Siamese Networks with triplet loss for Peer Group Scoring

Christian Menn, Daniel Oeltz, Christian Werner

frontmark joint work with RIVACON

Peer Group Clustering

Many finance applications use the concept of peer group clustering, e.g. to validate prices or as a selection criterion:

> "For a reference product, find a set of similar ones "

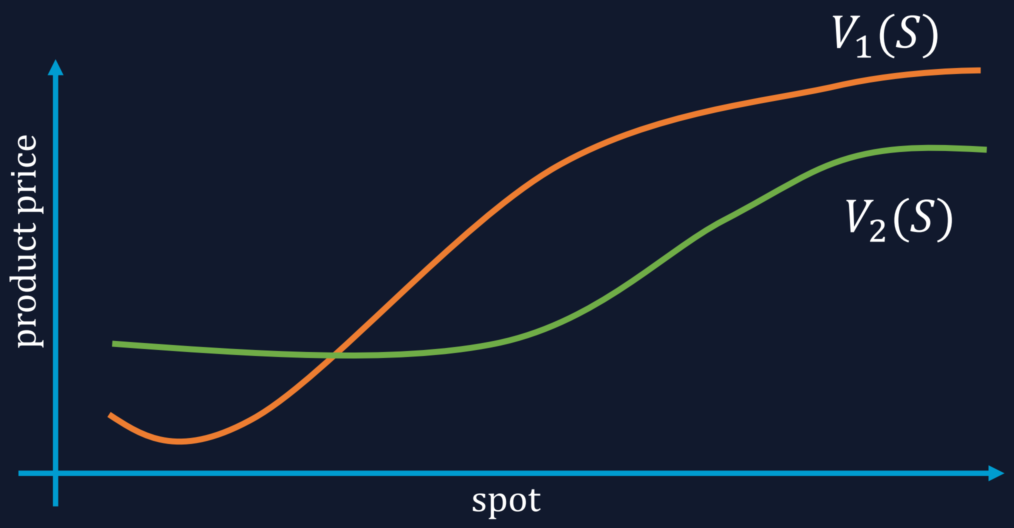
The common approach is to derive peers by defining rules on the product static data:

> "A static ruleset needs iterative, expensive human action and may still result in empty or oversized peer groups calling for further action"

? Find a generic, self-contained process including similarity distances

Peer Group Distance

Compare pricing splines of two products as construction principle to create different peer group distances



$$d^2_{L^2}(V_1, V_2) = \|V_1 - V_2\|_{L^2} = \int (V_1 - V_2)^2 dx$$

$$d^2_{H^1}(V_1, V_2) = |V_1 - V_2|_{H^1} = \int (\partial_x V_1 - \partial_x V_2)^2 dx$$

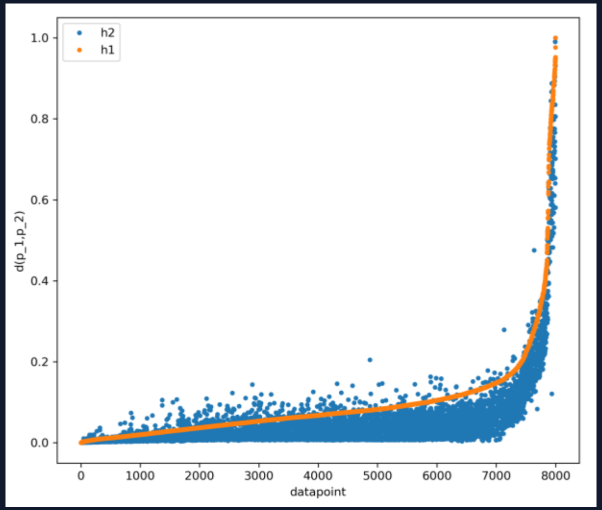
$$d^2_{H^2}(V_1, V_2) = |V_1 - V_2|_{H^2} = \int (\partial_{xx} V_1 - \partial_{xx} V_2)^2 dx$$

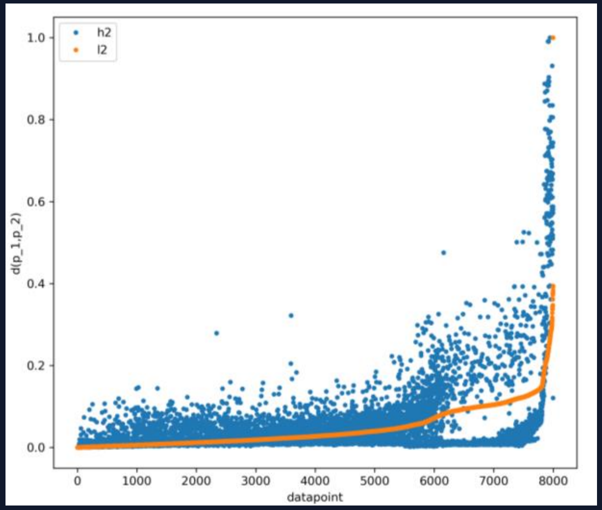
$$d^2_{similarity}(V_1, V_2) = \frac{\int \partial_x V_1 \cdot \partial_x V_2 dx}{|V_1|_{H^1} |V_2|_{H^2}}$$

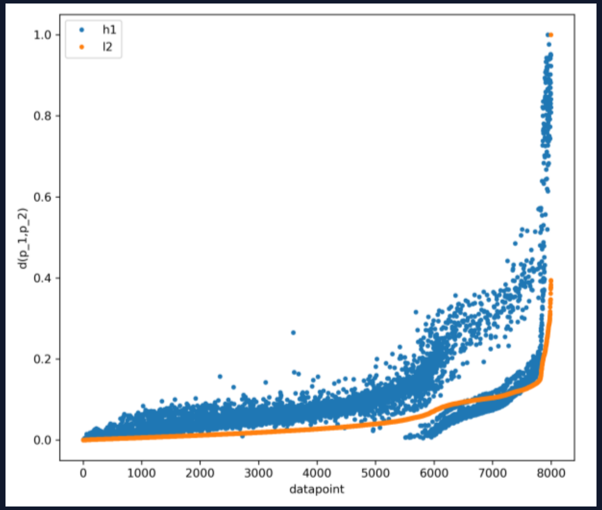
Depending on the application “inconsistencies” may arise

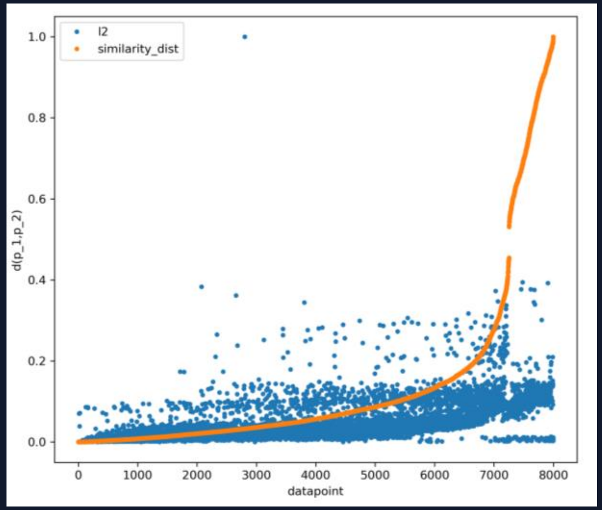
Peer Distances for DAX Structured Products

- Sample with replacement 10'000 pairs of structured products on the DAX
- Compute each of the four different peer distance measures from the previous slides for each of the sampled pairs
- For each distance measure, sort the pairs ascending by one distance (plotted in orange) and compare it to one of the other distances (plotted in blue)



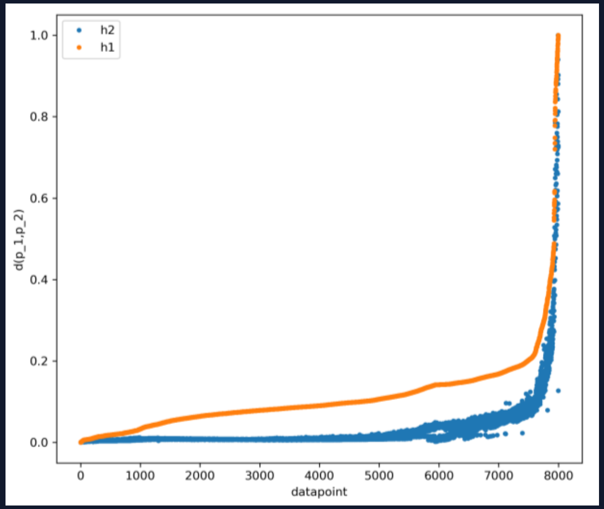


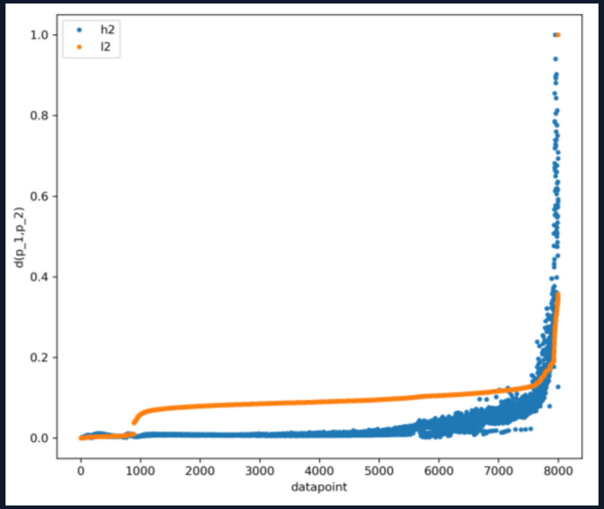


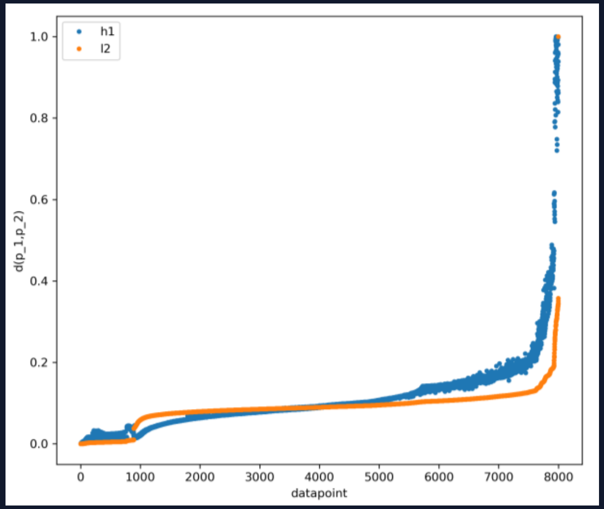


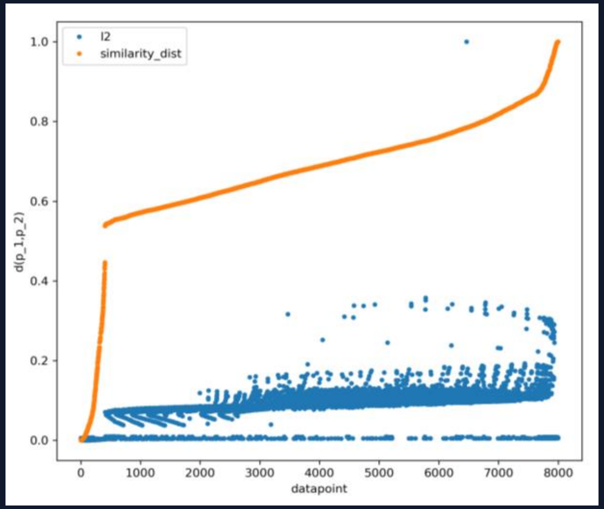
Peer Distances for DAX Structured Products

- Select only one specific DAX structured product
- Compute each of the four peer distance measures between the selected product and all other products
- For each distance measure, sort the computed distance ascending (plotted in orange) and compare it to each of the other distance measures (plotted in blue)



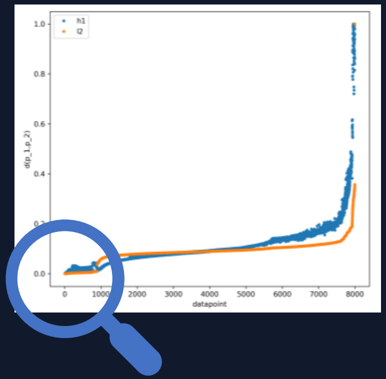


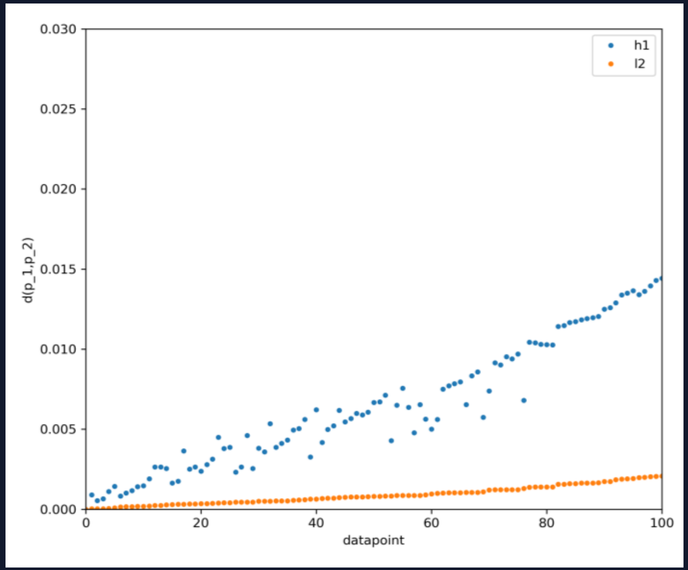


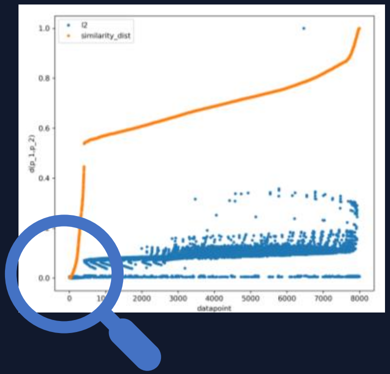


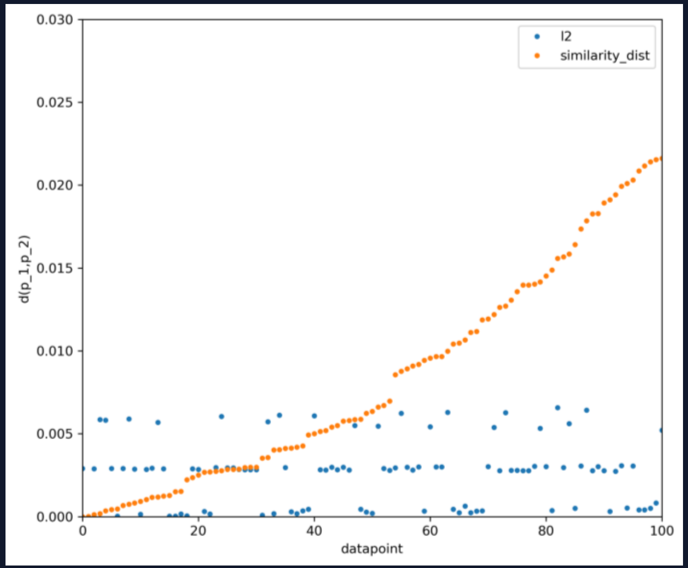
Peer Distances for DAX Structured Products

Zooming into the diagram shows: The different distance measures are not co-monotone. Consequently, different distance measures may result in different peer groups!







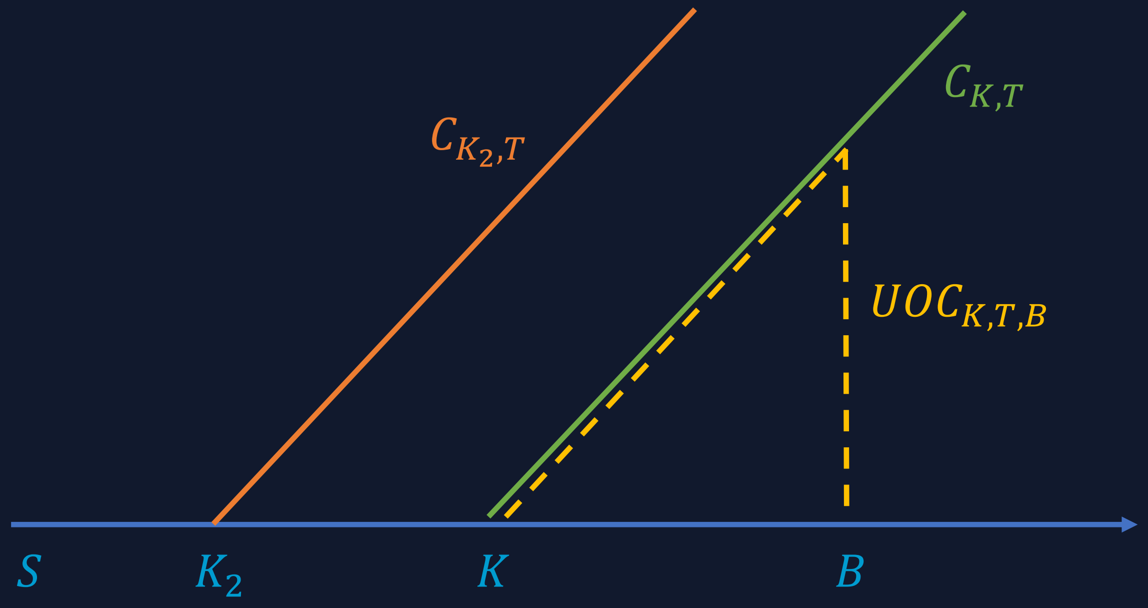


Peer Group Distance-consistency Constraints

- Depending on the application and the user preferences, a-priori consistency conditions (constraints) may exist and must be considered:

$$d(C_{K,T}\,, UOC_{K,T,B}) \leq d(C_{K,T}, C_{K_2,T})$$

with $B \gg S, K_2 \ll K (K_2 \gg K)$

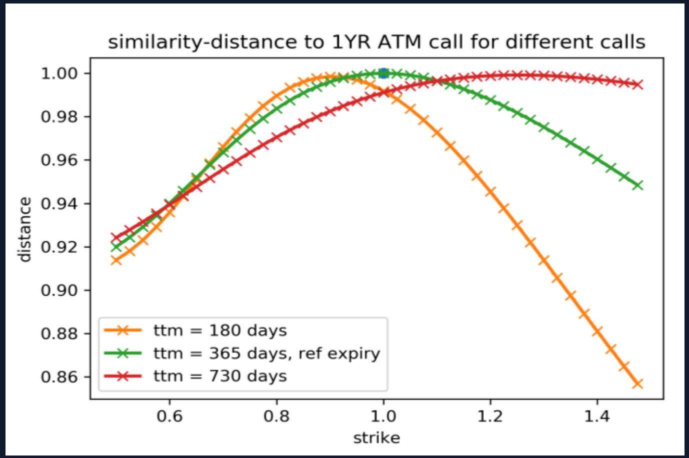


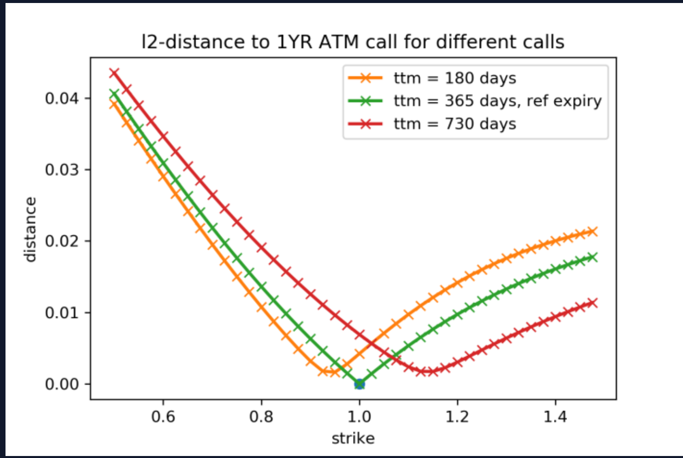
Peer Group Distance-consistency Constraints

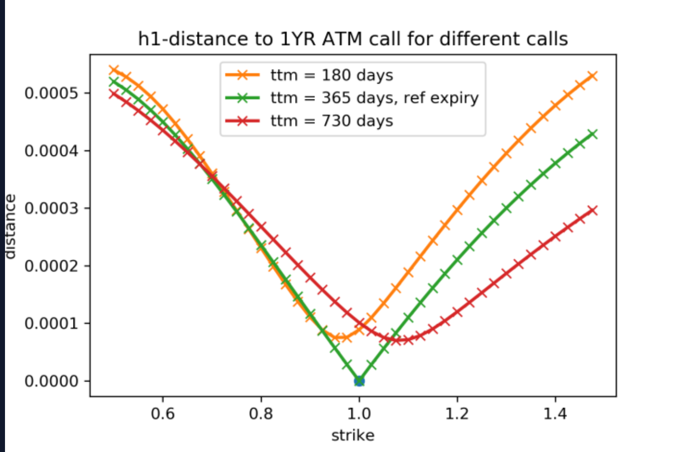
- Obvious violations of Call constraints:

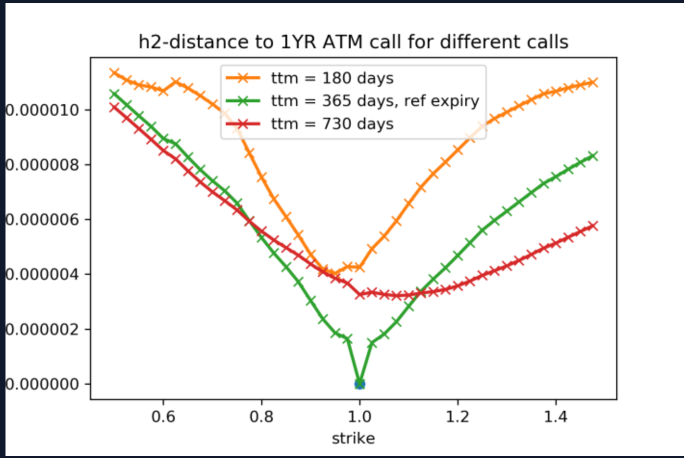
$$d(C_{K,T}\,, C_{K_1,T_1}) \leq d(C_{K,T}, C_{K_2,T_2}) \text{ for all}$$

$$|K - K_1| \leq |K - K_2|, |T - T_1| \leq |T - T_2|$$









More consistency constraints (Call & Put)

- European_P1_P1_P1:

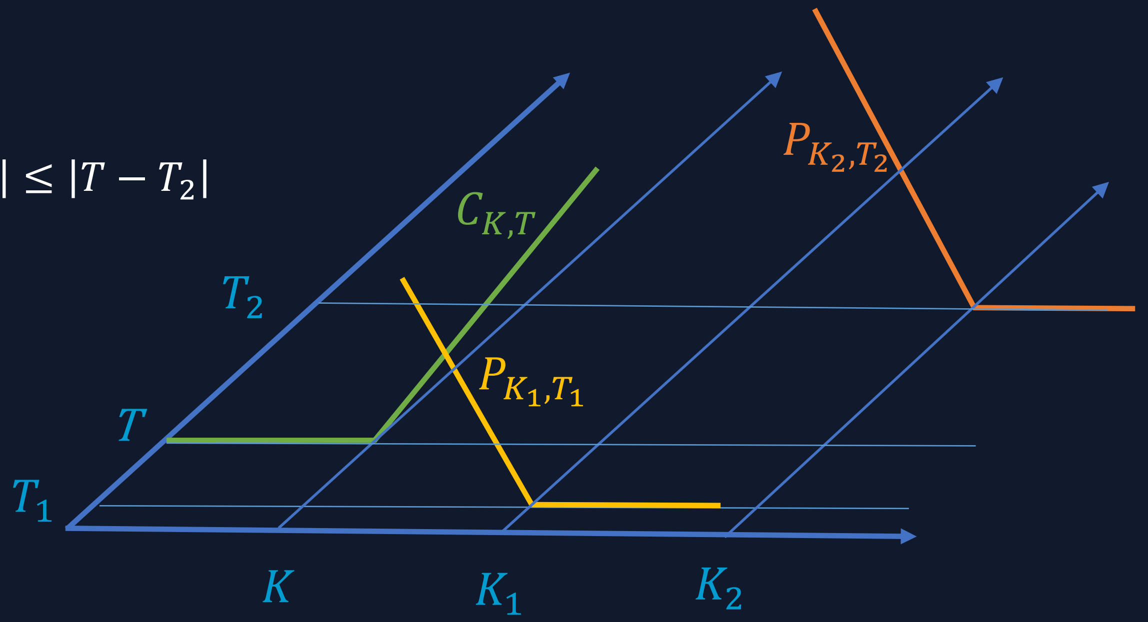
$d(C_{K,T}\, , C_{K_1,T_1}) \le d(C_{K,T}, C_{K_2,T_2})$ for all $|K - K_1| \le |K - K_2|, |T - T_1| \le |T - T_2|$

- European_P1_P1_P2:

$d(C_{K,T}\, , C_{K_1,T_1}) \le d(C_{K,T}, P_{K_2,T_2})$ for all K, K_1, K_2, T, T_1, T_2

- European_P1_P2_P2:

$d(C_{K,T}\, , P_{K_1,T_1}) \le d(C_{K,T}, P_{K_2,T_2})$ for all $|K - K_1| \le |K - K_2|, |T - T_1| \le |T - T_2|$



More consistency constraints (Butterflies)

- Butterfly_Butterfly_Butterfly:

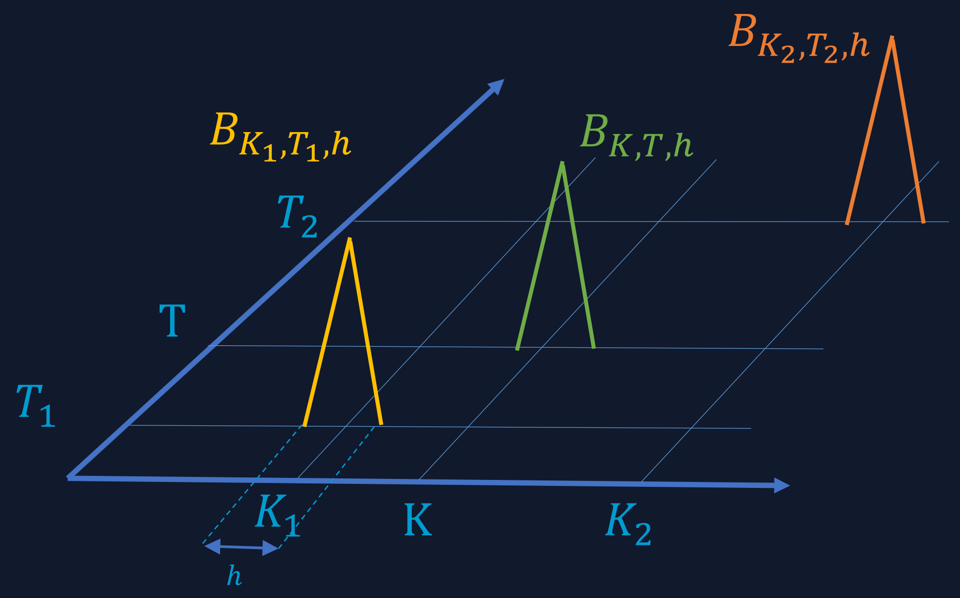
$$d(B_{K,T,h}\ , B_{K_1,T_1,h}) \leq d(B_{K,T,h}, B_{K_2,T_2,h})$$

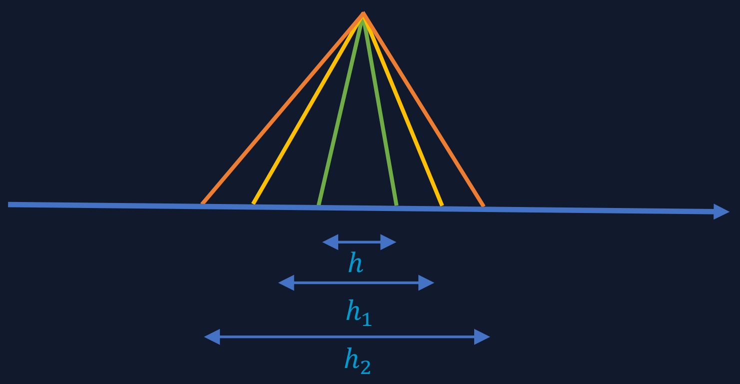
for all $|K - K_1| \leq |K - K_2|, |T - T_1| \leq |T - T_2|$

- Butterfly_Butterfly_Butterfly_2:

$$d(B_{K,T,h}\ , B_{K,T,h_1}) \leq d\big(B_{K,T,h}, B_{K,T,h_2}\big)$$

for all $h < h_1 < h_2$





More consistency constraints (Straddles)

- Straddle_Straddle_Straddle:

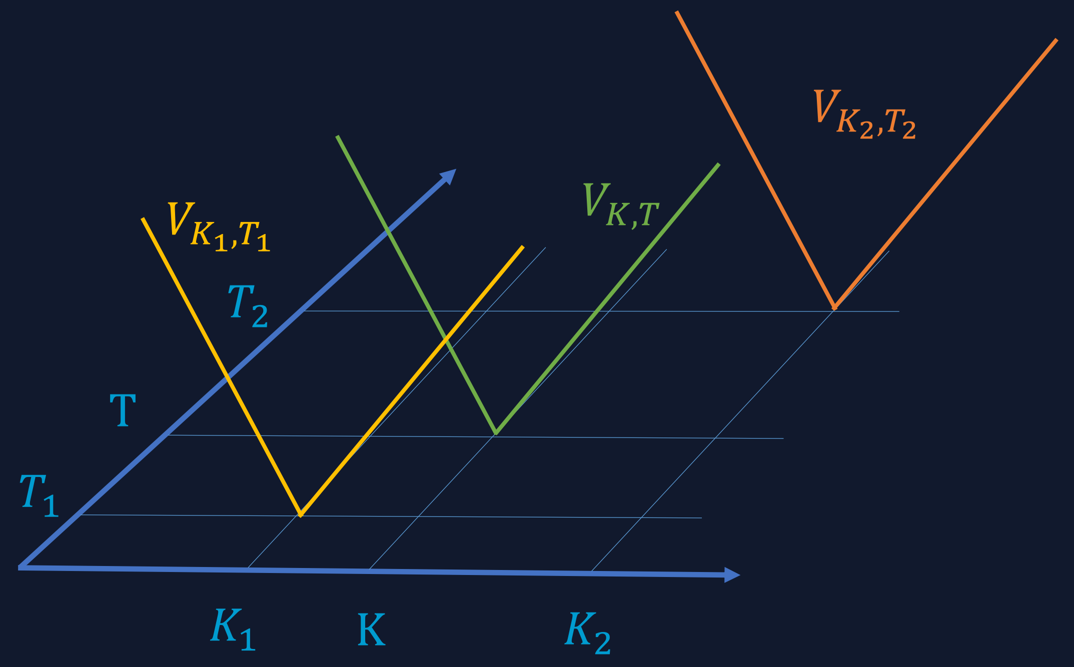
$$d(V_{K,T}\,,\,V_{K_1,T_1}) \leq d(V_{K,T},V_{K_2,T_2})$$

for all $|K - K_1| \leq |K - K_2|, |T - T_1| \leq |T - T_2|$

- Straddle_P1_P2:

$$d(V_{K,T}\,,V^1_{K_1,T_1}) \leq d(V_{K,T},V^2_{K_2,T_2})$$

for all K, K_1, K_2, T, T_1, T_2 where $V^1_{K_1,T_1}$ and $V^2_{K_1,T_1}$ are call or put



More consistency constraints (Barriers)

- UIC_UIC_UIC:

$$d(UIC_{K,T,b}\,, UIC_{K,T,b_1}) \leq d\left(UIC_{K,T,b}, UIC_{K,T,b_2}\right) \text{ for all b} < b_1 < b_2$$

- UIC_UIC_UIC_2:

$$d(UIC_{K,T,b}\,, UIC_{K_1,T,b}) \leq d\left(UIC_{K,T,b}, UIC_{K_2,T,b}\right) \text{ for all } |K - K_1| \leq |K - K_2|$$

- UIC_UIC_UIC_3:

$$d(UIC_{K,T,b}\,, UIC_{K,T_1,b}) \leq d\left(UIC_{K,T,b}, UIC_{K,T_2,b}\right) \text{ for all } |T - T_1| \leq |T - T_2|$$

More consistency constraints (Barriers)

- UIC_C_C:

$$d(UIC_{K,T,b}\,, C_{K_1,T}) \leq d\left(UIC_{K,T,b}, C_{K_2,T}\right) \text{ for all } |K-K_1| \leq |K-K_2|$$

- C_UIC_UIC:

$$d(C_{K,T}\,, UIC_{K,T,b_1}) \leq d\left(C_{K,T}, UIC_{K,T,b_2}\right) \text{ for all } b_1 < b_2$$

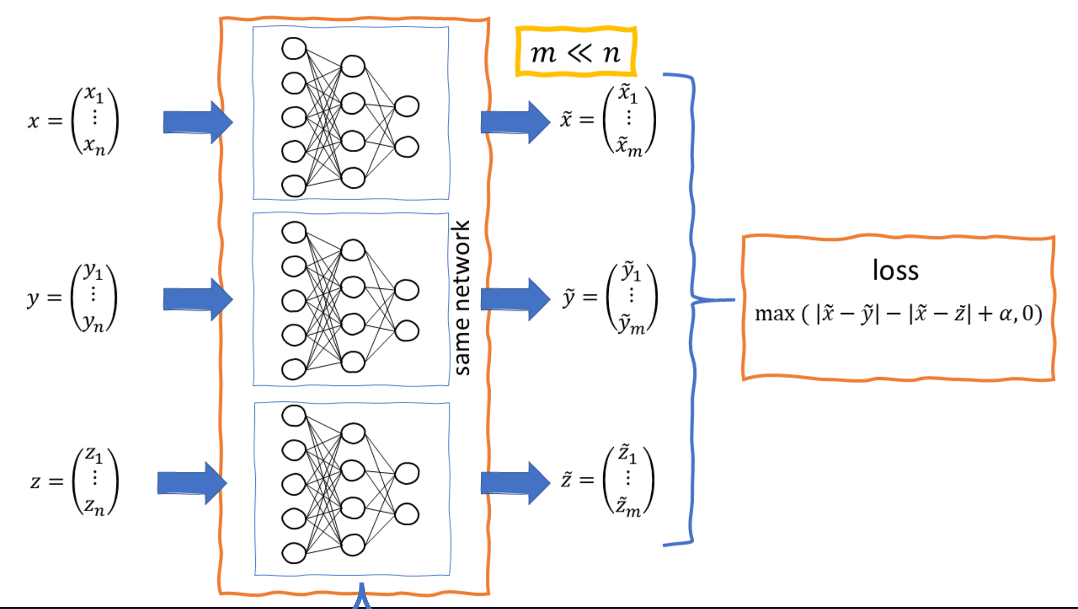
- UOC_UOC_UOC, UOC_UOC_UOC_2, UOC_UOC_UOC_3, C_UOC_UOC:

analogously to the UIC versions above

How to ensure that all application dependent constraints hold for a peer distance measure?

The Machine Learning Approach

- Define problem dependent constraints on the distance
- Find new distance measure (e.g. based on the distances seen on previous slides) that minimizes inconsistencies
- Compute new distance by means of a Siamese network



inner network M

$$d(x, y) = \|M(x) - M(y)\|$$

How to choose α?
Idea: Choose different α for each sample, depending on some similarity score d_{ref} :

$$\alpha_{x,y,z} = d_{ref}(y, z) - d_{ref}(x, y)$$

Loss:

$$\max(|\tilde{x} - \tilde{y}| - |\tilde{y} - \tilde{z}| + \alpha_{x,y,z}, 0)$$

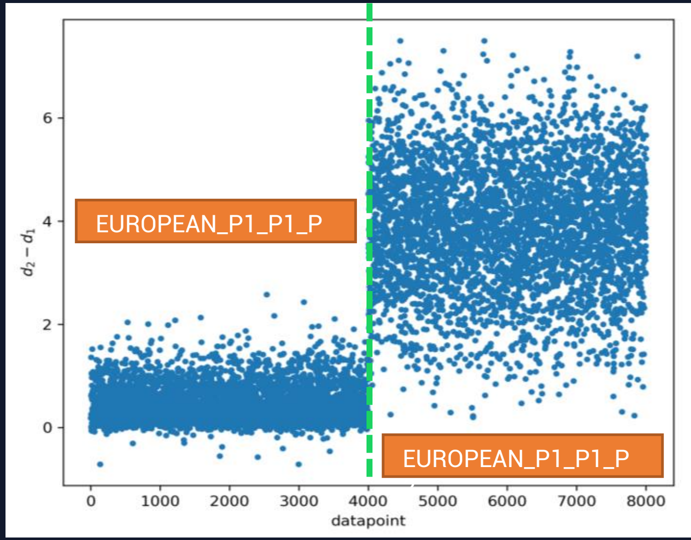
The Machine Learning Approach - Setup

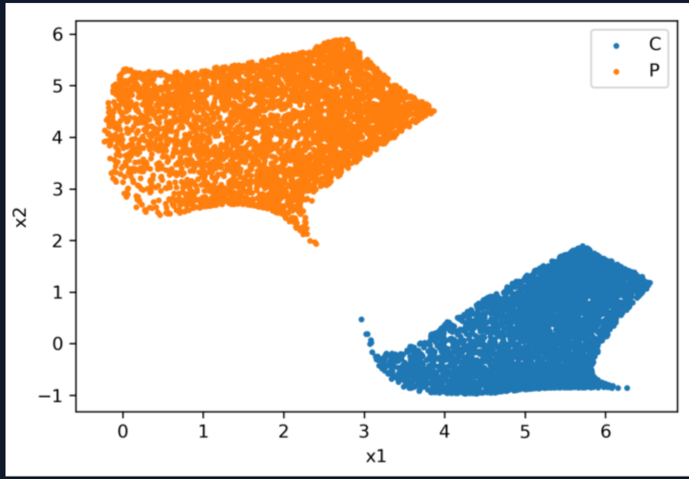
- Data sampled for each constraint with uniformly distributed
  - maturity between 20 days and 4 years
  - strikes between 60% and 140% spot moneyness
- 4000 samples for each constraint
- Input data:
  - Pricing spline with uniform spots between 55% and 145% spot moneyness, yielding 20 spline points)
  - Zero interest rates and no dividends
- Time to maturity, Vega, Theta
- Reference similarity score: $d^2_{ref}(.,.) = \left(1.0 - d^2_{similarity}(.,.)\right) d^2_{L^2}(.,.)$

1st Example

- Very simple inner Network with 3 layers:
  - 10 neurons in 1st layer
  - 5 neurons in 2nd layer
  - 2 neurons in output layer (m=2)
- Two Constraints:
  - European P1_P1_P1: Calls are closer to each other if when maturity/strikes are closer
  - European P1_P1_P2: Calls are closer to Calls then Calls to Puts

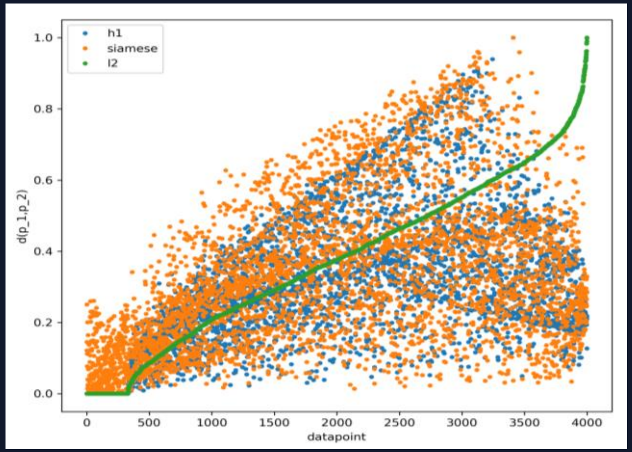
# inconsistencies d_{ref}	743	9.3%
# inconsistencies Siamese	215	2.7%

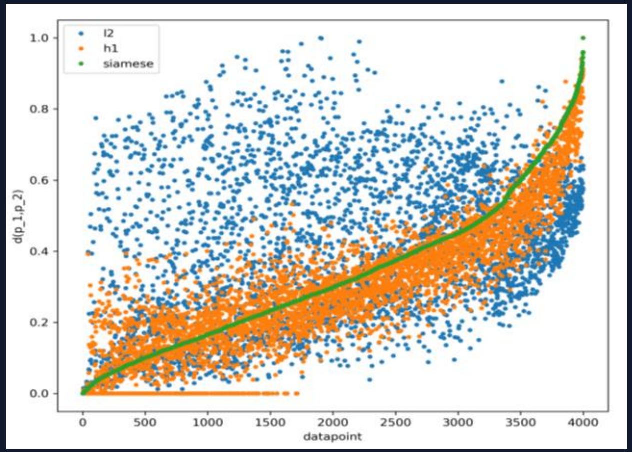




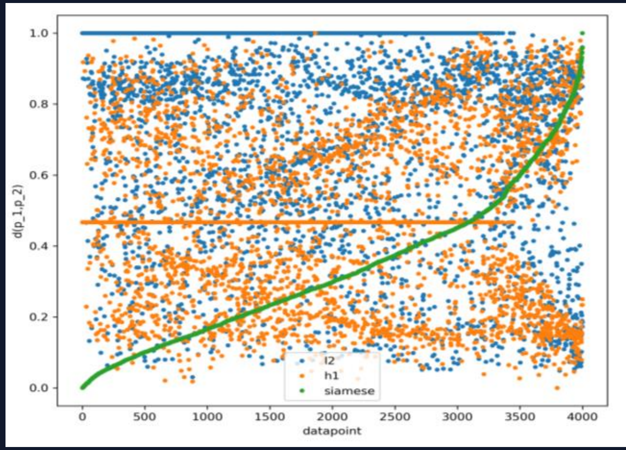
1st Example - Spline peer distances

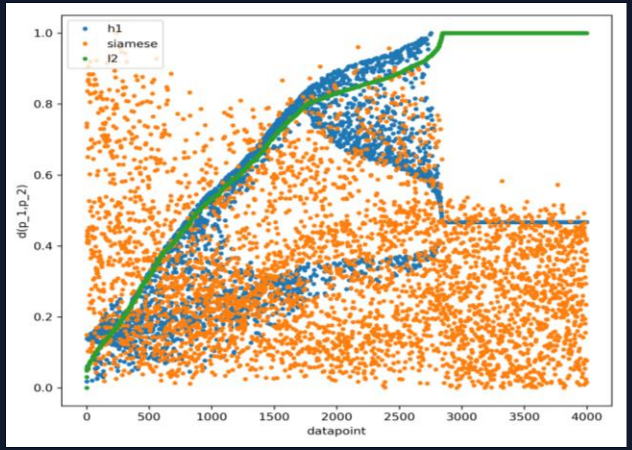
- Sample two products from dataset
- Compute peer distance and sort ascending (by distance plotted in orange)





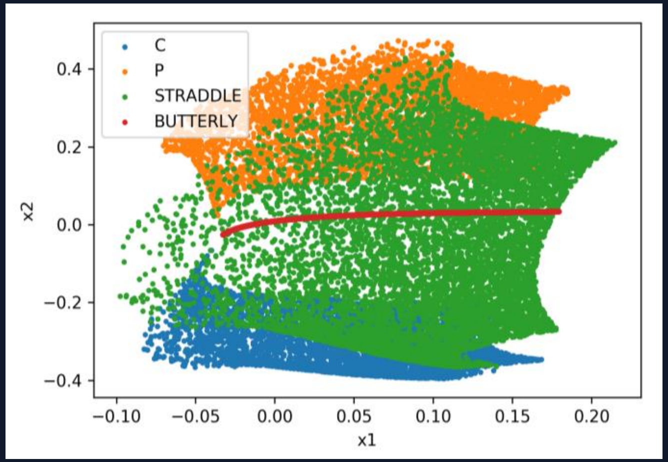
- Select one product and compute peer distance to all other products
- Sort (by distance plotted in orange) computed distances ascending

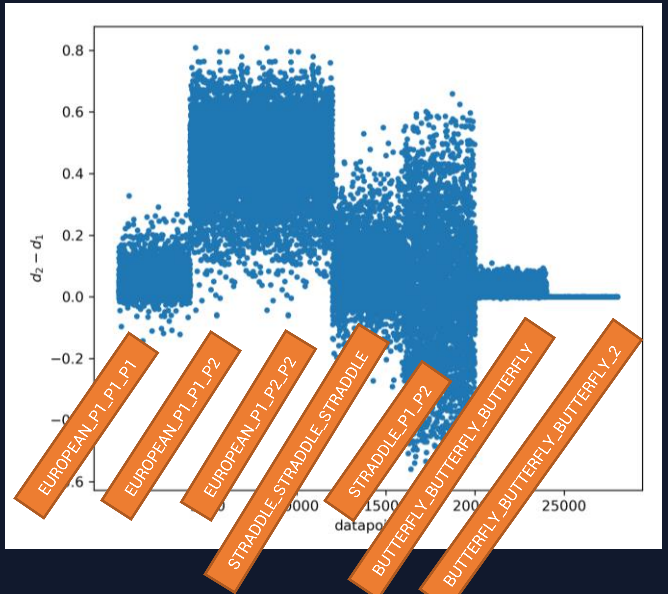




1st Example – Does the model generalize?

Data	similarity distance		Siamese distance	
	absolute	%	absolute	%
EUROPEAN_P1_P1_P1	737	18.43%	210	5.25%
EUROPEAN_P1_P1_P2	6	0.15%	5	0.125%
total	*743*	*9.3%*	*215*	*2.7%*
EUROPEAN_P1_P2_P2	1919	47.98%	2098	52.45%
STRADDLE_STRADDLE_STRADDLE	563	14.08%	567	14.18%
STRADDLE_P1_P2	1762	44.05%	1914	47.85%
BUTTERFLY_BUTTERFLY_BUTTERFLY	710	17.75%	64	1.60%
BUTTERFLY_BUTTERFLY_BUTTERFLY_2	693	17.325%	277	6.93%
total	6390	22.82%	5135	18.33%





Improvements

- Siamese Networks consider relations between distances, but the value of a distance may not give a good intuition on the level of similarity between products.
- Solution 2: For $\boldsymbol{\beta_1}$<1.0<$\boldsymbol{\beta_2}$:

$$\beta_1 d_{ref}(x,y) \leq |\tilde{x} - \tilde{y}| \leq \beta_2 d_{ref}(x,y)$$

  leads to modified loss

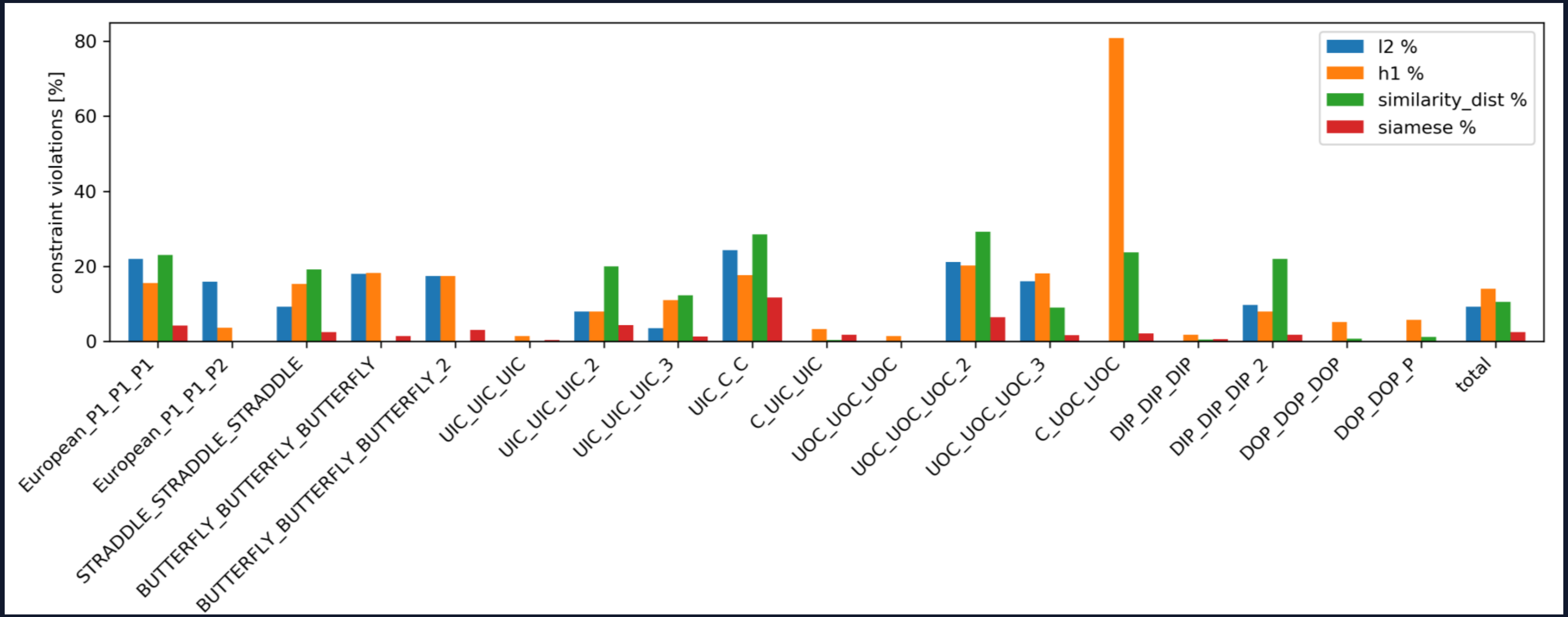
$$\gamma \max\left(|\tilde{x} - \tilde{y}| - |\tilde{y} - \tilde{z}| + \alpha_{x,y,z}, 0\right) + \max(\beta_1 d_{ref}(x,y) - |\tilde{x} - \tilde{y}|) + \max(|\tilde{x} - \tilde{y}| - \beta_2 d_{ref}(\tilde{x},\tilde{y}), 0)$$

- Solution 1 (used e.g. for face recognition problems): Output $\tilde{\boldsymbol{x}}$ in Siamese Network is scaled to vectors of unit length $\frac{\tilde{x}}{|\tilde{x}|_2}$

2^{nd} Example

- Include 18 different constraints
- Each constraint is sampled with 2'500 datapoints points resulting in an overall training set of 45'000 datapoints
- Neural Network with (15,10,4) layer structure (714 trainable parameters) and an output layer normalized to one
- Adam optimizer with manual reduction of the learning rate

2nd Example



	l2	h1	Similarity Distance	Siamese Network
Inconsistencies (training set)	9.0%	14.4%	11.1%	2.2%
Inconsistencies (test set)	9.2%	14.0%	10.5%	2.4%

Conclusion

A Generic approach using Siamese Networks to create application specific peer group scores has been setup and tested.

Siamese Network significantly reduces inconsistencies compared to classical function norms on pricing splines.

A simplified Plug and Play jupyter notebook can be found on GitHub:

https://github.com/pailabteam/ml_finance/

Contact

Christian Menn

c.menn@rivacon.com

+49 1577 3079723

Daniel Oeltz

d.oeltz@rivacon.com

+49 1577 7861885

Christian Werner

c.werner@frontmark.de

+49 172 4560471