Machine learning for Calabi–Yau manifolds

Harold Erbin

Asc, Lmu (Germany)

Machine Learning Landscape, Ictp, Trieste
– 12th December 2018

Unterstützt von / Supported by
Alexander von Humboldt
Stiftung / Foundation

LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Outline: 1. Motivations

String phenomenology

Goal

Find "the" Standard Model from string theory.

Method:

- type II / heterotic strings, M-theory, F-theory: $D = 10, 11, 12$
- vacuum choice (flux compactification):
  - (typically) Calabi–Yau (CY) 3- or 4-fold
  - fluxes and intersecting branes

  $\rightarrow$ reduction to $D = 4$
- check consistency (tadpole, susy. . .)
- read the $D = 4$ QFT (gauge group, spectrum. . .)

String phenomenology

Goal

Find "the" Standard Model from string theory.

Method:

- type II / heterotic strings, M-theory, F-theory: $D = 10, 11, 12$
- vacuum choice (flux compactification):
  - (typically) Calabi–Yau (CY) 3- or 4-fold
  - fluxes and intersecting branes

  $\to$ reduction to $D = 4$
- check consistency (tadpole, susy...)
- read the $D = 4$ QFT (gauge group, spectrum...)

No vacuum selection mechanism $\Rightarrow$ string landscape

Landscape mapping

String phenomenology:

- find consistent string models
- find generic/common features
- reproduce the Standard Model

Landscape mapping

String phenomenology:

- find consistent string models
- find generic/common features
- reproduce the Standard Model

Typical challenges: properties and equations involving many integers

Types of data

Calabi–Yau (CY) manifolds

- CICY (complete intersection in products of projective spaces): 7890 (3-fold), 921,497 (4-fold)
- Kreuzer–Skarke (reflexive polyhedra): 473,800,776 ($d = 4$)

String and F-theory models involve huge numbers

- 10^{500}
- 10^{755}
- $10^{272,000}$
- ...

Types of data

Calabi–Yau (CY) manifolds

- CICY (complete intersection in products of projective spaces): 7890 (3-fold), 921,497 (4-fold)
- Kreuzer–Skarke (reflexive polyhedra): 473,800,776 ($d = 4$)

String and F-theory models involve huge numbers

- 10^{500}
- 10^{755}
- $10^{272,000}$
- ...

$\rightarrow$ use machine learning

Plan

Analysis of CICY 3-fold

- ML methodology
- results and discussions of Hodge numbers

In progress with: Vincent Lahoche, Mohamed El Amine Seddik, Mohamed Tamaazousti (List, Cea).

Outline: 2. Machine learning

Definition

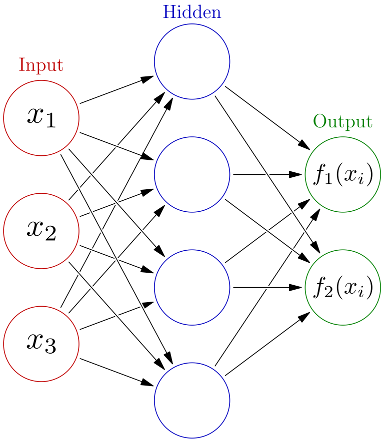
Machine learning (Samuel)

The field of study that gives computers the ability to learn without being explicitly programmed.

Machine learning (Mitchell)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

Deep neural network



Architecture:

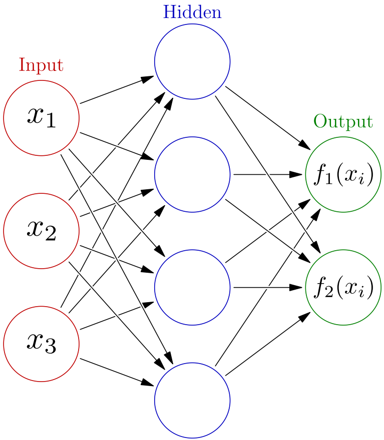
- 1–many hidden layers
- link: weighted input
- neuron: non-linear "activation function"

Summary: $x^{(n+1)} = g^{(n+1)}(W^{(n)}x^{(n)})$.
Generic method: fixed functions $g^{(n)}$, learn weights $W^{(n)}$

Deep neural network

$$x^{(1)}_{i_1} \equiv x_{i_1}$$
$$x^{(2)}_{i_2} = g^{(2)}(W^{(1)}_{i_2 i_1} x^{(1)}_{i_1})$$
$$f_{i_3}(x_{i_1}) \equiv x^{(3)}_{i_3} = g^{(3)}(W^{(2)}_{i_3 i_2} x^{(2)}_{i_2})$$

$i_1 = 1, 2, 3;\ i_2 = 1, \ldots, 4;\ i_3 = 1, 2$



Summary: $x^{(n+1)} = g^{(n+1)}(W^{(n)} x^{(n)})$.
Generic method: fixed functions $g^{(n)}$, learn weights $W^{(n)}$

Learning method

- define a loss function L

$$L = \sum_{i=1}^{N_{\text{train}}} \mathrm{distance}(y_i^{(\text{train})}, y_i^{(\text{pred})})$$

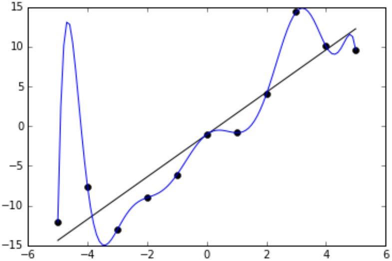
- minimize the loss function (iterated gradient descent...)

Learning method

- define a loss function L

$$L = \sum_{i=1}^{N_{\text{train}}} \mathrm{distance}(y_i^{(\text{train})}, y_i^{(\text{pred})})$$

- minimize the loss function (iterated gradient descent. . .)
- main risk: overfitting (= cannot generalize)
  $\rightarrow$ various solutions (regularization, dropout. . .)
  $\rightarrow$ split data set in two (training and test)



ML workflow

"Naive" workflow:

1. get raw data
2. write neural network with many layers
3. feed raw data to neural network
4. get nice results (or give up)

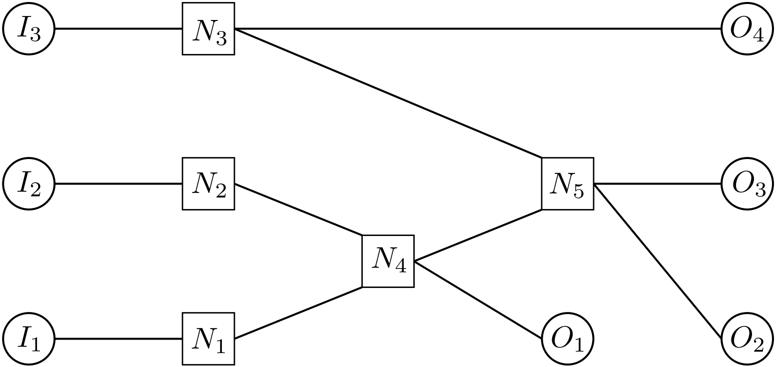
ML workflow

Real-world workflow:

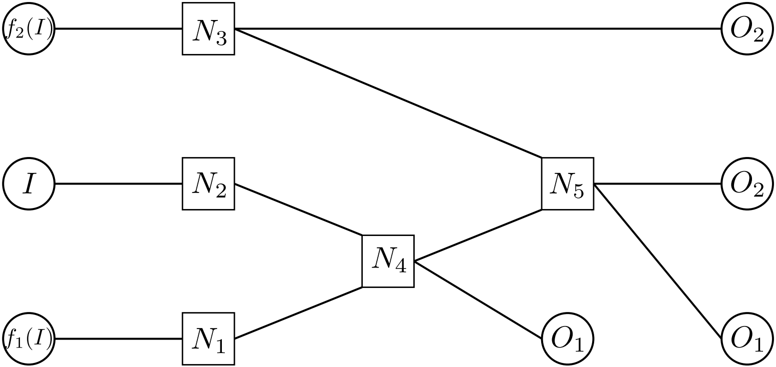
1. understand the problem
2. exploratory data analysis
    - feature engineering
    - feature selection
3. baseline model
    - full working pipeline
    - lower-bound on accuracy
4. validation strategy
5. machine learning model
6. ensembling

Pragmatic ref.:
`coursera.org/learn/competitive-data-science`

Complex neural network



Complex neural network



Particularities:

- $f_i(I)$: engineered features
- identical outputs (stabilisation)

Outline: 3. Calabi–Yau 3-folds

Calabi-Yau

Complete intersection Calabi–Yau (CICY) 3-fold:

- CY: complex manifold with vanishing first Chern class
- complete intersection: non-degenerate hypersurface in products of projective spaces
- hypersurface = solution to system of homogeneous polynomial equations

Calabi-Yau

Complete intersection Calabi–Yau (CICY) 3-fold:

- CY: complex manifold with vanishing first Chern class
- complete intersection: non-degenerate hypersurface in products of projective spaces
- hypersurface = solution to system of homogeneous polynomial equations
- described by configuration matrix $m \times k$

$$X = \left[\begin{array}{c|ccc} \mathbb{P}^{n_1} & a_1^1 & \cdots & a_k^1 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{P}^{n_m} & a_1^m & \cdots & a_k^m \end{array}\right]$$

$$\dim_{\mathbb{C}} X = \sum_{r=1}^{m} n_r - k = 3, \qquad n_r + 1 = \sum_{\alpha=1}^{k} a_\alpha^r$$

- a_α^r power of coordinates on $\mathbb{P}^{n_r}$ in αth equation

Configuration matrix

Examples

- quintic

$$\left[\ \mathbb{P}^4_x \ \middle|\ 5\ \right] \quad \Longrightarrow \quad \sum_a (X^a)^5 = 0$$

- 2 projective spaces, 3 equations

$$\left[\begin{array}{c|ccc} \mathbb{P}^3_x & 3 & 0 & 1 \\ \mathbb{P}^3_y & 0 & 3 & 1 \end{array}\right] \quad \Longrightarrow \quad \begin{cases} f_{abc}\, X^a X^b X^c = 0 \\ g_{\alpha\beta\gamma}\, Y^\alpha Y^\beta Y^\gamma = 0 \\ h_{a\alpha}\, X^a Y^\alpha = 0 \end{cases}$$

Configuration matrix

Examples

- quintic

$$\left[\ \mathbb{P}^4_x \ \middle|\ 5\ \right] \quad \Longrightarrow \quad \sum_a (X^a)^5 = 0$$

- 2 projective spaces, 3 equations

$$\left[\begin{array}{c|ccc} \mathbb{P}^3_x & 3 & 0 & 1 \\ \mathbb{P}^3_y & 0 & 3 & 1 \end{array}\right] \quad \Longrightarrow \quad \begin{cases} f_{abc}\, X^a X^b X^c = 0 \\ g_{\alpha\beta\gamma}\, Y^\alpha Y^\beta Y^\gamma = 0 \\ h_{a\alpha}\, X^a Y^\alpha = 0 \end{cases}$$

Classification

- invariances ($\to$ huge redundancy)
  - permutation of lines and columns
  - identities between subspaces
- but:
  - constraints $\Rightarrow$ bound on matrix size
  - $\exists$ "favourable" configuration

Topology

Why topology?

- no metric known for compact CY (cannot perform KK reduction explicitly)
- topological numbers $\rightarrow$ 4d properties (number of fields, representations, gauge symmetry. . .)

Topology

Why topology?

- no metric known for compact CY (cannot perform KK reduction explicitly)
- topological numbers $\rightarrow$ 4d properties (number of fields, representations, gauge symmetry. . .)

Topological properties

- Hodge numbers $h_{p,q}$ (number of harmonic (p,q)-forms) here: $h_{1,1}$, $h_{2,1}$
- Euler number $\chi = 2(h_{11} - h_{21})$
- Chern classes
- triple intersection numbers
- line bundle cohomologies

Topology

Why topology?

- no metric known for compact CY (cannot perform KK reduction explicitly)
- topological numbers → 4d properties (number of fields, representations, gauge symmetry. . .)

Topological properties

- Hodge numbers $h_{p,q}$ (number of harmonic (p, q)-forms) here: $h_{1,1}$, $h_{2,1}$
- Euler number $\chi = 2(h_{11} - h_{21})$
- Chern classes
- triple intersection numbers
- line bundle cohomologies

Datasets

CICY have been classified

- 7890 configurations (but $\exists$ redundancies)
- number of product spaces: 22
- $h_{1,1} \in [0, 19]$, $h_{2,1} \in [0, 101]$
- 266 combinations $(h_{1,1}, h_{2,1})$
- $a^r_\alpha \in [0, 5]$

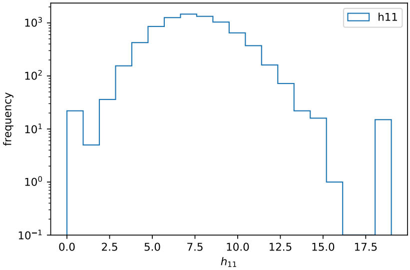
Original [Candelas-Dale-Lutken-Schimmrigk '88][Green-Hubsch-Lutken '89]

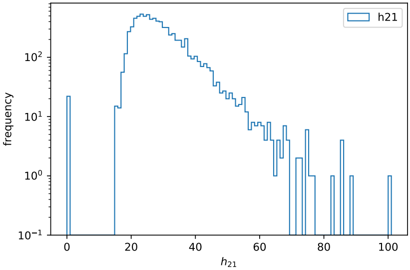
- maximal size: 12×15
- number of favourable matrices: 4874

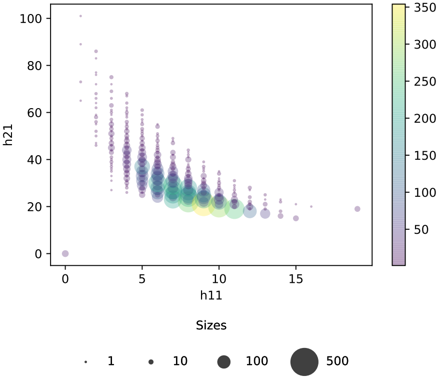
Favourable [1708.07907, Anderson-Gao-Gray-Lee]

- maximal size: 15×18
- number of favourable matrices: 7820

Data


h11
frequency
10^3
10^2
10^1
10^0
10^{-1}
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
h_{11}


h21
frequency
10^2
10^1
10^0
10^{-1}
0
20
40
60
80
100
h_{21}


h21
100
80
60
40
20
0
0
5
10
15
h11
350
300
250
200
150
100
50
Sizes
1
10
100
500

Goal and methodology

Philosophy

Start with the original dataset, derive everything else from configuration matrix and machine learning only.

Current goal

Input: configuration matrix $\longrightarrow$ Output: Hodge numbers

1. CICY: well studied, all topological quantities known
   $\rightarrow$ use as a sandbox
2. $h_{2,1}$: more difficult than $h_{1,1}$
   $\rightarrow$ prepare for studying CICY 4-folds
3. both original and favourable datasets

Continue the analysis from:
[1706.02714, He] [1806.03121, Bull-He-Jejjala-Mishra]

Outline: 4. Data analysis

Feature engineering

Process of creating new features derived from the raw input data.

Some examples:

- number of projective spaces (rows), $m = \texttt{num\_cp}$
- number of equations (columns), k
- number of $\mathbb{C}P^1$
- number of $\mathbb{C}P^2$
- number of $\mathbb{C}P^n$ with $n \neq 1$
- Frobenius norm of the matrix
- list of the projective space dimensions and statistics thereof (min, max, mean, median)
- K-nearest neighbour (KNN) clustering (with $K = 2, \ldots, 5$)

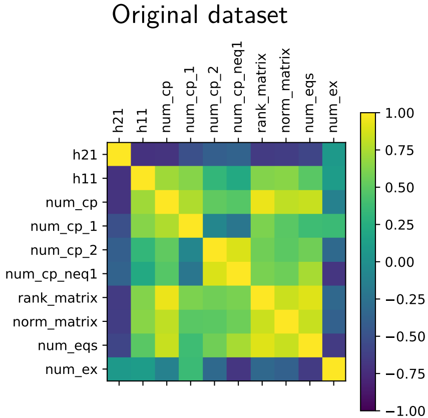
Feature selection

Select the most important features to draw attention of the ML algorithm to salient features in order to ease the learning.

Discovery methods:

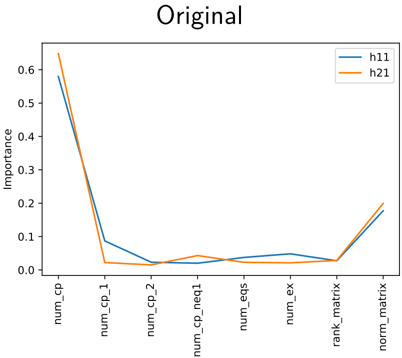
- correlation matrix
- random forests
- scatter plots
- trial and error
- etc.

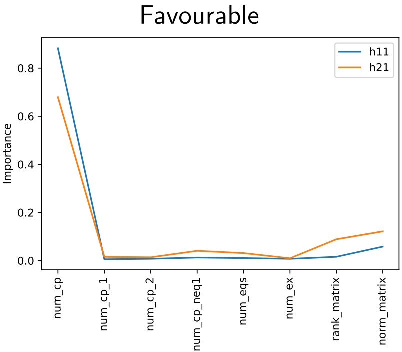
Correlation matrix



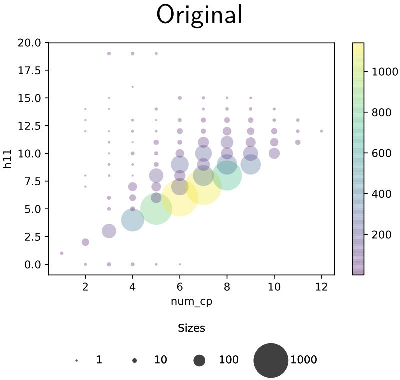
Random forest

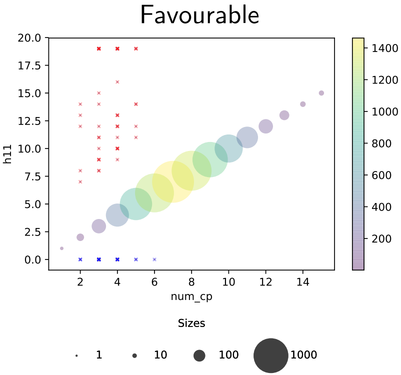
Large number of decision trees trained on different subsets and averaged on the outputs. The most relevant features appear at the top of the trees.
$\Longrightarrow$ classify feature importance



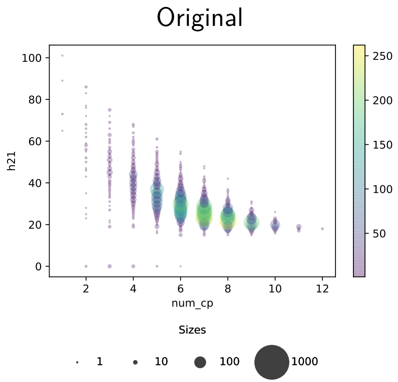


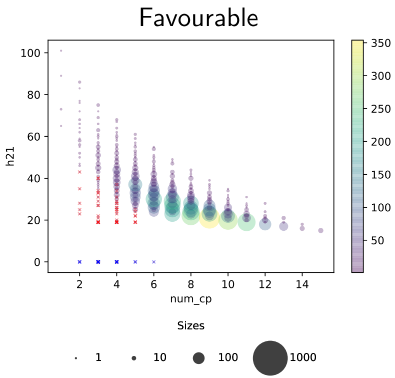
Scatter plots: $h_{1,1}$





Scatter plots: $h_{2,1}$





Outline: 5. ML analysis

Strategy

Questions:

- data diminution: remove outliers? (0.74%)
- data augmentation: use data invariance to generate more inputs?
- classification or regression?
- normalise inputs/outputs? (shift by mean, divide by variance)

Classification vs regression:

- classification: assume knowledge of boundaries
- regression: outputs of different size
  $\rightarrow$ normalize data $\approx$ use continuous variable

Regression: better for generalization

Algorithms

Possibilities (starting from original dataset):

- neural network with trivial architecture
  (matrix $\to$ hodges)
- neural network with non-trivial architecture
  (matrix + engineered features $\to$ hodges and tuned topology)
- boosting:
  1. linear regression: $h^{\text{lin}}_{p,q} = a \times \texttt{num\_cp} + b$
  2. neural network for $h_{p,q} - h^{\text{lin}}_{p,q}$
- other ensemble methods
  (average different ML models, train on different subsets...)
- convert dataset
  1. find favourable representation
  2. apply any method

Results (1)

Implementation and training

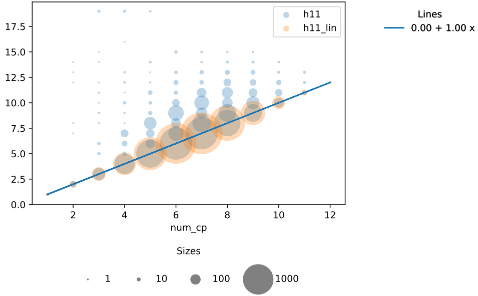
- sets: training (20%), test (80%)
- training time: few minutes

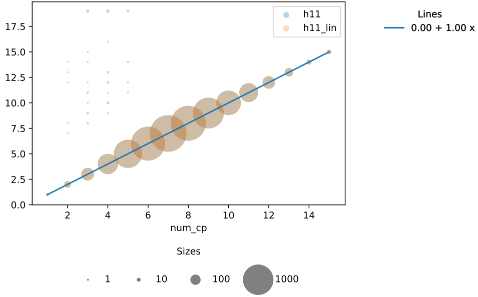
Accuracy:

- linear regression:
  - orig.: $h_{1,1} \approx 61\%$, $h_{2,1} \approx 8.5\%$
  - fav.: $h_{1,1} \approx 99.5\%$, $h_{2,1} \approx 4.5\%$

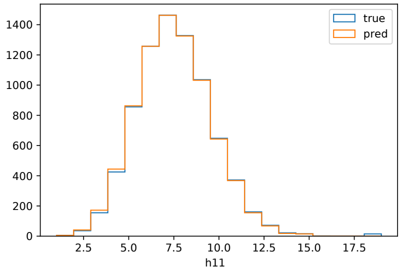
  (note: regression on several scalars $\to h_{2,1} \approx 12.5\%$)
- basic neural network (regression)
  - orig.: $h_{1,1} \approx 68\%$ (split: 30%), $\approx 78\%$ (split: 80%)
  - fav.: $h_{1,1} \approx 93\%$, $h_{2,1} \approx 16\%$
- boosting
  - orig.: $h_{1,1} \approx 72\%$, $h_{2,1} \approx 15\%$
  - fav.: $h_{1,1} \approx 99.5\%$, $h_{2,1} \approx 16\%$

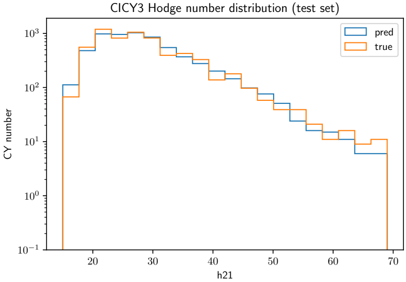
Results (2)


h11
h11_lin
Lines
0.00 + 1.00 x
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12
num_cp
Sizes
1
10
100
1000


h11
h11_lin
Lines
0.00 + 1.00 x
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12
14
num_cp
Sizes
1
10
100
1000

Results (3)





- Hodge numbers not exactly reproduced
- but distribution quite well learned
  (ex.: within $\pm 5\%$ error, $h_{2,1}$ is accurate more than 70%)

Discussion

In progress: test different architectures (multi-inputs, multi-tasks. . .)

Possible extensions:

- neural network performs very badly on $h_{2,1}$
  $\rightarrow$ challenge for ML community
- find a mapping original $\rightarrow$ favourable (GAN, cyclic GAN. . .)
- representation learning: find better / invariant representation (PCA, autoencoder. . .)

Outline: 6. Conclusion

Conclusion

- machine learning = extremely promising tool
- can help to learn how computer scientists / engineers work
- possible wide range of applications
- need to define clearly the (short- and long-term) objectives