$$f(T\vec{x}) = f(\vec{x})$$

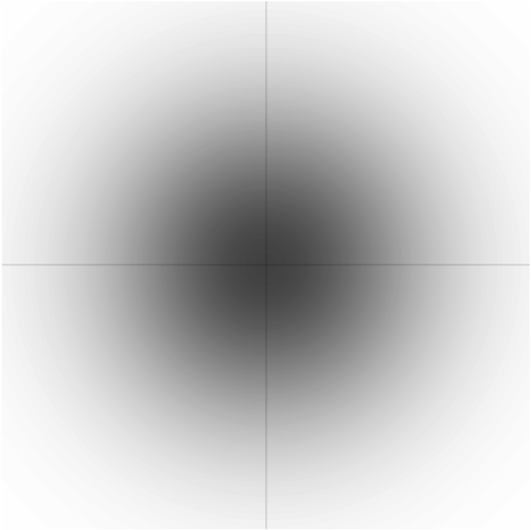


Using Rotation Matrices

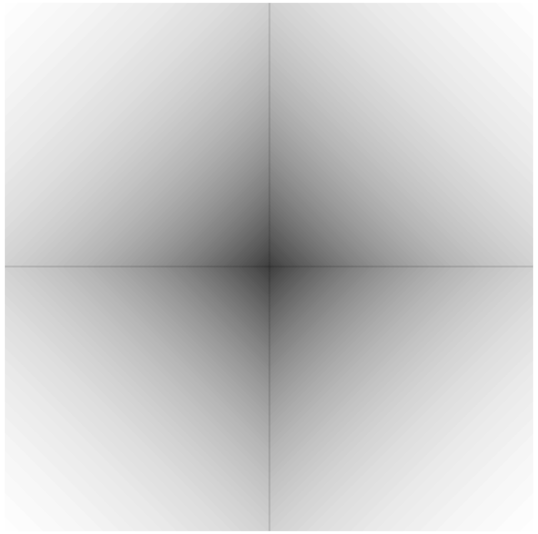
- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)

Standard normal joint PDF

$$f(T\vec{x}) = f(\vec{x})$$



Laplace joint PDF

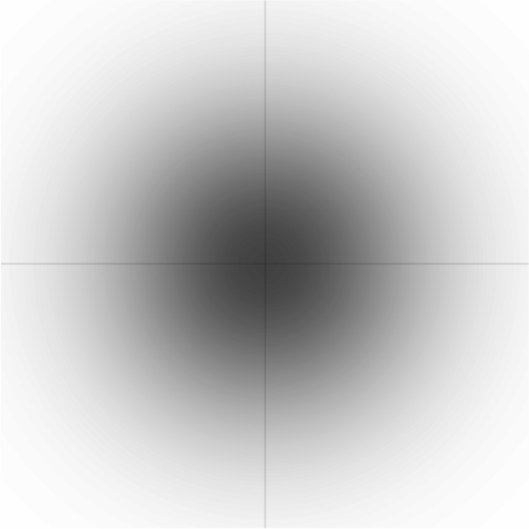


Using Rotation Matrices

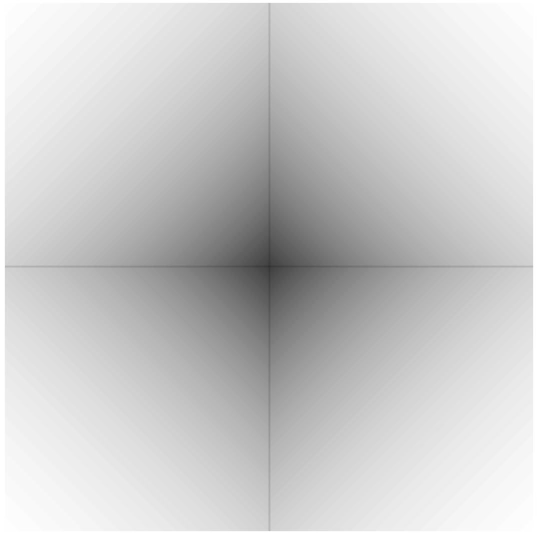
- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)

Standard normal joint PDF

$$f(T\vec{x}) = f(\vec{x})$$

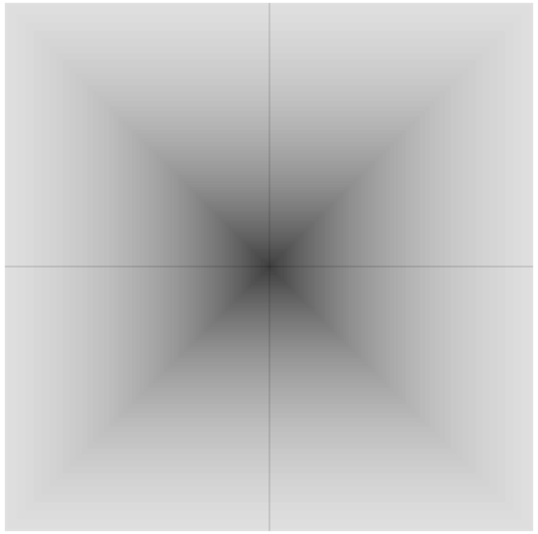


Laplace joint PDF



NOT a product distribution.

Rotations do **NOT** preserve independence.



Using Rotation Matrices

- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)
- The Skitovitch-Darmois Theorem:

Using Rotation Matrices

- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)
- The Skitovitch-Darmois Theorem:

$$X_i \text{ independent} \qquad \alpha_i, \beta_i \neq 0 \qquad i \in \{1 \ldots k\}$$

Using Rotation Matrices

- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)
- The Skitovitch-Darmois Theorem:

$$X_i \text{ independent} \qquad \alpha_i, \beta_i \neq 0 \qquad i \in \{1 \ldots k\}$$

$$\left.\begin{aligned} L_1 &= \sum_{i=1}^{k} \alpha_i X_i \\ L_2 &= \sum_{i=1}^{k} \beta_i X_i \end{aligned}\right\} \text{independent}$$

Using Rotation Matrices

- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)
- The Skitovitch-Darmois Theorem:

$$X_i \text{ independent} \qquad \alpha_i, \beta_i \neq 0 \qquad i \in \{1 \ldots k\}$$

$$\left.\begin{aligned} L_1 &= \sum_{i=1}^{k} \alpha_i X_i \\ L_2 &= \sum_{i=1}^{k} \beta_i X_i \end{aligned}\right\} \text{independent} \quad \Longrightarrow \quad X_i \sim N(\mu_i, \sigma_i^2)$$

Using Rotation Matrices

- When does a transformation **preserve stochastic independence?**
  (Rectangular decomposition requires independent random variables.)
- The Skitovitch-Darmois Theorem:

$$X_i \text{ independent} \qquad \alpha_i, \beta_i \neq 0 \qquad i \in \{1 \dots k\}$$

Contrapositive

$$\left.\begin{aligned} L_1 &= \sum_{i=1}^{k} \alpha_i X_i \\ L_2 &= \sum_{i=1}^{k} \beta_i X_i \end{aligned}\right\} \not\text{independent} \quad \Longleftarrow \quad X_i \not\sim N(\mu_i, \sigma_i^2)$$

Bottom line: Rectangular Decomposition can only use linear transformations on **normally distributed variables**.

Using Rotation Matrices

- A little bit of wiggle room:
  - We can restrict rotations to the “Gaussian subspace”
  - We can still handle *any* multivariate normal distribution
    - (shift and scale to standard form)

Using Rotation Matrices

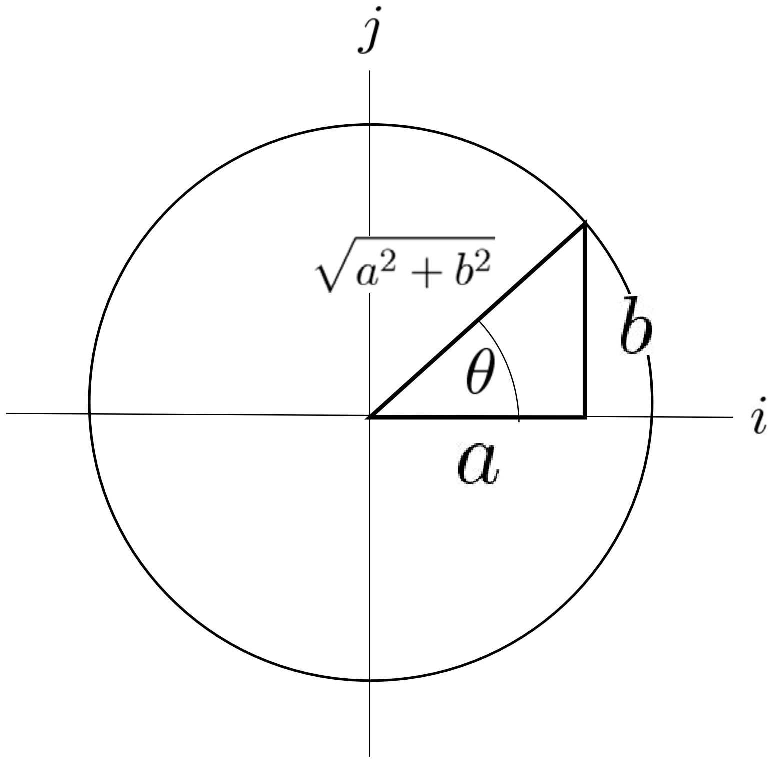
- How do we make sure the transformed formula has **rational coefficients?**
- Rational **Givens Rotations**:

$$G(i,j) = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & -b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & b & \cdots & a & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

Using Rotation Matrices

- How do we make sure the transformed formula has **rational coefficients?**
- Rational **Givens Rotations**:

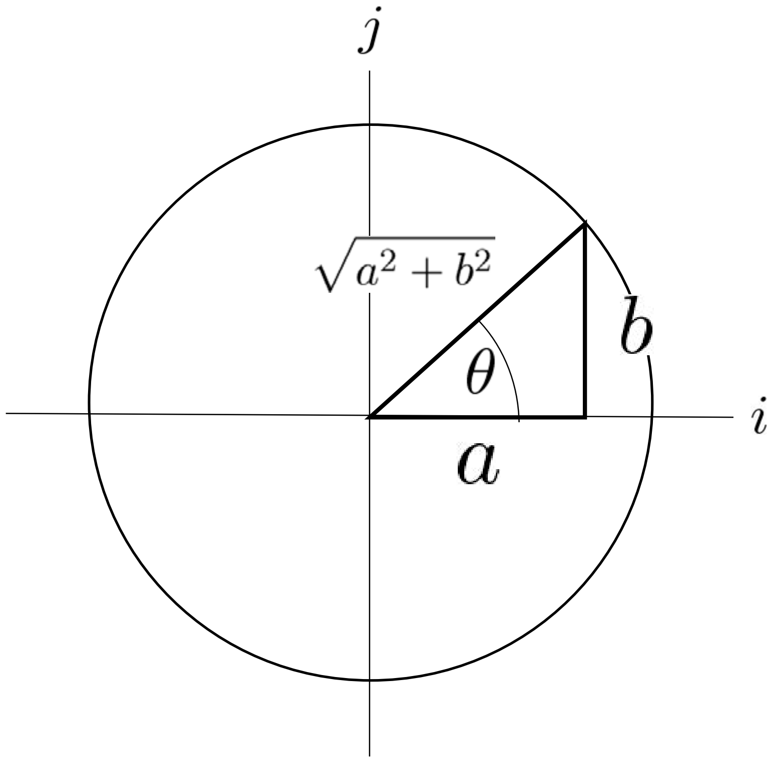
$$G(i,j) = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & -b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & b & \cdots & a & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$



Using Rotation Matrices

- How do we make sure the transformed formula has **rational coefficients?**
- Rational **Givens Rotations**:

$$G(i,j) = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & -b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & b & \cdots & a & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$



Pythagorean Triples

$$a, b, \sqrt{a^2+b^2} \in \mathbb{Z}$$

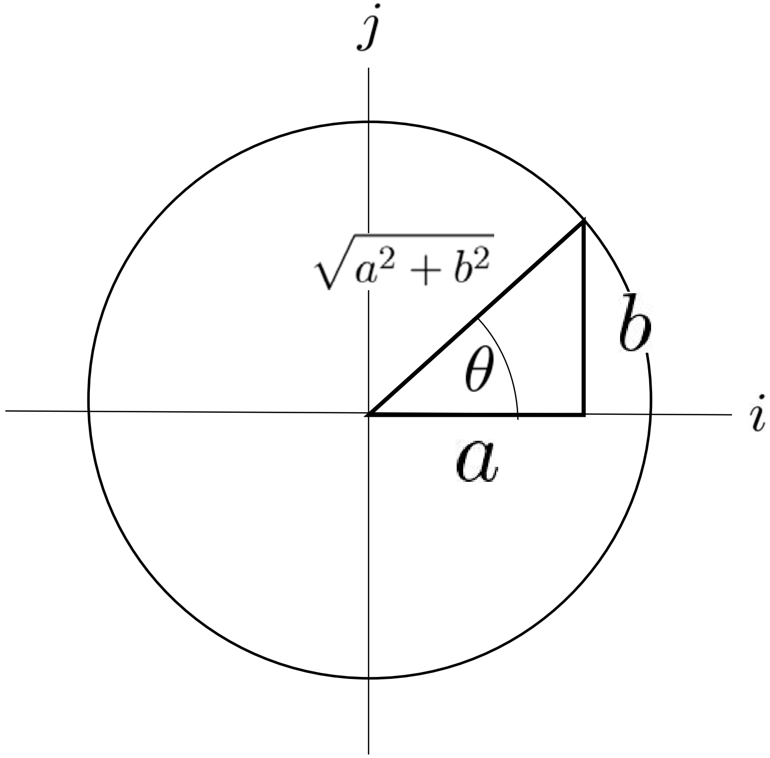
$$\Downarrow$$

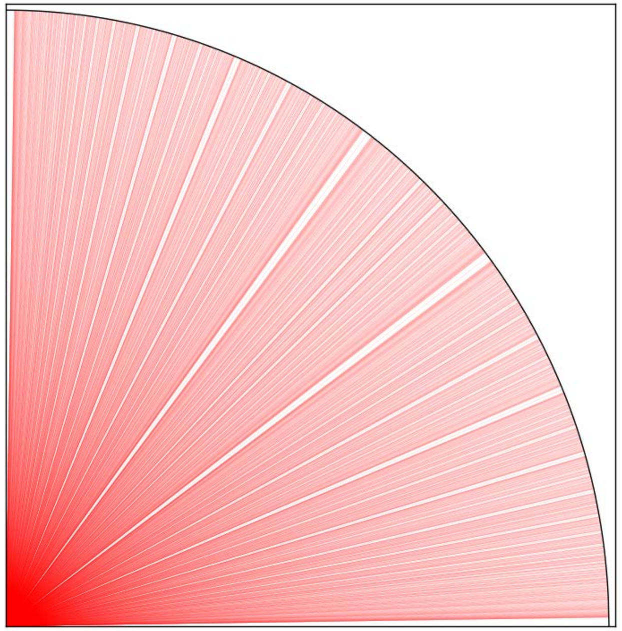
$$G(i,j) \in \mathbb{Q}^{k \times k}$$

Using Rotation Matrices

- How do we make sure the transformed formula has **rational coefficients?**
- Rational **Givens Rotations**:

$$G(i,j) = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & -b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & b & \cdots & a & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

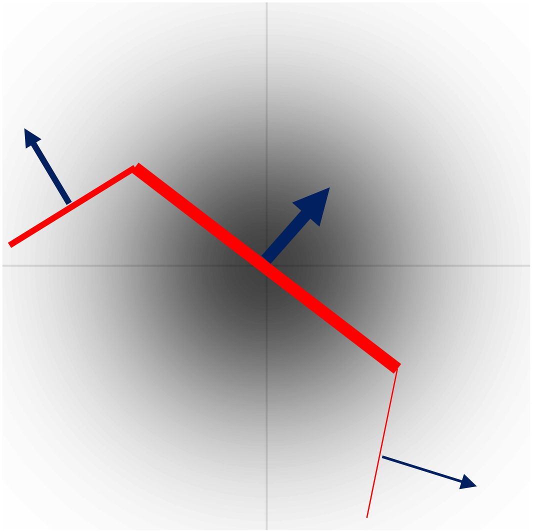




[Shiu 83]:
algorithm for obtaining **integers** *a* and *b* that approximate cosθ and sinθ

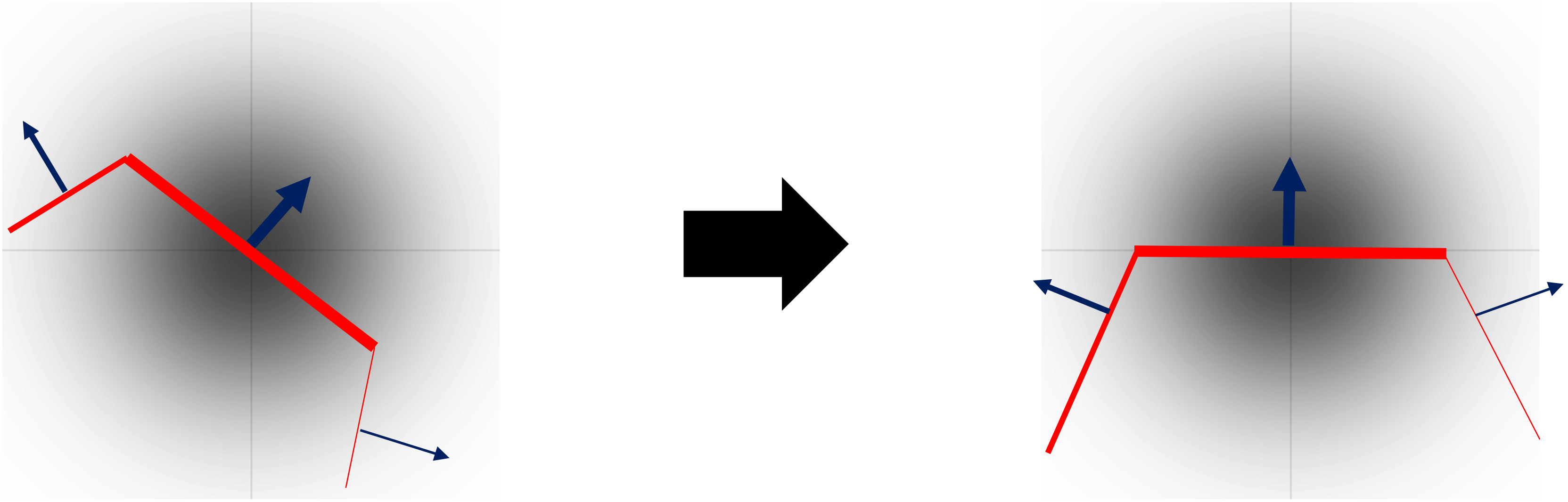
Using Rotation Matrices

- How do we **choose a good rotation?**



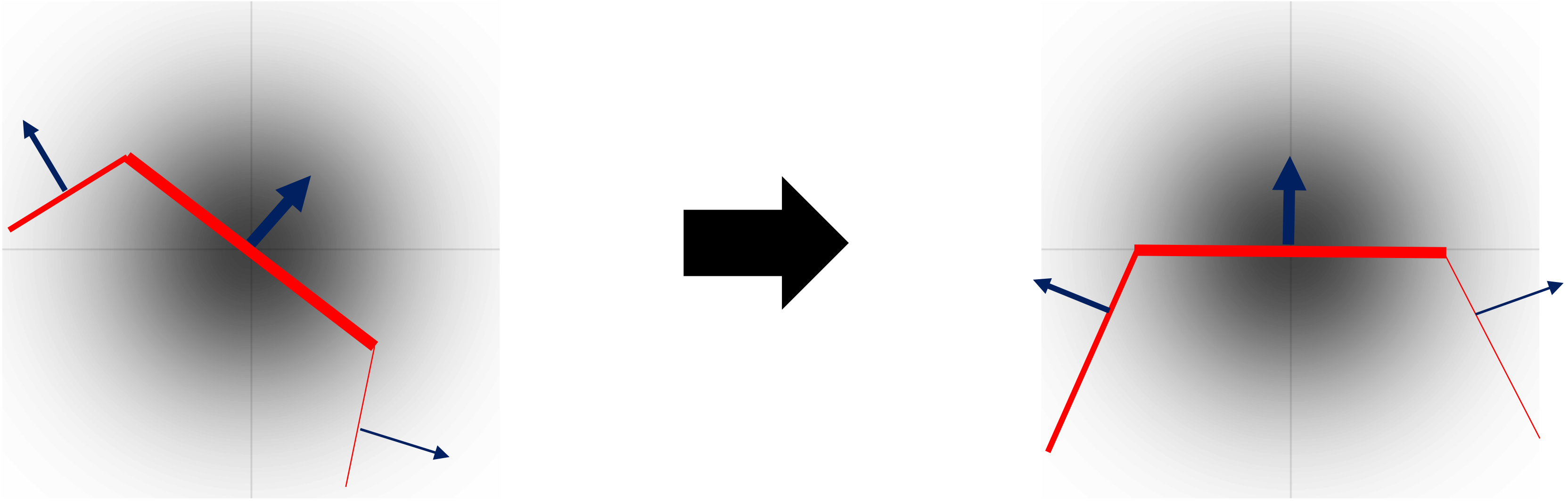
Using Rotation Matrices

- How do we **choose a good rotation?**
  - Heuristic: align heaviest faces with axes (tricky in high dimensions)



Using Rotation Matrices

- How do we **choose a good rotation?**
  - Heuristic: align heaviest faces with axes (tricky in high dimensions)
  - Compose from rational Givens rotations



Implementation

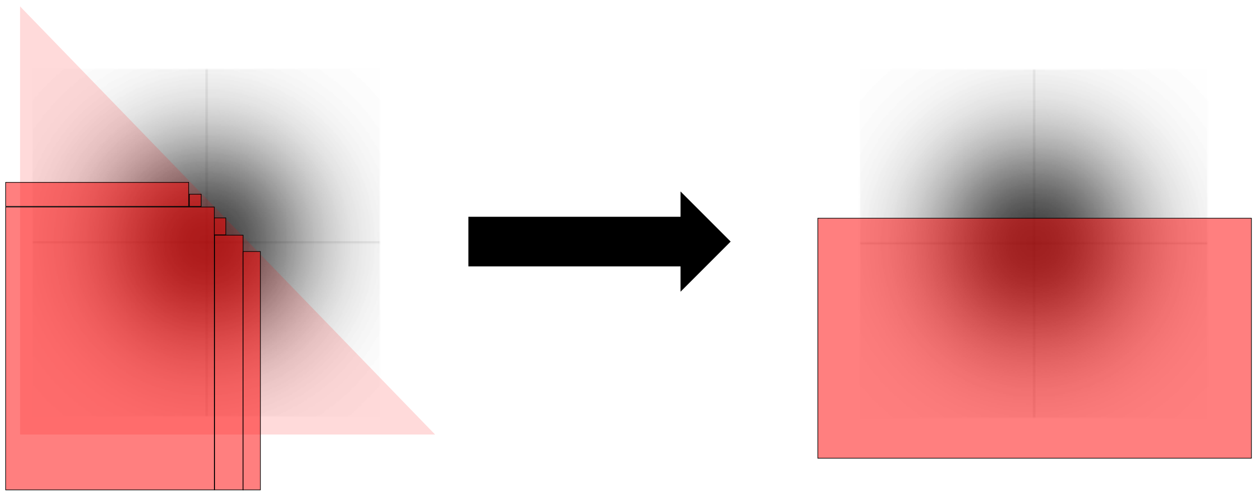
- Python
- Z3

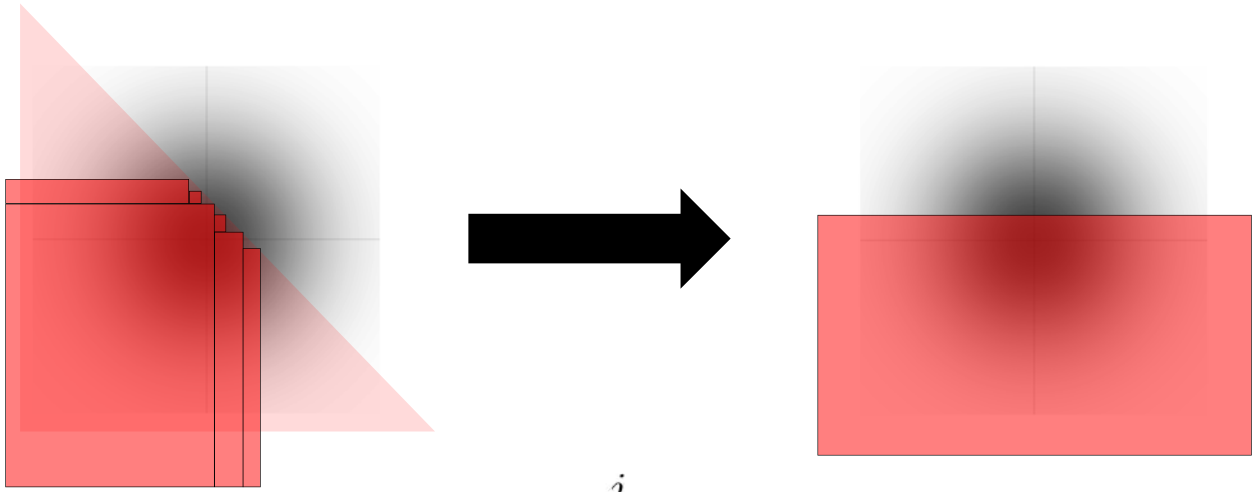
Evaluation

- Randomly generated formulas
  - Generated 8 kinds of formula—40 instances each.
  - Sampled rectangles 100s for each instance.
  - Average improvement between **17%** (3-variable formulas) and **1500%** (7-variable formulas)

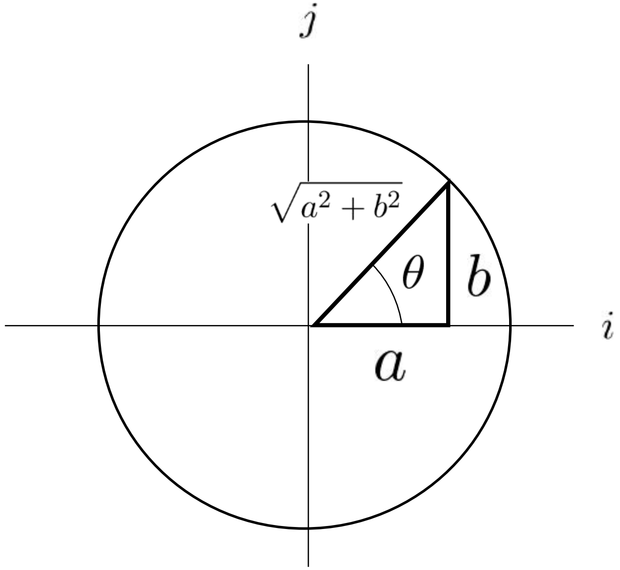
Evaluation

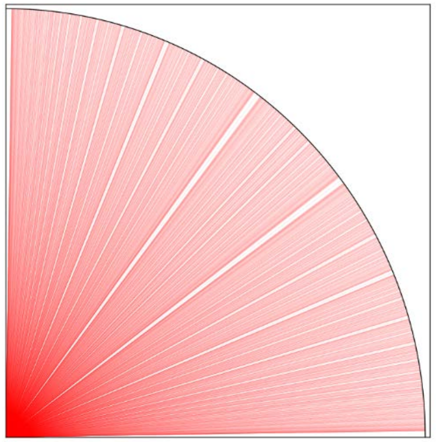
- Randomly generated formulas
  - Generated 8 kinds of formula—40 instances each.
  - Sampled rectangles 100s for each instance.
  - Average improvement between **17%** (3-variable formulas) and **1500%** (7-variable formulas)
- Probabilistic programs from literature
  - Improved efficiency for 10 of 12 example programs.



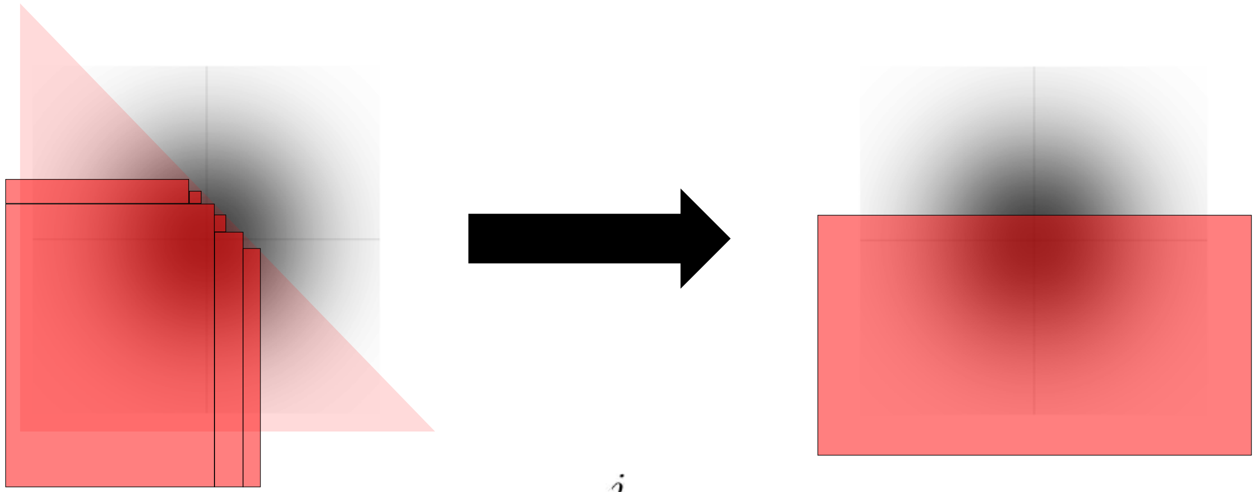


$$G(i,j) = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & -b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & b & \cdots & a & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

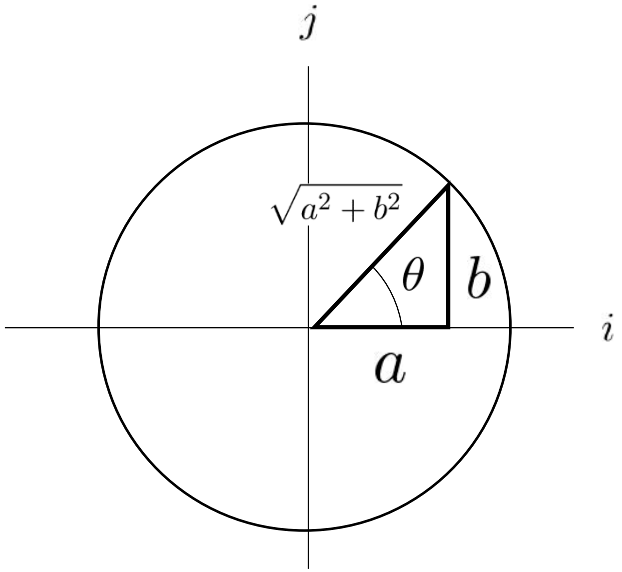


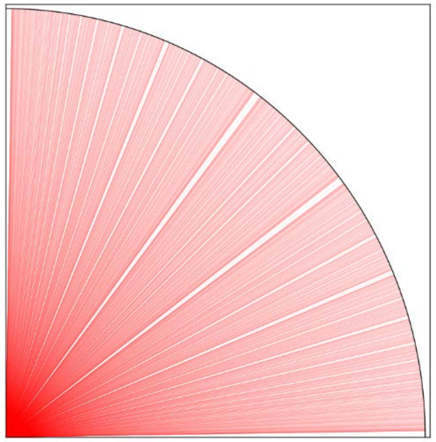


Questions?



$$G(i,j) = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & -b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & b & \cdots & a & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

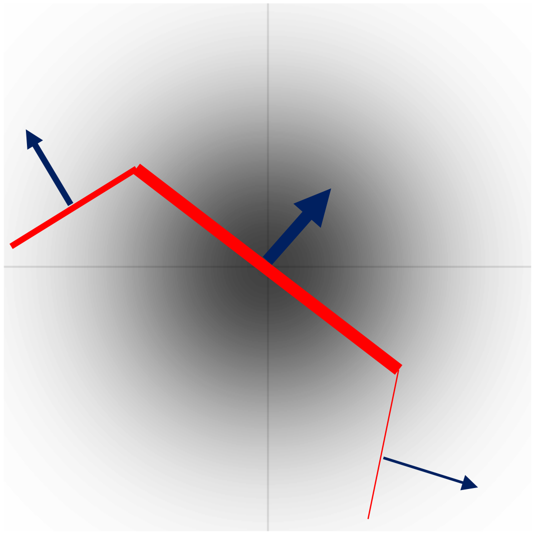




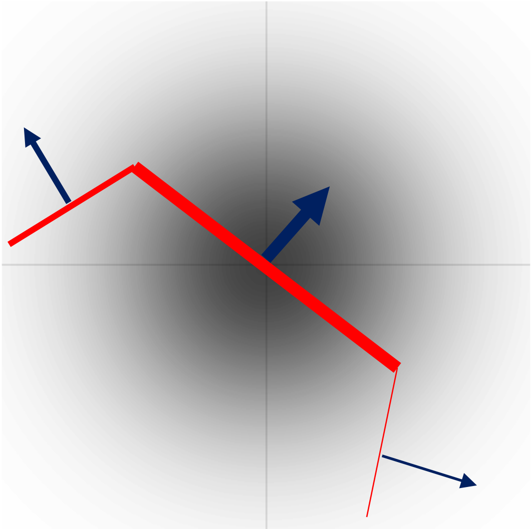
Extras

Using Rotation Matrices

- How do we **choose a good rotation?**



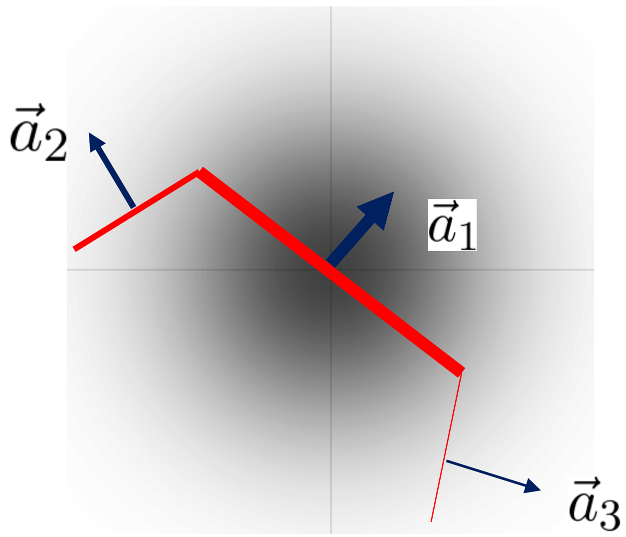
Using Rotation Matrices



- How do we **choose a good rotation?**
- Heuristic:
  - Compute weights of faces

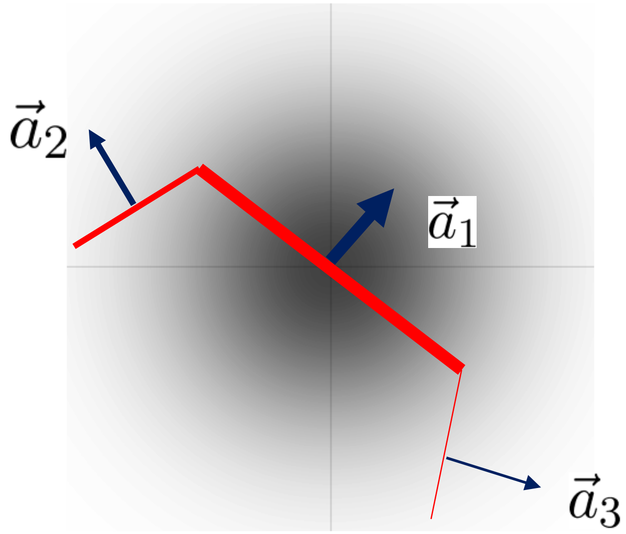
Using Rotation Matrices

- How do we **choose a good rotation?**
- Heuristic:
  - Compute weights of faces
  - Order faces by weight



Using Rotation Matrices

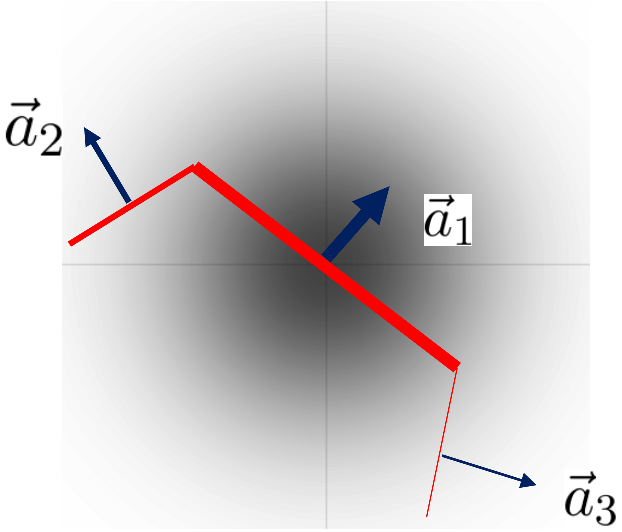
- **How do we choose a good rotation?**
- Heuristic:
  - Compute weights of faces
  - Order faces by weight



$$A = [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n]$$

Using Rotation Matrices

- How do we **choose a good rotation?**
- Heuristic:
  - Compute weights of faces
  - Order faces by weight
  - QR-factorize the face matrix

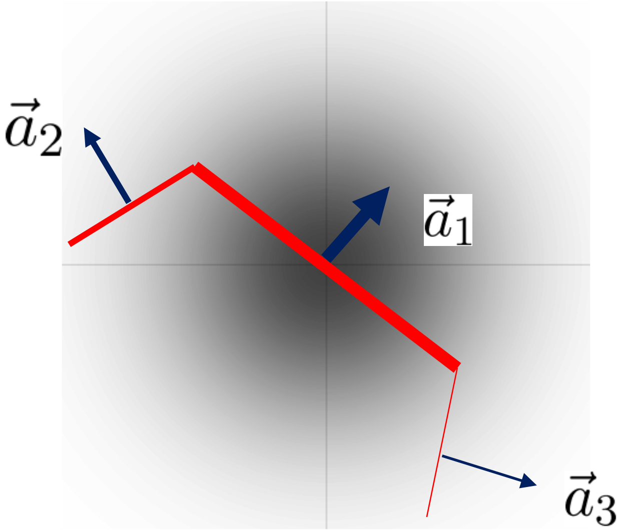


$$A = [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n]$$

$$A = QR$$

Using Rotation Matrices

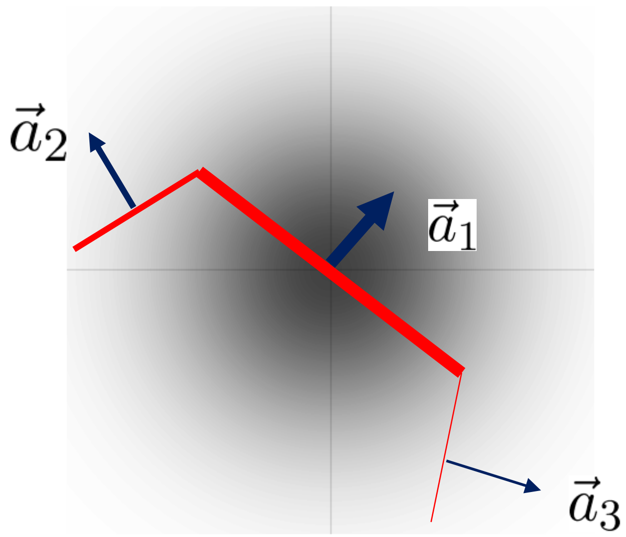
- **How do we choose a good rotation?**
- Heuristic:
  - Compute weights of faces
  - Order faces by weight
  - QR-factorize the face matrix
    - **Use approximate QR-factorization by rational Givens Rotations**



$$A = [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n]$$

$$A = QR$$

Using Rotation Matrices



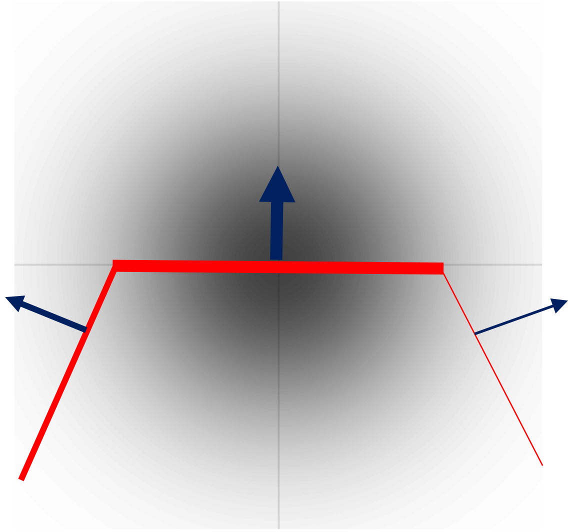
- How do we **choose a good rotation?**
- Heuristic:
  - Compute weights of faces
  - Order faces by weight
  - QR-factorize the face matrix
    - **Use approximate QR-factorization by rational Givens Rotations**
  - Use Q^T as our rotation.

$$A = [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n]$$

$$A = QR$$

Using Rotation Matrices

- **How do we choose a good rotation?**
- Heuristic:
  - Compute weights of faces
  - Order faces by weight
  - QR-factorize the face matrix
    - **Use approximate QR-factorization by rational Givens Rotations**
  - Use Q^T as our rotation.

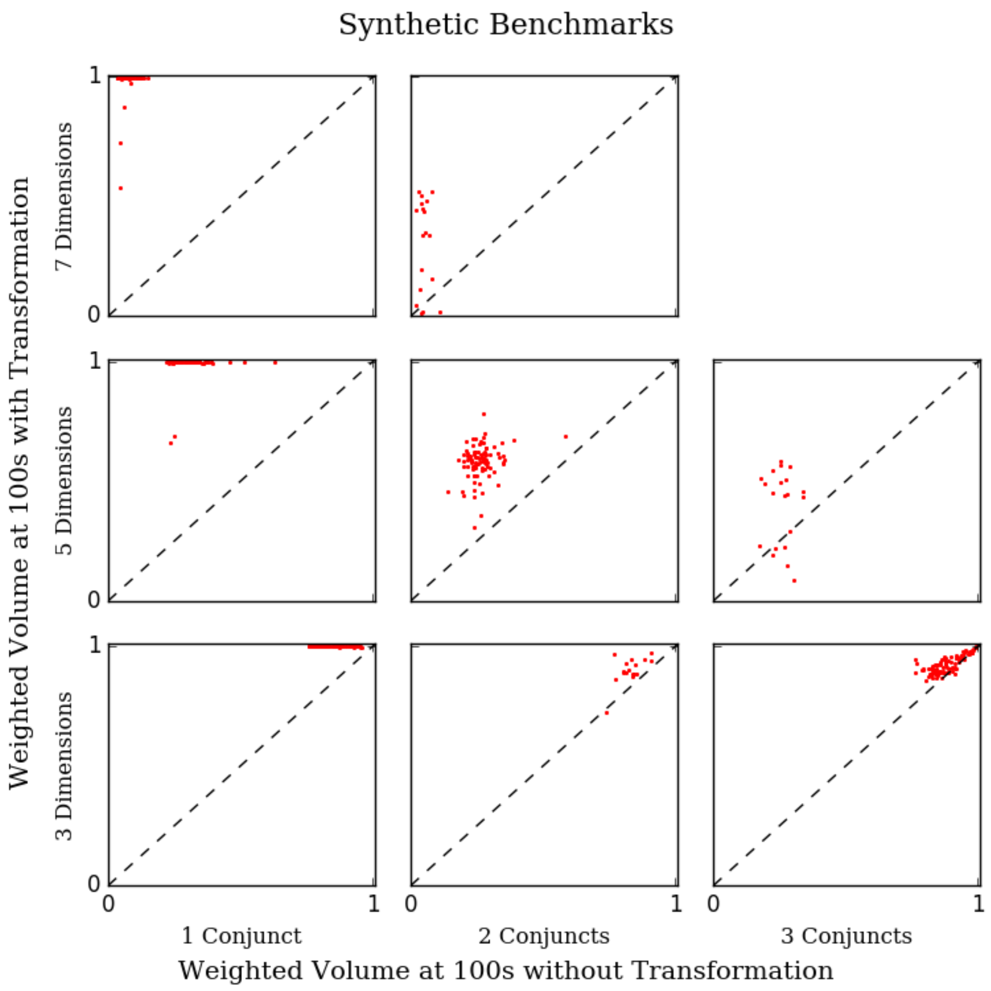


$$A = [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n]$$

$$A = QR$$

Evaluation

- Randomly generated formulas

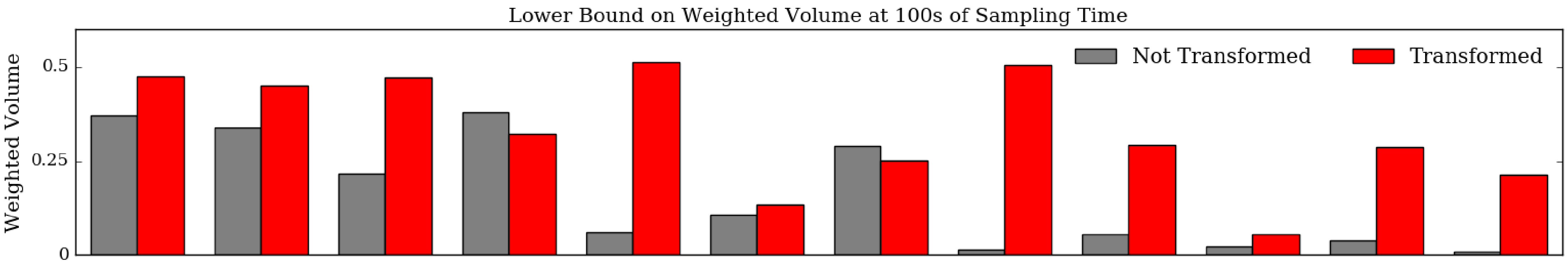


Evaluation

- Randomly generated formulas
- Probabilistic programs

Evaluation

- Randomly generated formulas
- Probabilistic programs



Benchmark	1d-2s-simp	1d-2s-b2	1d-3s-simp	1d-3s-b2	2d-2s-simp	2d-2s-b	2d-2s-b2	2d-3s-simp	3d-2s-simp	3d-2s-b	3d-2s-b2	3d-3s-simp
# Predicates	14	36	18	50	25	61	55	31	36	90	76	45
# Free Vars	3	5	4	7	6	10	8	8	9	15	11	8
# Quant. Vars	12	27	16	36	23	46	43	29	34	67	61	43

Conclusions

- An **orthogonal transformation preprocessing step** can improve sampling efficiency for Weighted Model Integration methods.

- We addressed challenges and limitations of this approach.

Future Work

- Can we **choose better rotations** for this preprocessing step?

- Are there new ways to perform exact Weighted Model Integration, **beyond rectangular decomposition**? E.g., numerical quadrature?