Extension of the Dip-test Repertoire - Efficient and Differentiable p-value Calculation for Clustering

SIAM International Conference on Data Mining (SDM23)
April 27-29, 2023, Minneapolis, Minnesota, U.S.

Lena G. M. Bauer*[,1], Collin Leiber*[,2], Christian Böhm[1], Claudia Plant[1]

[1] Universität Wien, Vienna, Austria
[2] Ludwig-Maximilians-Universität, Munich, Germany

* Authors contributed equally

Outline

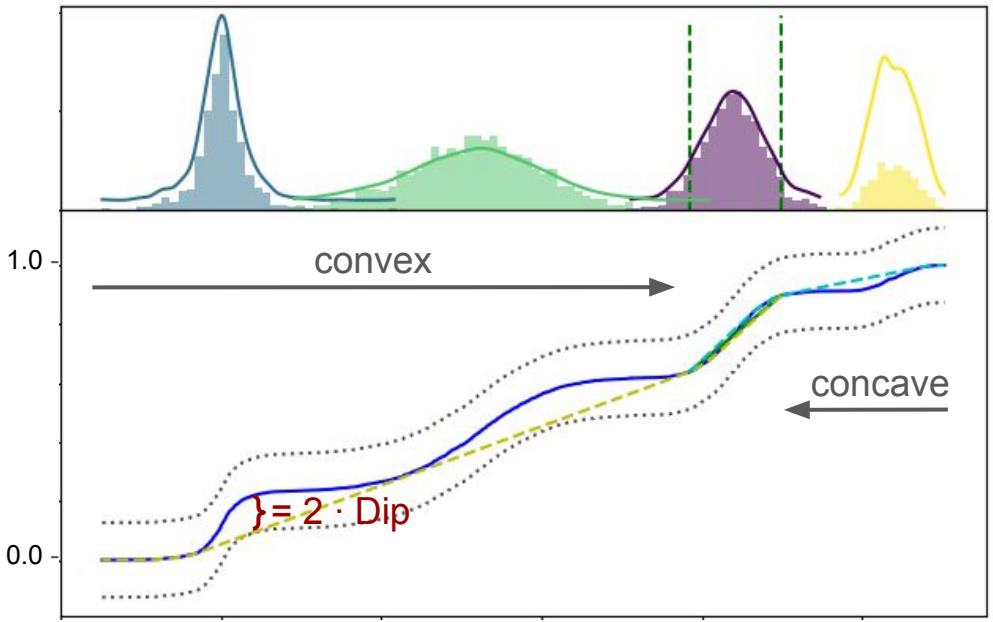
- The Dip-test and its *p*-value
- Function fit
- *p*-value experiments

- Gradient of the *p*-value
- The Dip’n’Sub algorithm
- Clustering experiments

- Conclusion

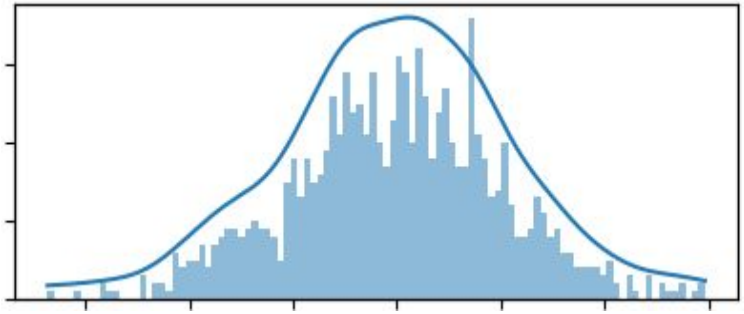
The Dip-test and its *p*-value (I)

- Measures the modality in one-dimensional samples
- Returns a Dip-value $D \in (0, 0.25]$
  - $D \approx 0 \rightarrow$ unimodal
  - $0 \ll D \leq 0.25 \rightarrow$ multimodal
- Is calculated by fitting a unimodal distribution onto the Empirical Cumulative Distribution Function (ECDF)

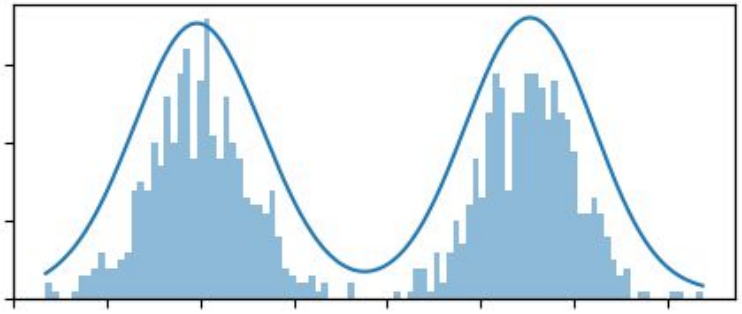


The Dip-test and its *p*-value (II)

- Used by various clustering algorithms to determine the number of clusters



D = 0.0096



D = 0.1103

- **But:** Both values are far from 0.25

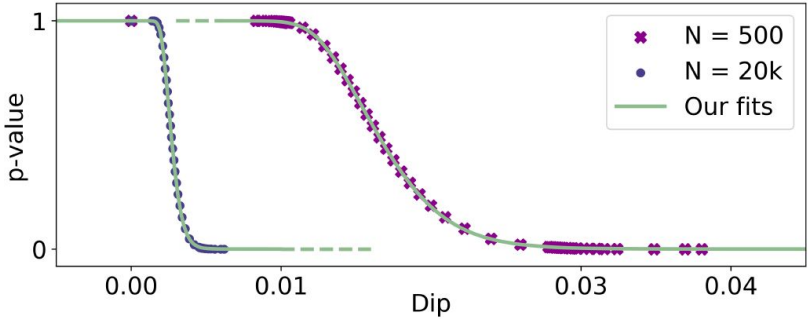
- How high must a Dip-value be to indicate a rather multimodal structure?
  → use *p*-value instead

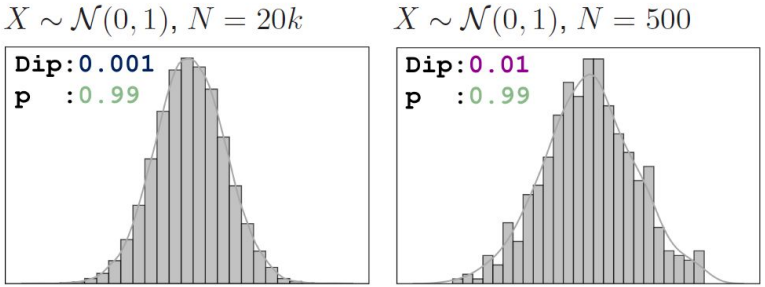
The Dip-test and its *p*-value (III)

- We can obtain a corresponding *p*-value for each Dip-value
- The transformation from Dip- to *p*-value is dependent on the sample size *N*

- Existing approaches use precomputed bootstrapped look-up tables
- Maximum number of samples in the most common table is 72,000

- **Our goal:** Designing a transformation function to obtain *p*-values for each combination of Dip-value and *N*

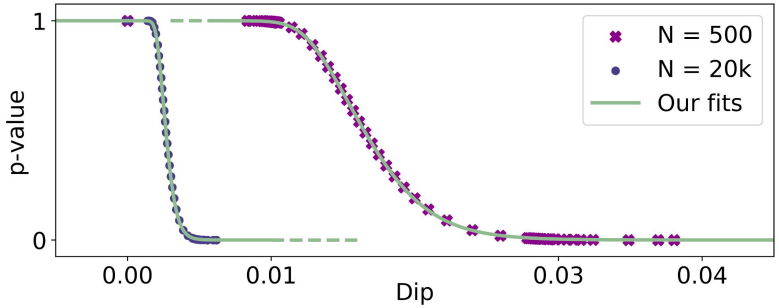




Function fit (I)

- We computed 307 (Dip, p)-pairs for 63 sample sizes by sampling from a uniform distribution with 100,000 repetitions

- To fit our transformation function we are guided by the generalised logistic function: $p(D) = d + \frac{a-d}{(c + h \cdot e^{-b \cdot D})^{\frac{1}{g}}}$



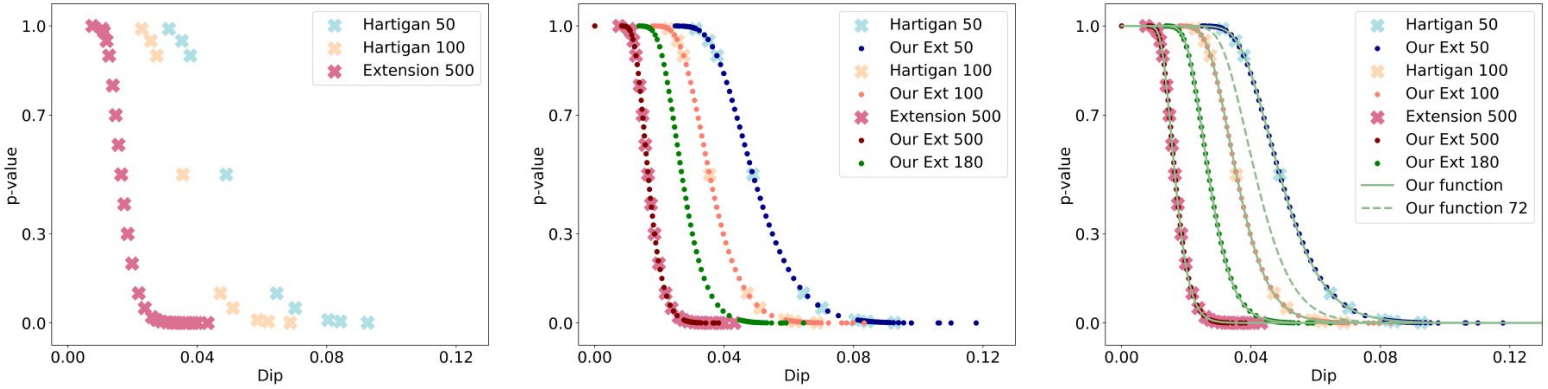
- Our final result:

$$\hat{p}(D, \hat{b}(N)) = 1 - \left[0.6 \cdot (1 + 1.6 \cdot e^{-\hat{b}(N) \cdot D + 6.5})^{1/1.6} + 0.4 \cdot (1 + 0.2 \cdot e^{-\hat{b}(N) \cdot D + 6.5})^{1/0.2}\right]^{-1}$$

with $\hat{b}(N) = 17.30784 \cdot \sqrt{N} + 12.04918$

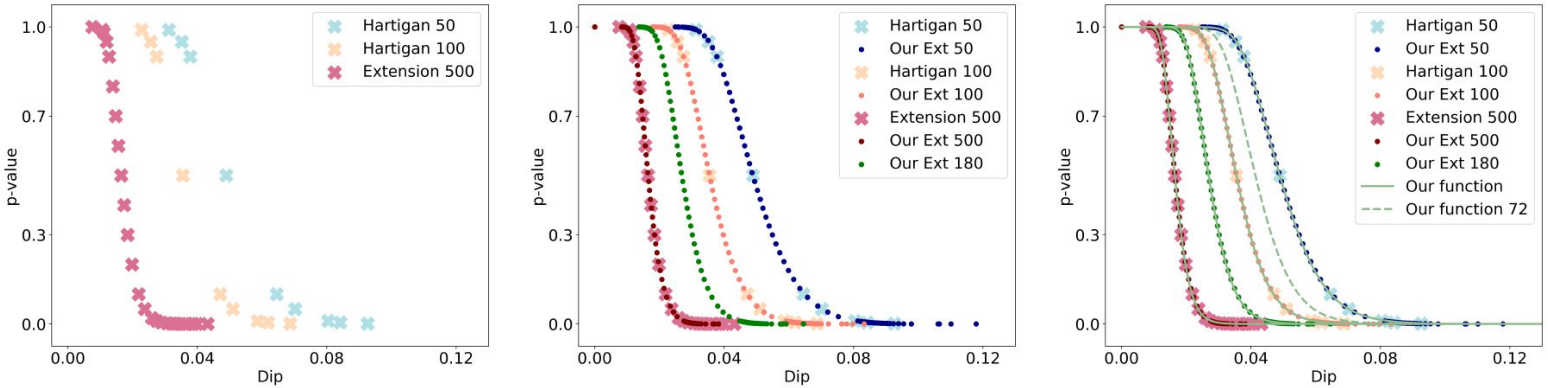
Function fit (II)

- Our transformation function is well-defined for all *N* and all Dip-values
- It allows a fast and precise calculation of *p*-values



Function fit (II)

- Our transformation function is well-defined for all *N* and all Dip-values
- It allows a fast and precise calculation of *p*-values



- Regarding the computed (Dip-p)-pairs we achieve a MSE of **3.43e^{-6}**
  → The original look-up table achieves a MSE of **7.92e^{-6}**
- Regarding previously unseen data we achieve a MSE of **3.14e^{-6}**
  → The original look-up table achieves a MSE of **8.12e^{-6}**

p-value experiments

- *p*-values for different unimodal (left) and multimodal (right) distributions
- All given values are averages for 100 random samples
- T = Look-up **t**able, F = **F**unction fit, B = **B**ootstrapped

unim. Distr.	Method	$N = 50$	$N = 234$	$N = 2345$	$N = 100k$	**multim. Distr.**	$N = 50$	$N = 234$	$N = 2345$	$N = 100k$
$\mathcal{N}(4,1)$	T	0.77 ± 0.24	$0.86 \pm 0.19^*$	$0.97 \pm 0.07^*$	†	$\mathcal{N}(4,1)$	8.94 ± 15.0	$0.06 \pm 0.23^*$	$0.00 \pm 0.00^*$	†
	F	0.77 ± 0.24	0.86 ± 0.19	0.97 ± 0.07	1.00 ± 0.02	$\cup$	8.83 ± 14.9	0.09 ± 0.25	0.00 ± 0.00	0.00 ± 0.00
	B	0.77 ± 0.24	0.86 ± 0.19	0.97 ± 0.07	1.00 ± 0.02	$\mathcal{N}(0,1)$	8.78 ± 15.0	0.06 ± 0.21	0.00 ± 0.00	0.00 ± 0.00
$\mathcal{T}_{nc}(4,2,0,1)$	T	0.80 ± 0.21	$0.89 \pm 0.14^*$	$0.98 \pm 0.03^*$	†	$\mathcal{T}_{nc}(4,2,0,1)$	0.79 ± 2.20	$0.00 \pm 0.00^*$	$0.00 \pm 0.00^*$	†
	F	0.80 ± 0.21	0.90 ± 0.14	0.98 ± 0.03	1.00 ± 0.00	$\cup$	0.83 ± 2.07	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	B	0.80 ± 0.21	0.90 ± 0.14	0.99 ± 0.03	1.00 ± 0.00	$\mathcal{T}_{nc}(4,2,7,1)$	0.74 ± 2.17	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
$\mathcal{L}(0,2)$	T	0.85 ± 0.19	$0.95 \pm 0.11^*$	$0.99 \pm 0.04^*$	†	$\mathcal{L}(0,2)$	24.1 ± 24.9	$2.36 \pm 7.87^*$	$0.00 \pm 0.00^*$	†
	F	0.85 ± 0.19	0.95 ± 0.11	0.99 ± 0.04	1.00 ± 0.00	$\cup$	23.9 ± 25.1	2.37 ± 7.82	0.00 ± 0.00	0.00 ± 0.00
	B	0.85 ± 0.19	0.95 ± 0.11	0.99 ± 0.04	1.00 ± 0.00	$\mathcal{L}(7,2)$	23.8 ± 24.8	2.34 ± 8.06	0.00 ± 0.00	0.00 ± 0.00

multiplied by 100

Gradient of the *p*-value

- Another advantage is the differentiability of our function

$$\nabla_\rho(\hat{p}(D,\hat{b}(N))) = \left(-\hat{b}(N)\nabla_\rho(D)\right)\cdot e^{-\hat{b}(N)\cdot D+6.5}\cdot\left[0.6\cdot(1+1.6e^{-\hat{b}(N)\cdot D+6.5})^{\frac{-0.6}{1.6}}+0.4\cdot(1+0.2e^{-\hat{b}(N)\cdot D+6.5})^{\frac{0.8}{0.2}}\right]$$
$$\cdot\left[0.6\cdot(1+1.6e^{-\hat{b}(N)\cdot D+6.5})^{\frac{1}{1.6}}+0.4\cdot(1+0.2e^{-\hat{b}(N)\cdot D+6.5})^{\frac{1}{0.2}}\right]^{-2}$$

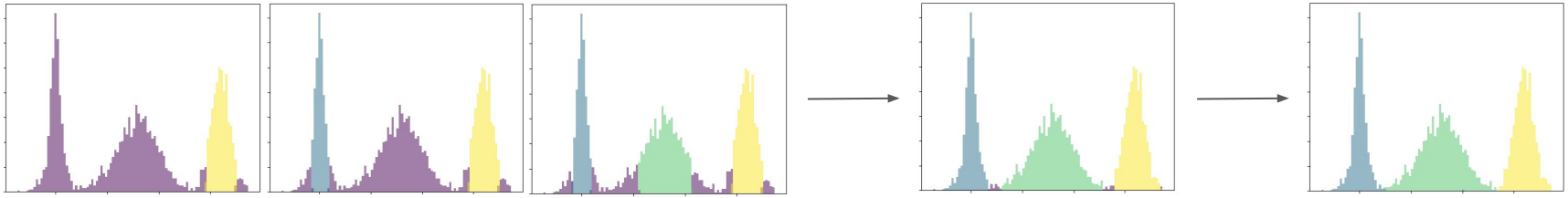
- This gradient can be utilized to identify cluster-friendly projection axes using stochastic gradient descent
- Those axes can form a lower-dimensional subspace of the given dataset
- In theory this gradient can also be used for Deep Learning applications

The Dip'n'Sub algorithm (I)

- Simultaneously defines the clusters and a matching subspace without knowing the number of clusters *k* in advance
- Iteratively identifies a new projection axis ρ that minimizes: $\frac{1}{N}\sum_{i=1}^{k}|C_i|\hat{p}\big(\text{Dip}(\overline{C_i^{\rho}}),\hat{b}(|C_i|)\big)$
- Check if more than *T*% of samples are contained in a multimodal cluster on ρ
- Split clusters on ρ using our subroutine *TailoredDip*

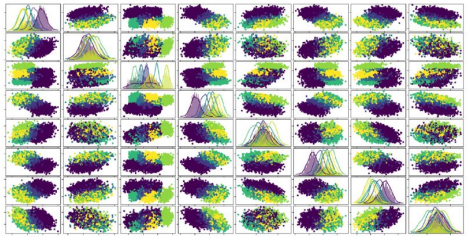
The Dip'n'Sub algorithm (I)

- Simultaneously defines the clusters and a matching subspace without knowing the number of clusters *k* in advance
- Iteratively identifies a new projection axis ρ that minimizes: $\frac{1}{N}\sum_{i=1}^{k}|C_i|\hat{p}\big(\text{Dip}(\overline{C_i^{\rho}}),\hat{b}(|C_i|)\big)$
- Check if more than *T*% of samples are contained in a multimodal cluster on ρ
- Split clusters on ρ using our subroutine *TailoredDip*



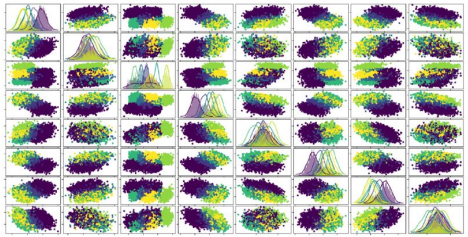
identify clusters add distribution tails assign outliers

The Dip’n’Sub algorithm (II)



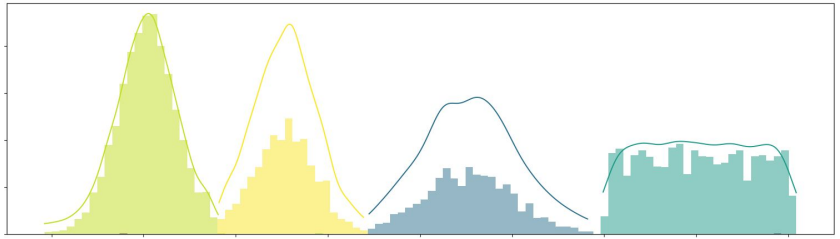
Input dataset

The Dip’n’Sub algorithm (II)

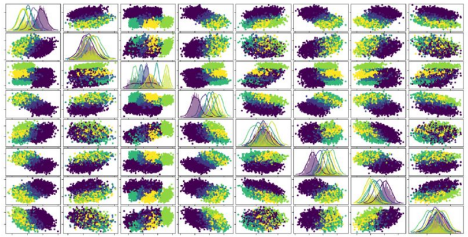


Input dataset

1st identified axis

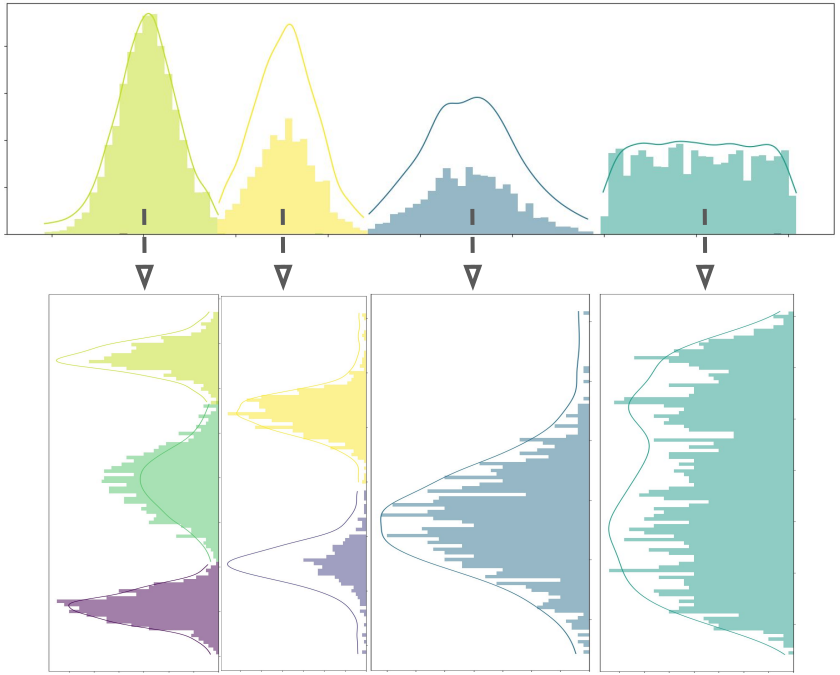


The Dip’n’Sub algorithm (II)



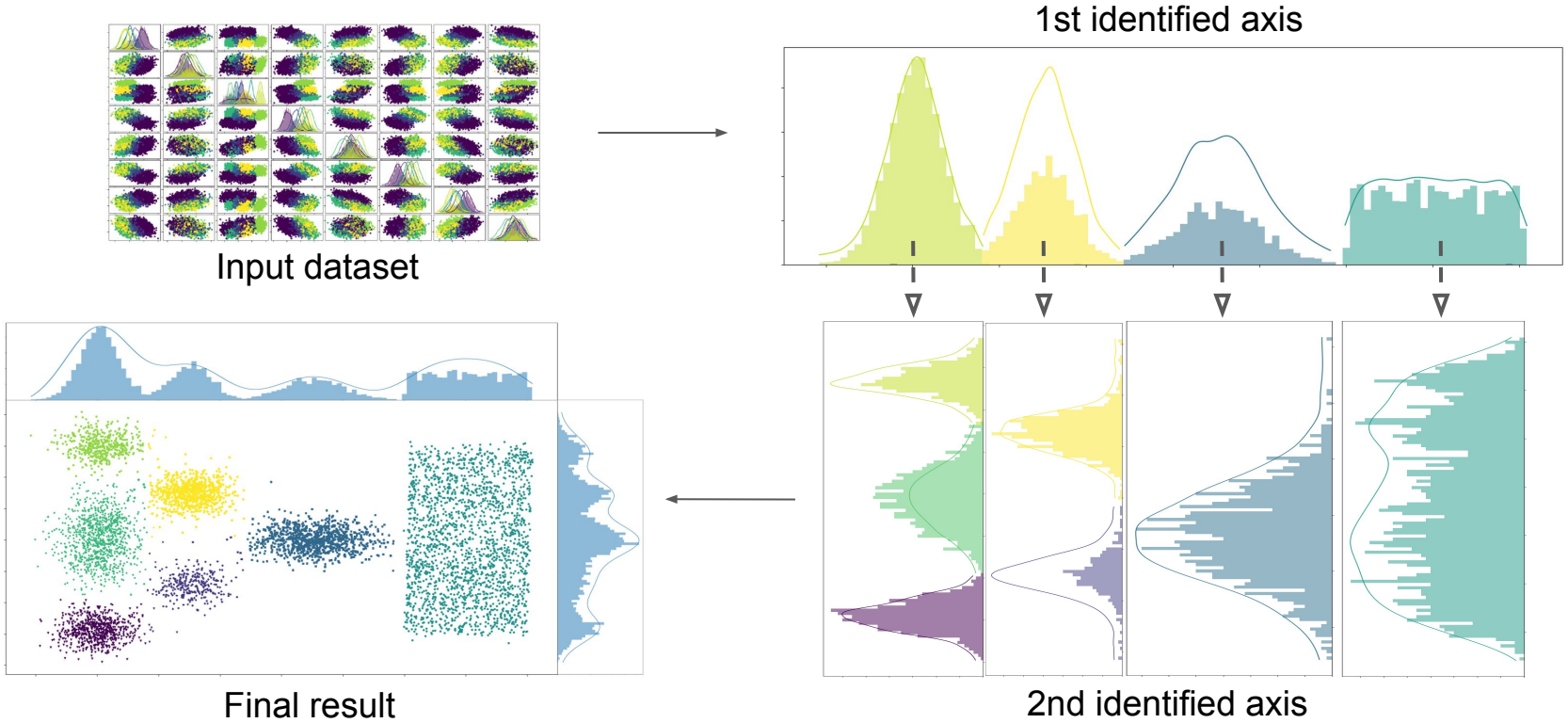
Input dataset

1st identified axis



2nd identified axis

The Dip’n’Sub algorithm (II)



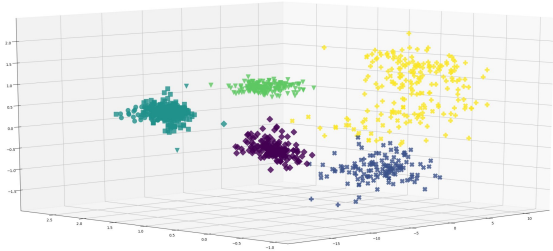
Clustering experiments (I)

- Maximum NMI results of different common subspace and Dip-based *k*-estimation algorithms after 10 runs

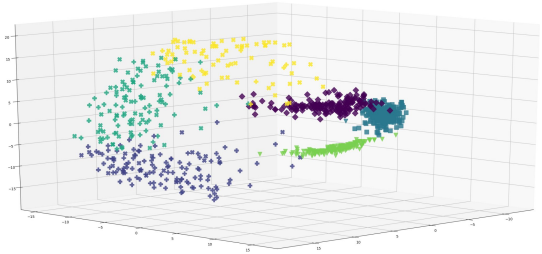
Dataset (k/d)	Common Subspace Algorithms							Dip-based k-estimation Algorithms		
	Dip'n'Sub	PCA+KM	ICA+KM	DipExt+KM	LDA-KM	SubKM	FOSSCLU	DipMeans	p. DipMeans	SkinnyDip
SYNTH (7/8)	**0.97** (7/2)	0.87 (7/4)	0.38 (7/7)	0.69 (7/1)	0.88 (7/6)	0.87 (7/6)	0.90 (7/5)	0.64 (3/8)	0.92 (6/8)	0.16 (2/8)
BANK (2/4)	0.41 (7/3)	0.03 (2/2)	0.01 (2/2)	**0.83** (2/1)	0.01 (2/1)	0.03 (2/1)	0.01 (2/4)	0.32 (41/4)	0.32 (36/4)	0.13 (2/4)
USER (4/5)	0.52 (10/1)	0.43 (4/5)	0.03 (4/4)	0.40 (4/2)	0.49 (4/3)	0.46 (4/3)	**0.65** (4/2)	0.00 (1/5)	0.36 (33/5)	0.15 (2/5)
HTRU2 (2/8)	**0.38** (3/2)	0.03 (2/2)	0.30 (2/2)	0.03 (2/1)	0.28 (2/1)	0.03 (2/1)	0.32 (2/5)	0.00 (1/66)	0.18 (7/66)	0.08 (1/66)
ALOI (4/66)	0.98 (4/1)	**1.00** (4/35)	**1.00** (4/4)	**1.00** (4/5)	**1.00** (4/3)	**1.00** (4/3)	†	0.96 (5/66)	0.52 (69/66)	0.17 (16/66)
MICE (8/68)	0.55 (6/3)	0.27 (8/6)	0.39 (8/8)	**0.59** (8/3)	†	0.29 (8/7)	0.33 (8/5)	0.00 (1/68)	0.54 (278/68)	0.00 (1/68)
AIBO (2/70)	**0.68** (2/1)	**0.68** (2/20)	0.56 (2/2)	0.60 (2/3)	0.30 (2/1)	**0.68** (2/1)	0.52 (2/5)	0.00 (1/70)	0.34 (35/70)	0.02 (3/70)
MOTE (2/84)	0.37 (2/1)	0.30 (2/42)	0.37 (2/2)	**0.41** (2/2)	0.28 (2/1)	0.35 (2/1)	0.08 (2/5)	0.36 (3/84)	0.00 (1/84)	0.00 (1/84)
SYMB (6/398)	**0.84** (5/3)	0.80 (6/6)	0.79 (6/6)	0.65 (6/2)	0.80 (6/5)	0.80 (6/5)	†	0.82 (5/398)	0.74 (16/398)	0.02 (2/398)
OLIVE (4/570)	0.57 (4/2)	0.68 (4/4)	0.69 (4/4)	0.73 (4/58)	†	**0.75** (4/3)	0.16 (4/3)	0.50 (2/570)	0.68 (9/570)	0.04 (1/570)

Clustering experiments (II)

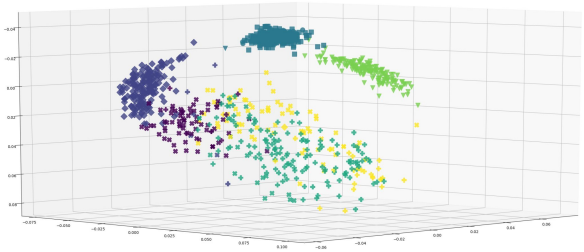
Dataset (k/d)	Common Subspace Algorithms							Dip-based k-estimation Algorithms		
	Dip'n'Sub	PCA+KM	ICA+KM	DipExt+KM	LDA-KM	SubKM	FOSSCLU	DipMeans	p. DipMeans	SkinnyDip
SYMB (6/398)	**0.84** (5/3)	0.80 (6/6)	0.79 (6/6)	0.65 (6/2)	0.80 (6/5)	0.80 (6/5)	†	0.82 (5/398)	0.74 (16/398)	0.02 (2/398)



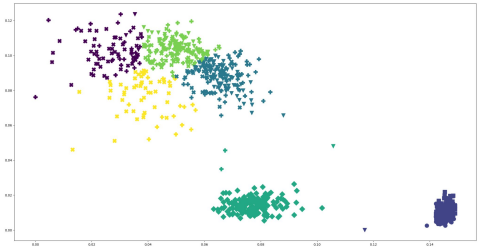
Dip'n'Sub (ACC=0.79)



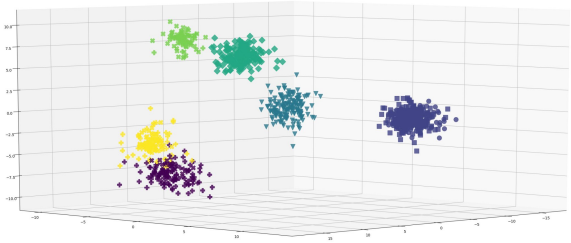
PCA+KM (ACC=0.69)



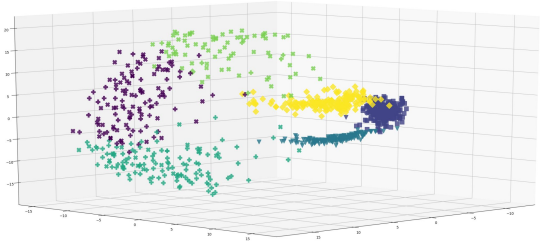
ICA+KM (ACC=0.71)



DipExt+KM (ACC=0.57)



LDA-KM (ACC=0.69)



SubKM (ACC=0.69)

Conclusion

- We created a differentiable function to translate Dip- to *p*-values
  → Automatic and fast transformation for any sample size
- Its error regarding MSE is lower than previously used look-up tables
- The gradient of our function can be useful for the Data Mining community as shown by our Dip'n'Sub clustering algorithm
- Experiments show that Dip'n'Sub produces high-quality clustering results while simultaneously defining a lower-dimensional subspace

Thank you for your attention!

Contact: leiber@dbs.ifi.lmu.de

Algorithm

Algorithm 1: The Dip'n'Sub algorithm

Input: data set X, significance α, threshold T
Output: labels

```
k = 1; labels = [0, ..., 0]; X_fin = []
while True do
    s = 1; ρ = 0⃗
    Q = log(d) features with lowest weighted
      p-values ∪ first log(d) components of PCA
    for each ρ_tmp ∈ Q do
        Update ρ_tmp with SGD using Eq. 3.1
        s_tmp = value of Eq. 3.1 using ρ_tmp
        if s_tmp < s then
            s = s_tmp; ρ = ρ_tmp
    P = {p-value(Dip(C̄_i^ρ), |C̄_i^ρ|) | i ∈ [1, k]}
    if sum{|C_i| | i∈[1,k] ∧ P_i<α} / N ≥ T then
        for each cluster i with P_i < α do
            labels_new = TailoredDip(C̄_i^ρ, α)
            update labels using labels_new
        X_fin = combine X_fin and {ρ^T x | x ∈ X}
        X = keep features orthogonal to ρ
    else
        break
return labels, X_fin
```

Eq. 3.1:

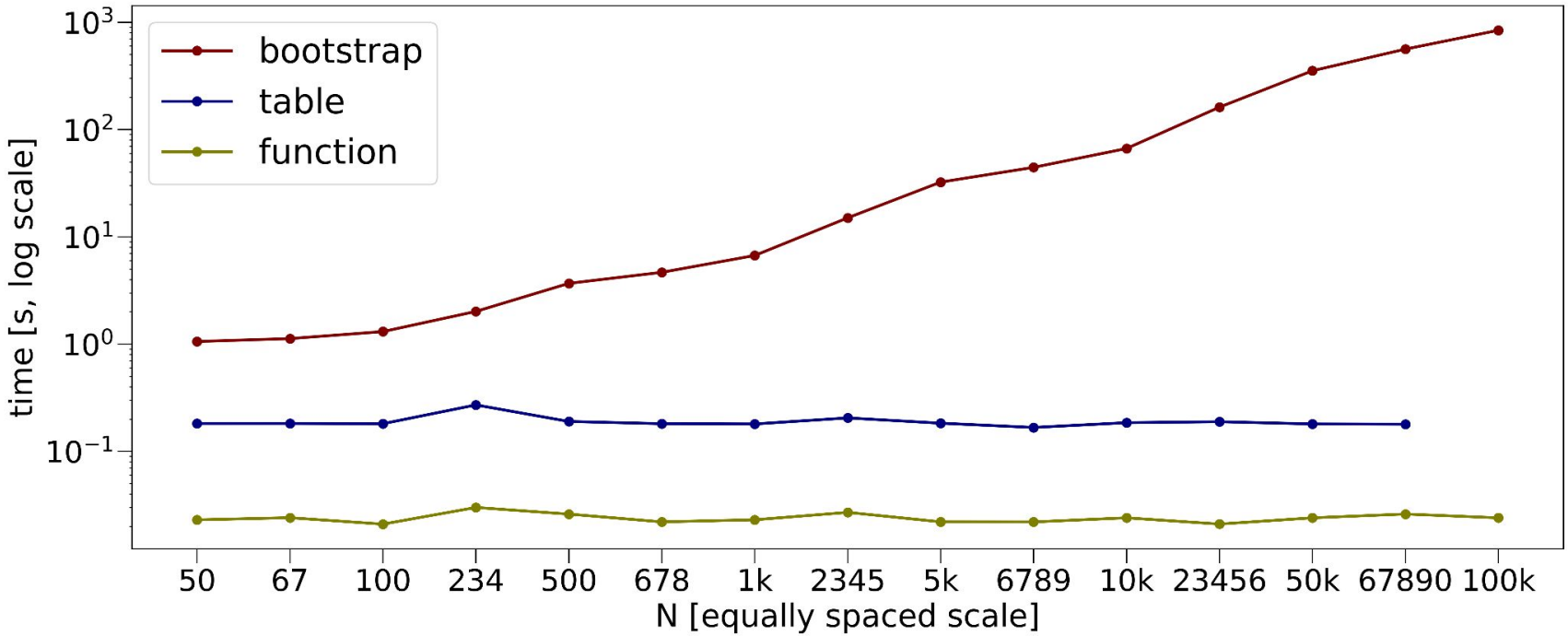
$$\frac{1}{N}\sum_{i=1}^{k}|C_i|\hat{p}\big(\mathrm{Dip}(\overline{C_i^{\rho}}),\hat{b}(|C_i|)\big)$$

p-value experiments using clustering algorithms

- Average NMI and runtime (RT) results for different Dip-based clustering methods after 10 runs

Dataset	DipMeans						p. DipMeans						SkinnyDip					
	NMI			RT			NMI			RT			NMI			RT		
	T	F	B	T	F	B	T	F	B	T	F	B	T	F	B	T	F	B
SYNTH	0.64	0.64	0.64	6.87	5.82	8.92	0.86	0.86	0.85	0.76	0.74	9.03	0.16	0.16	0.16	0.02	0.01	14.66
BANK	0.31	0.31	0.30	13.00	6.93	43.71	0.30	0.31	0.30	7.70	5.46	69.81	0.13	0.13	0.13	0.01	0.00	2.56
USER	0.00	0.00	0.00	0.05	0.02	0.08	0.34	0.34	0.35	1.54	1.13	11.37	0.15	0.15	0.15	0.00	0.00	0.68
HTRU2	0.00	0.00	0.00	28.87	28.32	29.03	0.17	0.17	0.17	2.20	2.18	19.05	0.08	0.08	0.08	0.04	0.04	40.78
ALOI	0.96	0.96	0.92	0.43	0.22	0.84	0.49	0.51	0.49	9.03	6.68	54.36	0.17	0.15	0.17	0.06	0.04	3.68
MICE	0.00	0.00	0.00	0.19	0.12	0.27	0.53	0.54	0.53	91.47	50.21	607.21	0.00	0.00	0.00	0.01	0.01	3.76
AIBO	0.00	0.00	0.00	0.09	0.04	0.14	0.28	0.33	0.27	20.02	5.33	129.60	0.02	0.02	0.02	0.02	0.01	3.29
MOTE	0.35	0.35	0.35	0.85	0.62	1.21	0.00	0.00	0.00	0.03	0.02	0.37	0.00	0.00	0.00	0.03	0.02	8.73
SYMB	0.82	0.82	0.82	1.50	1.15	2.23	0.70	0.74	0.70	36.70	5.64	73.69	0.02	0.02	0.02	0.11	0.10	5.16
OLIVE	0.50	0.50	0.50	0.06	0.04	0.16	0.64	0.52	0.64	1.26	0.15	3.08	0.04	0.04	0.04	0.03	0.02	0.39

Dataset information

Dataset	N	d	k
SYNTH	6,300	8	7
BANK	1,372	4	2
USER	403	5	4
HTRU2	17,898	8	2
ALOI	288	66	4
MICE	1,077	68	8
AIBO	621	70	2
MOTE	1,272	84	2
SYMB	1,020	398	6
OLIVE	60	570	4

Runtime experiments