# Machine-Learning Calabi-Yau 3-folds in Weighted $\mathbb{P}^4$

Edward Hirst

City, University of London
London Institute of Mathematical Sciences

*edward.hirst@city.ac.uk*
arXiv: 2112.xxxx
work with Prof. David Berman, Prof. Yang-Hui He

String Data 2021

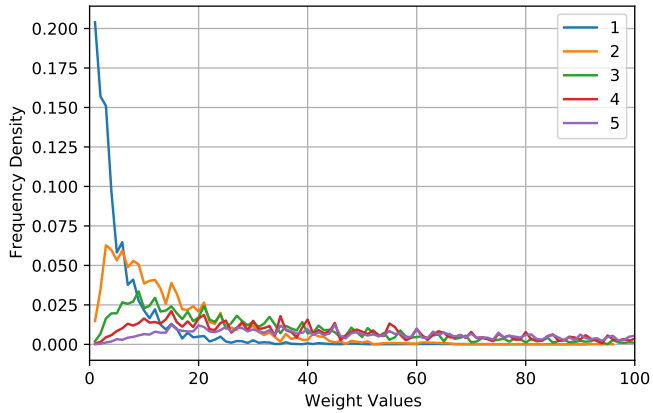
# Overview

1. Datasets
   - CY Data
   - Generated Data
2. Analysis
   - PCA
   - Hodge Clustering
3. ML
   - Hodge Numbers
   - Calabi-Yau Property
4. Summary

# Datasets

## Calabi-Yau Data

$\Rightarrow$ 7555 allowed 5-vectors of coprime $\mathbb{Z}^+$ that define weighted-$\mathbb{P}^4$s which admit a CY 3-fold hypersurface
...with hypersurface Hodge numbers.
(e.g. $\{[1,1,1,1,1],[1,101]\}$ )
Data provided within the KS database.

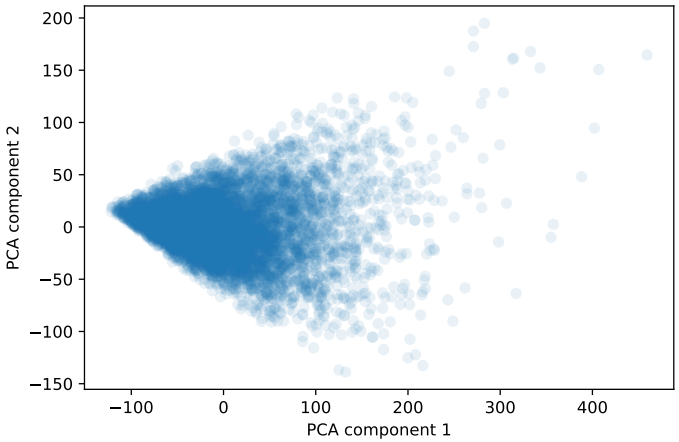



CY weight distribution

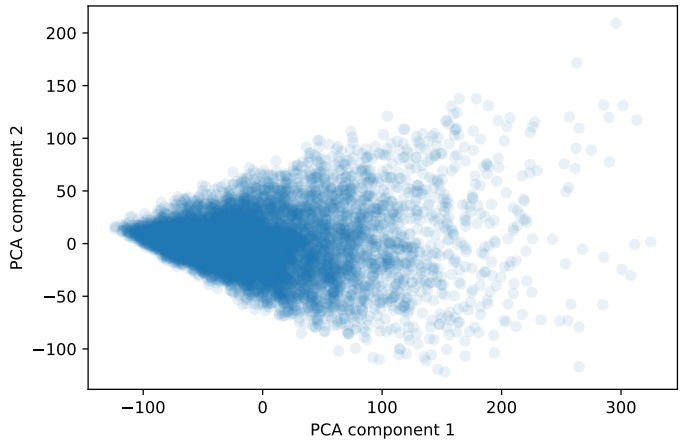
## Generated Data

For Analysis & Performance comparison, use 3 generated datasets:
(a) 'Random', (b) 'Coprime', (c) 'Transverse'
*n.b.* transversity requires: $\forall w_i \exists w_j \ s.t. \ w_i | \left( \sum(w_k) - w_j \right)$

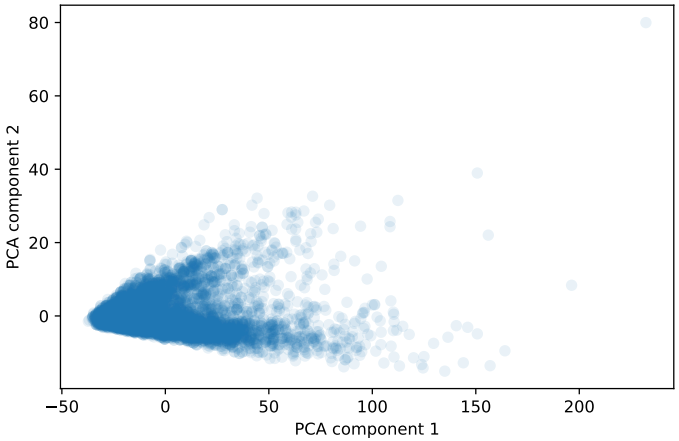
# Analysis: Principal Component Analysis



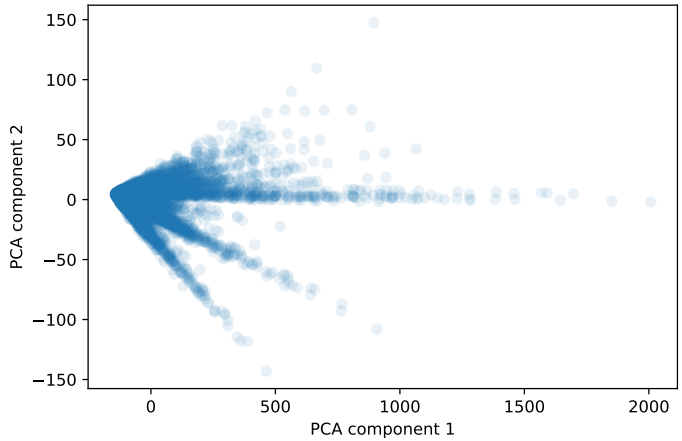

(a) Random dataset




(b) Coprime dataset




(c) Transverse dataset

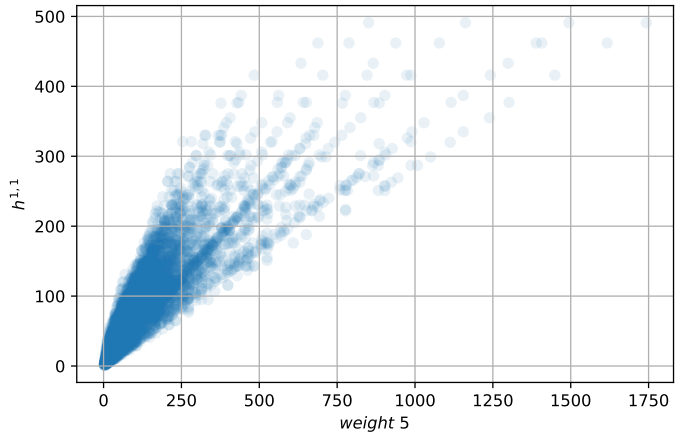



(d) CY dataset

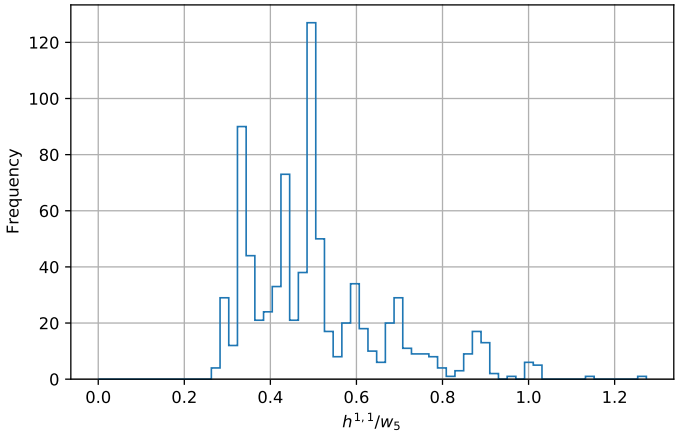
# Analysis: Hodge Clustering

## Plotting weights against $h^{1,1}$

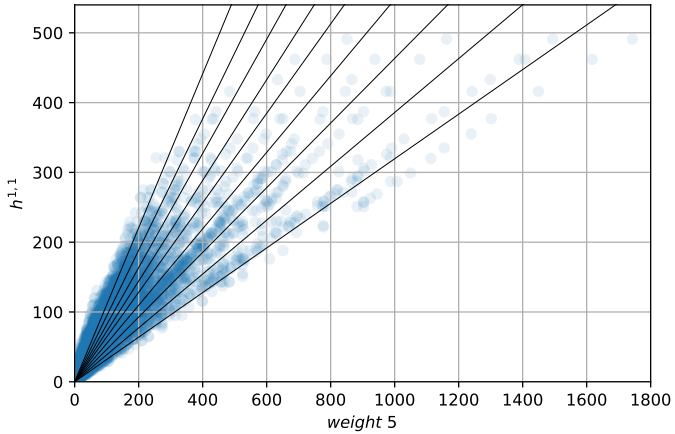
A surprising linear behaviour arises between the weights and $h^{1,1}$.
Further surprising is the apparent separation (alike PCA) into lines
$\Rightarrow$ K-Means Clustering.




CY $w_5$ vs $h^{1,1}$




Histogram of Gradients
$(h^{1,1}/w_5)$




Clustered outputs:
$\hat{\mathcal{I}}/range = 0.00084$

# ML: Hodge Numbers

## ML Regression Problem

5-vector $[w_i] \longmapsto \{h^{1,1}, h^{2,1}, \chi\}$

Regressor ReLU NN (layers: (32,64,32), MSE loss, Adam, 5-fold cross-validation)

Measure $R^2 = 1 - \frac{\sum(y_{true} - y_{pred})^2}{\sum(y_{true} - y_{truemean})^2} \in (-\infty, 1]$

| Measure | Parameter | | |
|---|---|---|---|
| | $h^{1,1}$ | $h^{2,1}$ | $\chi$ |
| $R^2$ | 0.9630 $\pm$ 0.0015 | 0.9450 $\pm$ 0.0133 | 0.9510 $\pm$ 0.0023 |

Exist known formulas from Vafa, KS, Batyrev, etc, but highly non-trivial.

e.g. $\chi = \frac{1}{\sum_i(w_i)} \sum_{l,r=0}^{\sum_i(w_i)-1} \left[ \prod_{i|lq_i \& rq_i \in \mathbb{Z}} \left(1 - \frac{1}{q_i}\right) \right]$, for $q_i = w_i / \sum_i(w_i)$

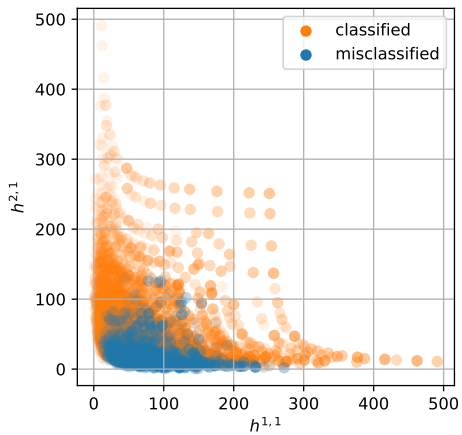
# ML: Calabi-Yau Property

## ML Classification Problem

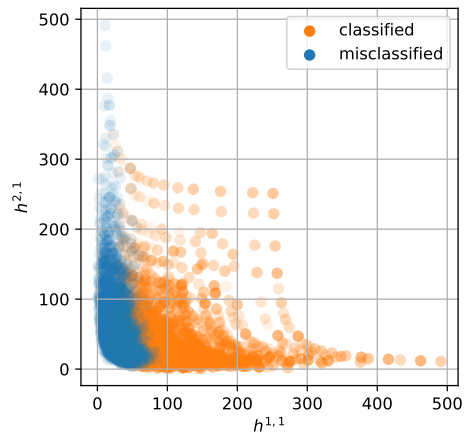
$$\{[w_i]_{\text{CY}}, [w_i]_{\textit{non-CY}}\} \longmapsto \{\text{CY}, \textit{non-CY}\}$$

Architectures: Logistic Regressor, Support Vector Machine, & Classifier ReLU NN

| Architecture | Dataset | | |
|---|---|---|---|
| | Random | Coprime | Transverse |
| LR | 0.7152 $\pm$ 0.0035 | 0.7199 $\pm$ 0.0037 | 0.7430 $\pm$ 0.0065 |
| SVM | 0.7253 $\pm$ 0.0029 | 0.7116 $\pm$ 0.0029 | 0.7464 $\pm$ 0.0014 |
| NN | 0.9189 $\pm$ 0.0037 | 0.9178 $\pm$ 0.0030 | 0.7575 $\pm$ 0.0024 |

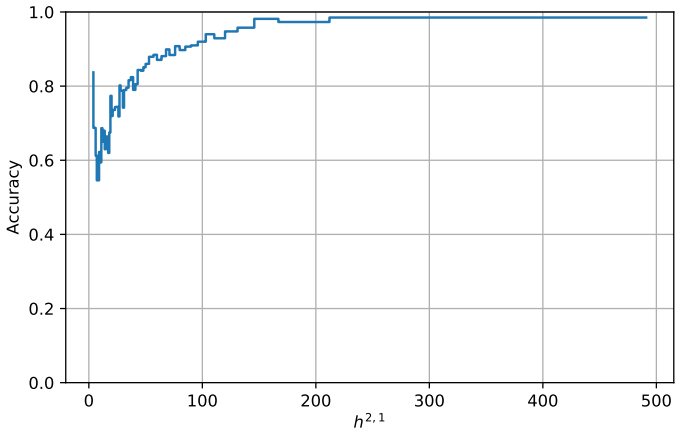



LR CY misclassification (Random)

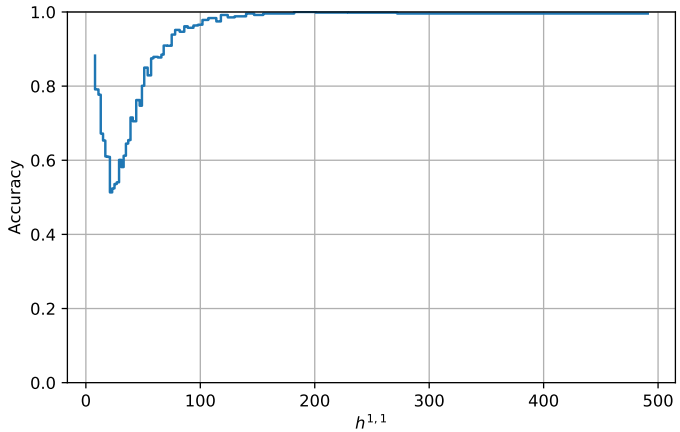



LR CY misclassification (Transverse)

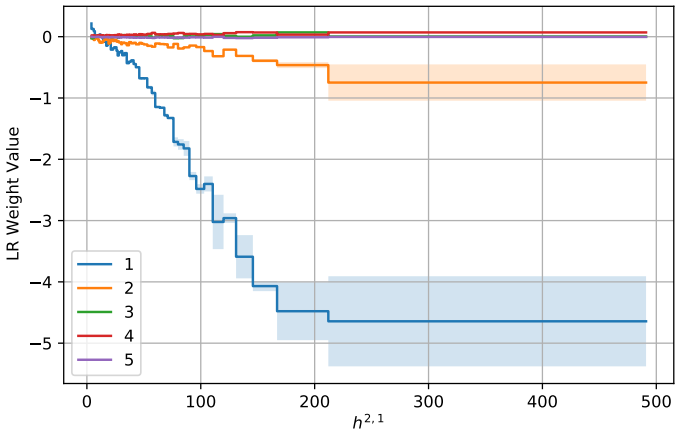
# ML: Hodge Partitioning



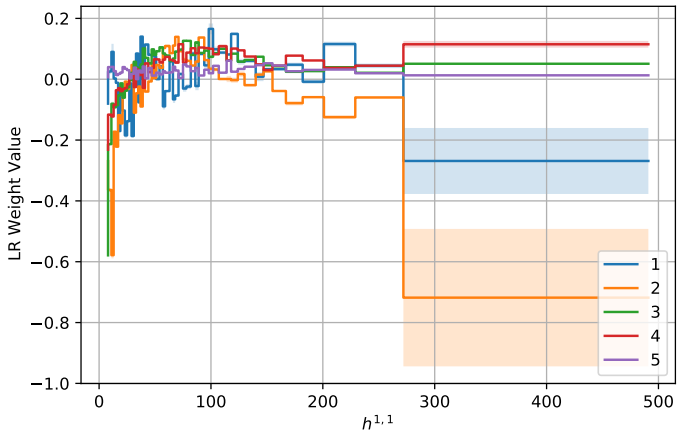

$h^{2,1}$-partitioned LR accuracies




$h^{1,1}$-partitioned LR accuracies




$h^{2,1}$-partition LR weights




$h^{1,1}$-partition LR weights

# Summary

## Unsupervised ML

The Calabi-Yau data sorts itself into a clear clustering based on $h^{1,1}$, confirmed by K-Means, and corroborated by PCA.

## Supervised ML

CY Hodge number formulas can be learnt exceptionally well from $\mathbb{P}^4$ weights, despite complicated formulas.
CY weights can be well distinguished from non-CY weights with simple LR architecture; dependent on the data used for training performance is optimal at different extremes of $h^{1,1}$ or $h^{2,1}$.