Unüberwachte Nächste Nachbarn

Ein effizientes Verfahren zur Dimensionsreduktion

Oliver Kramer

Department für Informatik
Carl von Ossietzky Universität Oldenburg

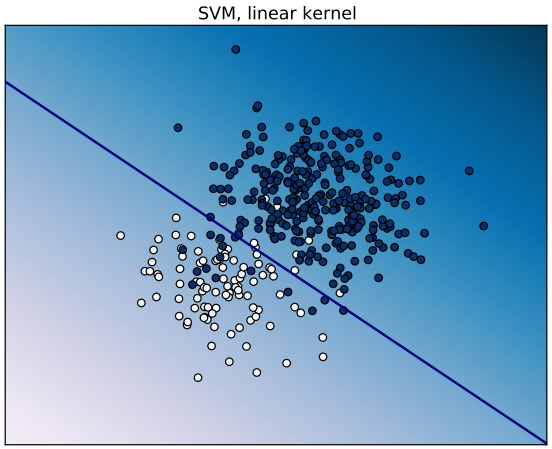
4. Mai 2012

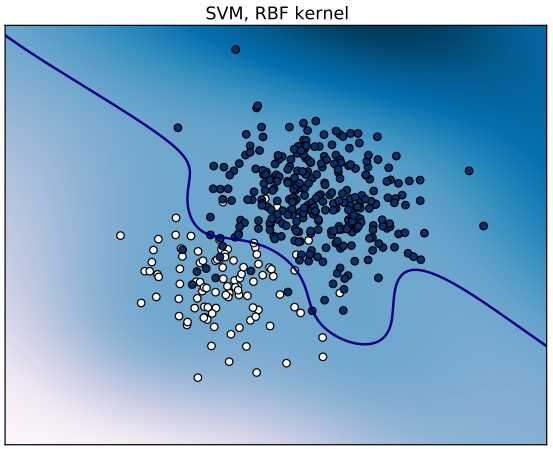
Übersicht

Maschinelles Lernen

Was ist maschinelles Lernen?

- Typische Fragestellung: geg. eine Menge von Mustern $\mathbf{x}_1, \ldots, \mathbf{x}_N$ mit Labeln (Klassen oder numerische Werte) $y_1, \ldots, y_N$
- Suche nach Vorhersagemodell f, so dass für unbekanntes x das 'richtige' y vorhergesagt wird
- empirisches Risiko minimieren
- K-Nächste Nachbarn, Support-Vektor-Maschinen





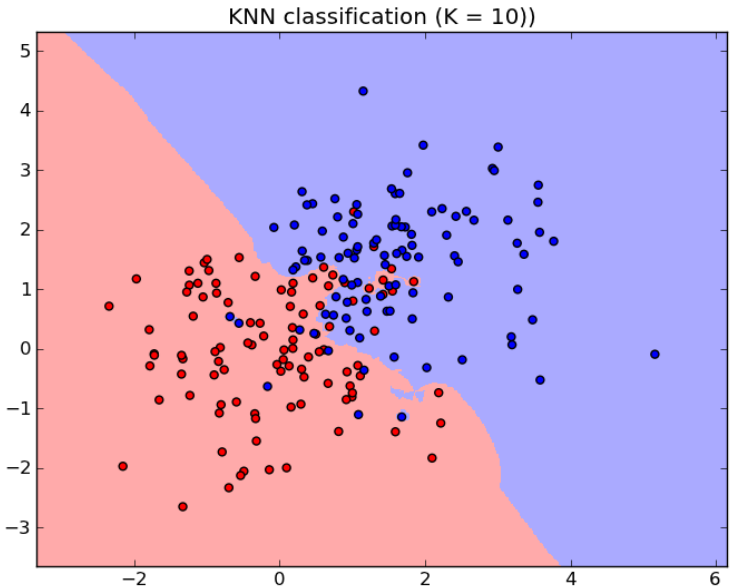
Nächste Nachbarn

K-Nächste Nachbarn

- Für unbekanntes Muster $\mathbf{x}'$ wähle Label der K nächsten Muster im Datenraum
- Distanz im Datenraum, z.B. $d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} (x_i - y_i)^2 \right)^{1/2}$

Klassifikation ($y \in \{-1, 1\}$)

$$f(\mathbf{x}') := \begin{cases} 1 & \text{falls } \sum_{i \in \mathcal{N}_K(\mathbf{x}')} y_i > 0 \\ -1 & \text{falls } \sum_{i \in \mathcal{N}_K(\mathbf{x}')} y_i < 0, \end{cases}$$



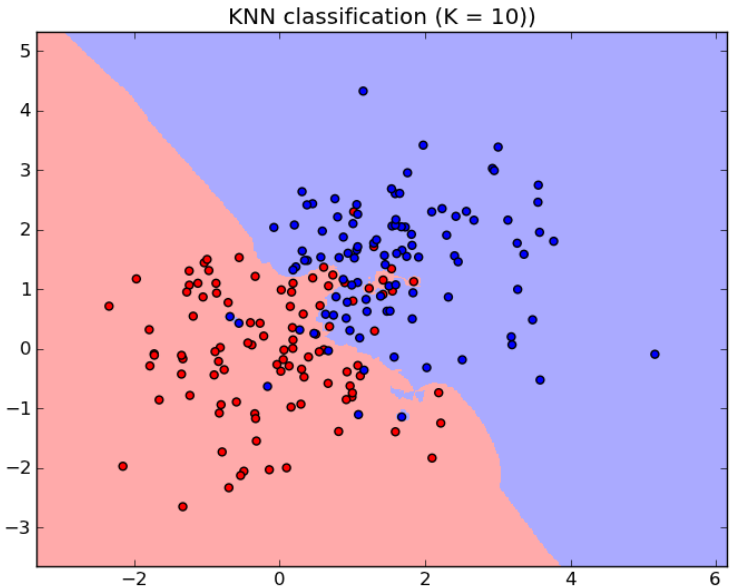
Nächste Nachbarn

K-Nächste Nachbarn

- Für unbekanntes Muster $\mathbf{x}'$ wähle Label der K nächsten Muster im Datenraum
- Distanz im Datenraum, z.B. $d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} (x_i - y_i)^2 \right)^{1/2}$

Regression ($\mathbf{y} \in \mathbb{R}^d$)

$$\mathbf{f}(\mathbf{x}') := \frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x}')} \mathbf{y}_i$$



Dimensionsreduktion

Dimensionsreduktion

- Warum Dimensionsreduktion?
- Antwort: Vorverarbeitung, Visualisierung
- Finde Abbildungs $\mathbf{F} : \mathbf{y} \rightarrow \mathbf{x}$ für Muster $\mathbf{y} \in \mathbf{Y} \subset \mathbb{R}^d$ und *latente* Punkte $\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^q$ mit $d > q$.
- Bilde mit $\mathbf{X}$ Topologie (Abstände, Nachbarschaften) des Datenraums ab
- Methoden: PCA, UKR, ISOMAP, LLE

Unüberwachte Regression

Unüberwachte Regression

- Sei $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ with $\mathbf{y}_i \in \mathbb{R}^d$ Matrix von Mustern
- Gesucht ist $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$
- Idee: Finde $\mathbf{X}$ durch Umkehr eines Regressions-Modelles $\mathbf{f}_{KNN}(\mathbf{X}) = [\mathbf{f}_{KNN}(\mathbf{x}_i, \mathbf{X})]_{i=1}^N]$
- Minimiere den *Datenraumrekonstruktions-Fehler* (DSRE)

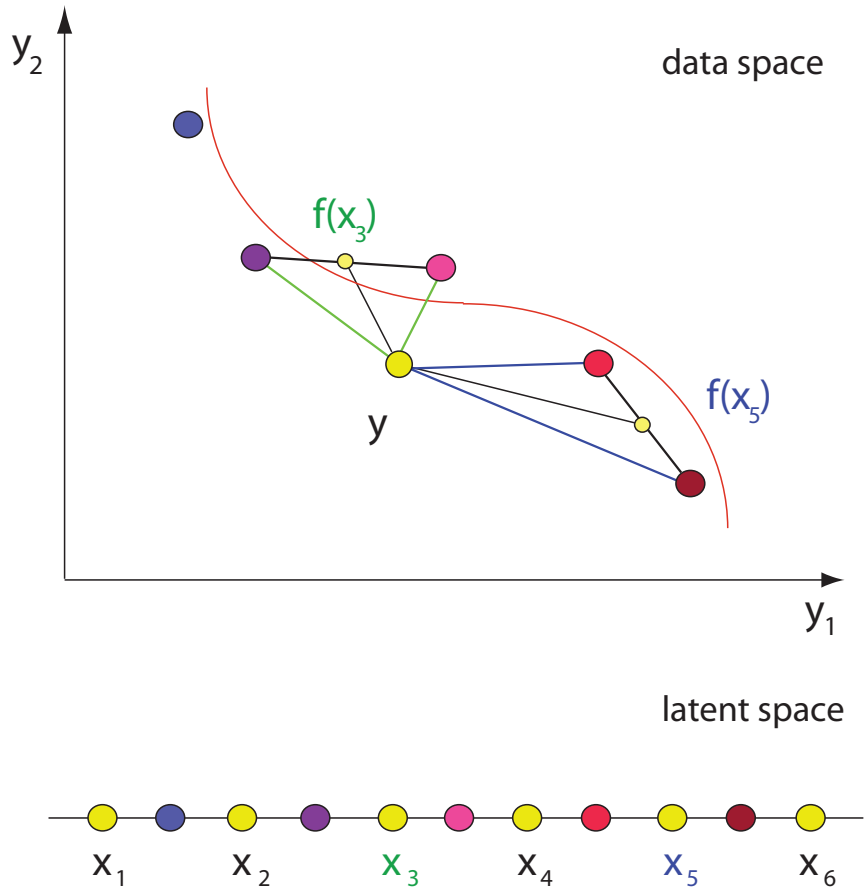
$$\text{minimiere } E(\mathbf{X}) = \frac{1}{N} \|\mathbf{Y} - \mathbf{f}(\mathbf{X})\|_F^2. \tag{1}$$

- Optimierung: Initialisierung durch Spektralmethoden (LLE), Gradientenabstieg in $\mathbb{R}^q$

Unüberwachte Nächste Nachbarn

UNN: UR mit KNN

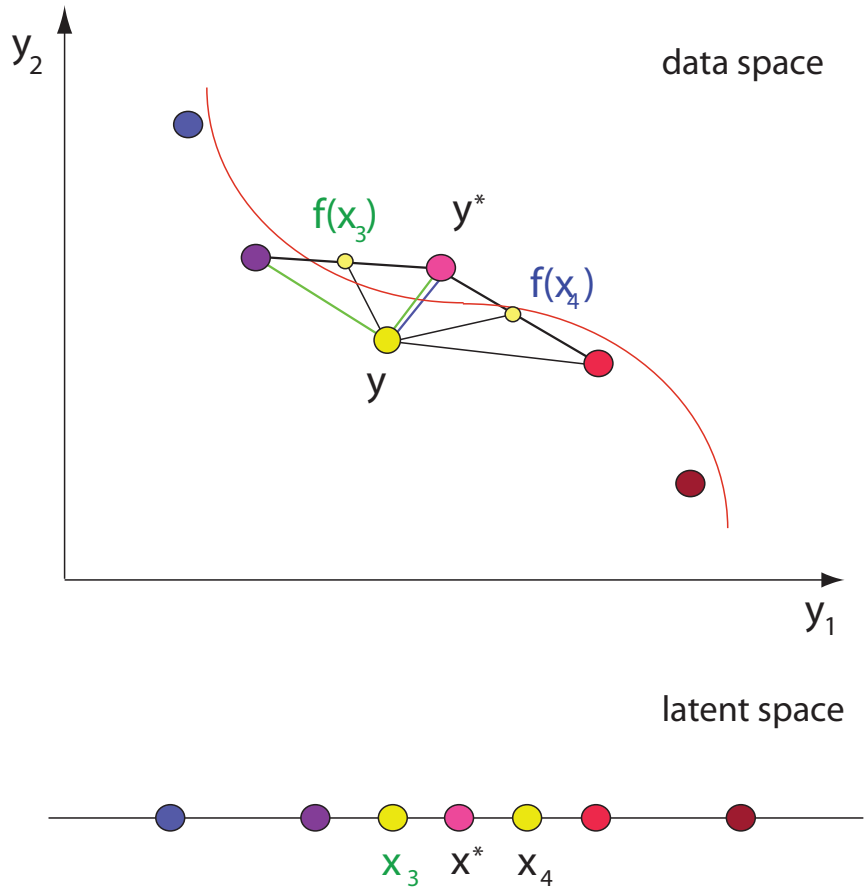
- **Idee 1**: Verwende KNN zur Regression
  - KNN gut in niedrigen Dimensionen $(q = 1, 2, 3)$, viele Daten
- **Idee 2**: Konstruiere Lösung *iterativ*
  - Bsp.: q $= 1$
  - Teste jeden Zwischenraum *oder* Nachbarn der nächsten Einbettung
  - Wähle Position mit geringstem DSRE



Unüberwachte Nächste Nachbarn

UNN: UR mit KNN

- **Idee 1**: Verwende KNN zur Regression
  - KNN gut in niedrigen Dimensionen $(q = 1, 2, 3)$, viele Daten
- **Idee 2**: Konstruiere Lösung *iterativ*
  - Bsp.: q $= 1$
  - Teste jeden Zwischenraum *oder* Nachbarn der nächsten Einbettung
  - Wähle Position mit geringstem DSRE



Unüberwachte Nächste Nachbarn (ii)

3-dimensionales S

- Einbettung von 500 Datenpunkten
- Farbe definiert Position im latenten Raum (ähnliche Farben = benachbarte Positionen)

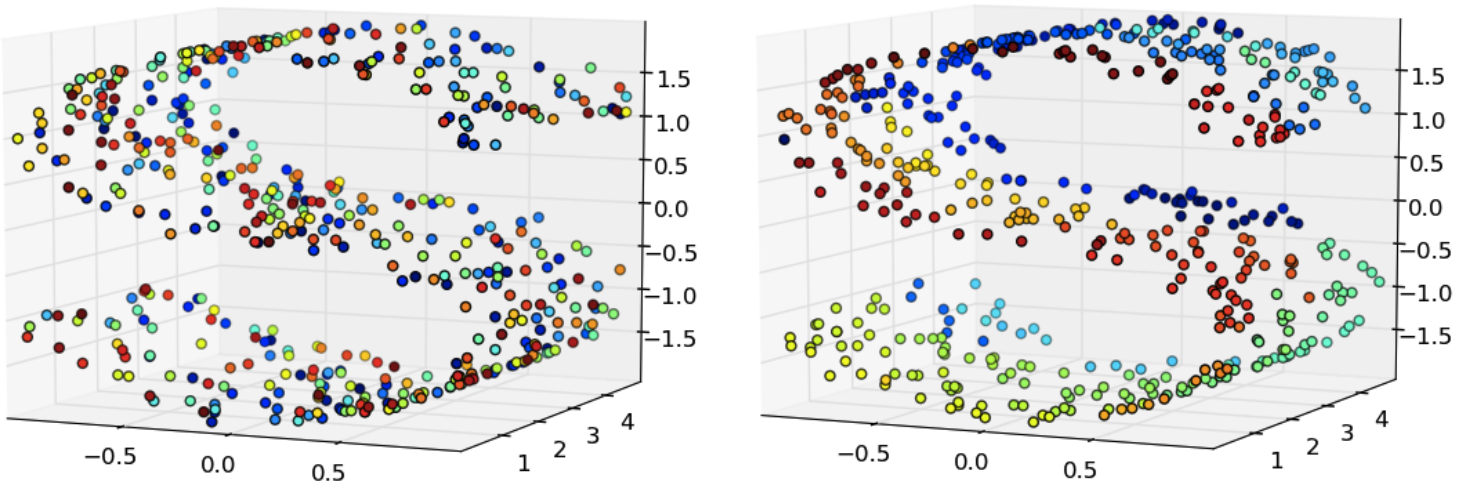


Abbildung : Einbettung des $3D - S$. Links vor der Einbettung, rechts: Einbettung durch UNN, Aus: *Kramer: Dimensionality Reduction by Unsupervised K-Nearest Neighbor Regression, ICMLA 2011.*

UNN - Vergleich

	2D-S			3D-S		
K	2	5	10	2	5	10
init	201.6	290.0	309.2	691.3	904.5	945.80
UNN	**19.6**	**27.1**	66.3	**101.9**	**126.7**	**263.39**
UNN$_g$	29.2	70.1	64.7	140.4	244.4	296.5
LLE	25.5	37.7	**40.6**	135.0	514.3	583.6

	3D-S$_h$			digits (7)		
K	2	5	10	2	5	10
init	577.0	727.6	810.7	196.6	248.2	265.2
UNN	**80.7**	**108.1**	**216.4**	**139.0**	**179.3**	**216.6**
UNN$_g$	101.8	204.4	346.8	145.3	195.4	222.1
LLE	94.9	198.9	387.4	147.8	198.1	217.8

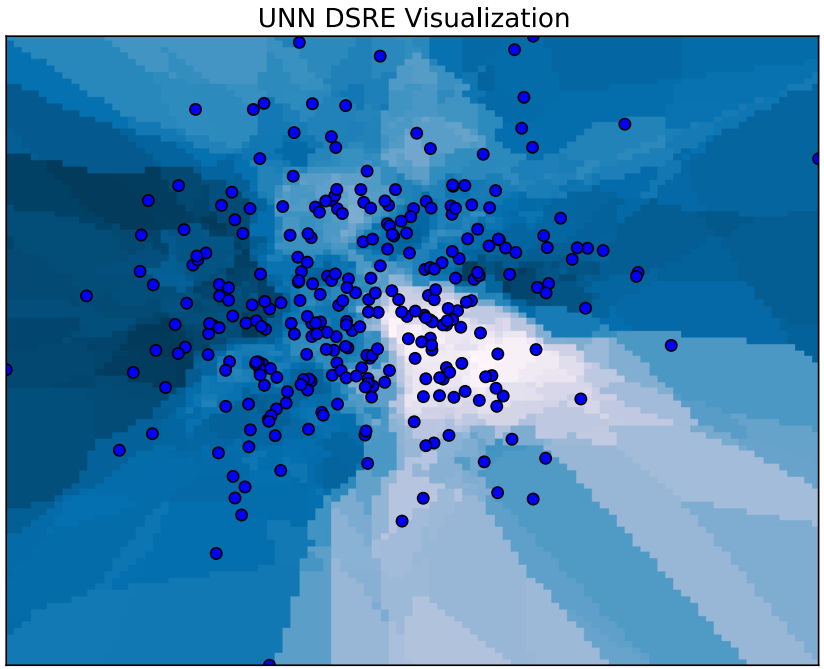
Tabelle : DSRE-Vergleich: initial, UNN, UNN$_g$ und Locally Linear Embedding

Übersicht

Warum Evolution?

Motivation

- Fitnesslandschaft ist multimodal
- Keine UNN Ableitung berechenbar
- Blackbox-Optimierungsverfahren!



Evolutionäre Algorithmen

Kontinuierlicher Ansatz

- Lösungskandidat $\mathbf{X} \in \mathbb{R}^{q \times N}$ ist Matrix latenter Vektoren $\mathbf{x} \in \mathbb{R}^q$
- Fitnessberechnung $f(\mathbf{X})$: eine komplette DSRE Berechnung
- Verwende CMA-ES (Gaussian-basierte Evolutionsstrategie ES mit Kovarianzmatrix-Adaptation)
- Regularisierung zur Vermeidung von Overfitting

Kombinatorischer Ansatz

- Anzahl möglicher K-Nachbarschaften ist beschränkt: $\binom{N}{K}$
- Platziere Lösungen auf Grid
- Verwende (1+1)-EA zur Evaluierung latenter Positionen auf Grid

Regularisierung

Restriktion auf $\mathbf{x} \in [0,1]^q$

- Beschränke latente Positionen auf Einheitswürfel $\mathbf{x} \in [0,1]^q$:

$$\min E(\mathbf{X})_r \text{ subject to } x_{ij} \in [0,1] \tag{2}$$

mit quadratischer Straffunktion:

$$p(\mathbf{X}) = \sum_{i,j} \epsilon_{ij} \text{ mit } \epsilon = \begin{cases} (x_{ij}-1)^2 & \text{if } x_{ij} > 1 \\ x_{ij}^2 & \text{if } x_{ij} < 0 \\ 0 & \text{else} \end{cases} . \tag{3}$$

Bestrafe Ausdehnung in latentem Raum mit $\lambda\|\mathbf{X}\|$

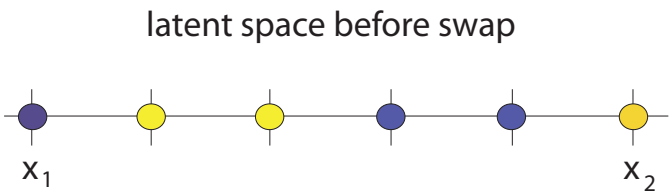
$$\min E(\mathbf{X})_p := \min \left(E(\mathbf{X}) + \lambda\|\mathbf{X}\|\right) \tag{4}$$

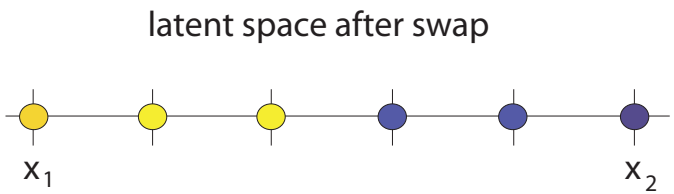
Kombinatorische Variante

(1+1)-EA

Require: data set $\mathbf{Y}$, **Request:** embedding $\mathbf{X}$

1: initialization: random order of $\mathbf{X} = (\mathbf{x}_1, \ldots \mathbf{x}_N)$.
2: **repeat**
3: choose two points $p_1, p_2 \in \mathbb{N}$
4: change $\mathbf{X}$ to $\mathbf{X}'$ by swapping $\mathbf{x}_{p_1}$ and $\mathbf{x}_{p_2}$
5: replace $\mathbf{X}$ by $\mathbf{X}'$ if $E(\mathbf{X}') \leq E(\mathbf{X})$
6: **until** termination condition
7: **return** embedding $\mathbf{X}$



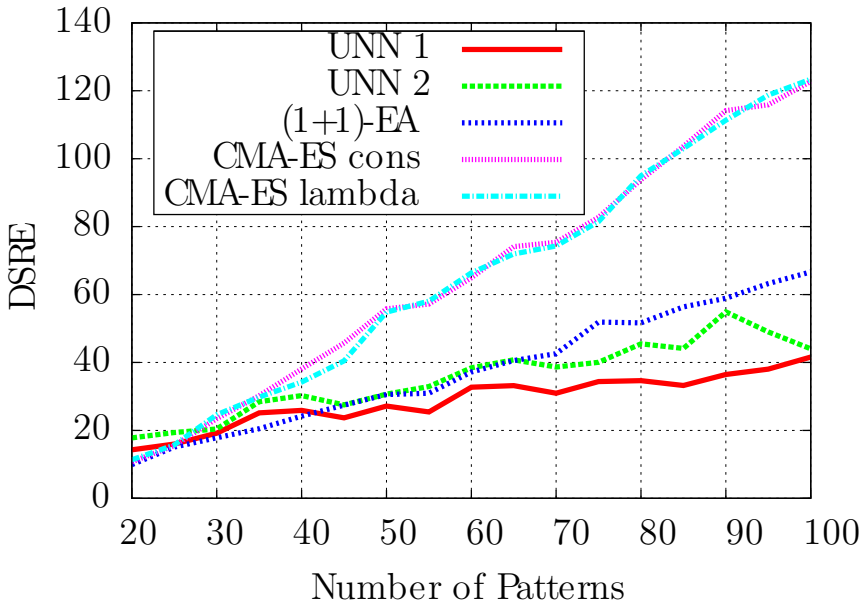


Evo-UNN

$N = 30$	3D-S		
K	2	5	10
init	34.8 ± 0.0	46.8 ± 0.0	51.3 ± 0.0
UNN_g	23.4 ± 0.0	31.3 ± 0.0	43.3 ± 0.0
CMA, $[0,1]^q$	24.5 ± 11.6	27.6 ± 8.4	36.3 ± 15.0
CMA, $\lambda \|\mathbf{X}\|$	22.2 ± 6.1	31.6 ± 10.8	41.4 ± 8.3
$(1+1)$-EA	$\mathbf{13.3} \pm 1.3$	$\mathbf{24.4} \pm 2.5$	$\mathbf{31.1} \pm 1.7$
LLE	13.7 ± 0.0	34.1 ± 0.0	49.6 ± 0.0

Tabelle : DSRE Vergleich für UNN_g, CMA-ES Ansatz, (1+1)-EA (1000 FFE), and LLE.

Curse of Dimensionality



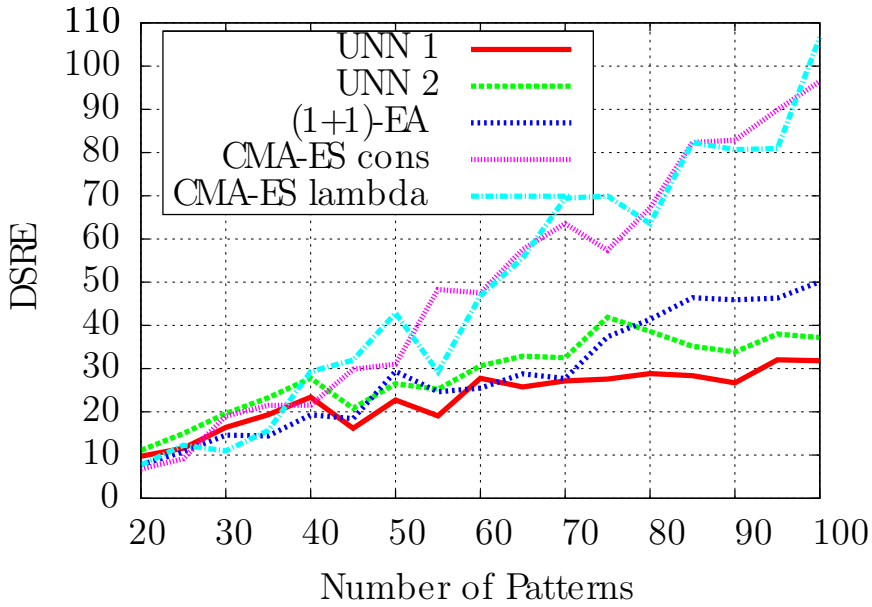


Abbildung : *Curse of dimensionality* für Evo-UNN Varianten auf (a) 3D-S und (b) 3D-S mit Loch

Übersicht

Stochastische UNN-Variante

Erweiterung für $q > 1$

- Sample im latenten Raum mit Gaussverteilung $\mathcal{N}(\mathbf{x}_i, \sigma)$
- σ ergibt sich durch Abstand von $\mathbf{y}$ zum nächsten eingebetteten Muster $\mathbf{y}^*$

Algorithmus

1. Wähle zufällig $\mathbf{y} \in \mathbf{Y}$
2. Suche nächstes Muster $\mathbf{y}^*$ mit latenter Position $\mathbf{x}^*$
3. FOR $i = 1$ **TO** μ
4. $\quad \mathbf{x}_i^* = \mathcal{N}(\mathbf{x}^*, \sigma)$ mit $\sigma = d(\mathbf{y}, \mathbf{y}^*)$
5. Wähle $\mathbf{x}^{**} = \arg\min_{1,\ldots,\mu} E(\mathbf{x}_i^*, \mathbf{X})$ und bette ein
6. $\mathbf{Y} = \mathbf{Y} \backslash \mathbf{y}$
7. Wiederhole ab Schritt 1 bis $\mathbf{Y} = \emptyset$

k-d-Bäume

Komplexität

- Iteratives Hinzufügen: Für N Muster: Suche in 1 bis N' Muster: $\mathcal{O}(N^2)$
- Durch k-d-Bäume:
  1. Suche nach $\mathbf{y}^*$ in $\mathcal{O}(\log N')$
  2. suche μ mal K nächste Nachbarn im latenten Raum: $\mathcal{O}(\mu \cdot K \cdot \log N') = \mathcal{O}(\log N')$
  3. füge $\mathbf{x}$ und $\mathbf{y}$ zu k-d-Bäumen zu: $\mathcal{O}(2 \log N') = \mathcal{O}(\log N')$
- resultiert in $\mathcal{O}(N \log N)$

UNN - Digits

Digits-Datensatz

Einbettung von handgeschriebenen Zahlen (0,1 und 2)

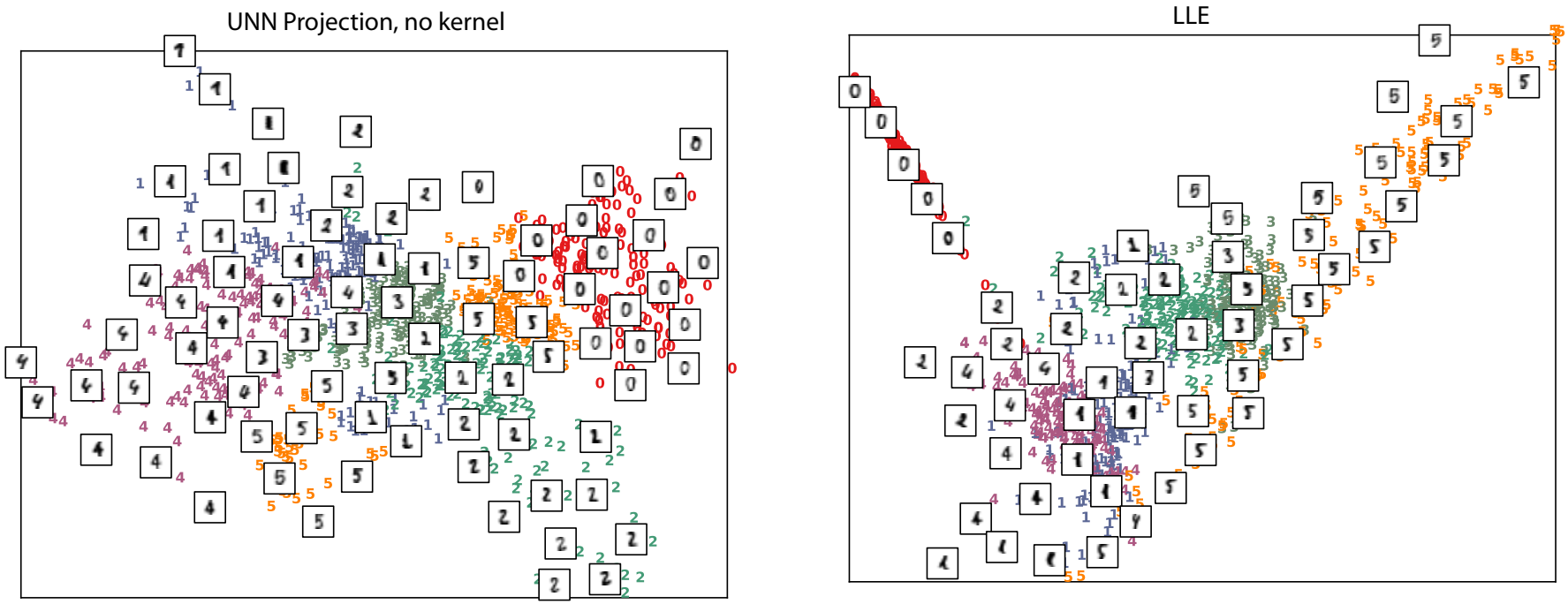


Abbildung : Vergleich von UNN (links) und LLE (rechts) bei Einbettung des *Digits* Datensatzes.

UNN - Galaxien

Astronomie

Einbettung von Galaxie-Bildern (40×40 RGB-Bilder)

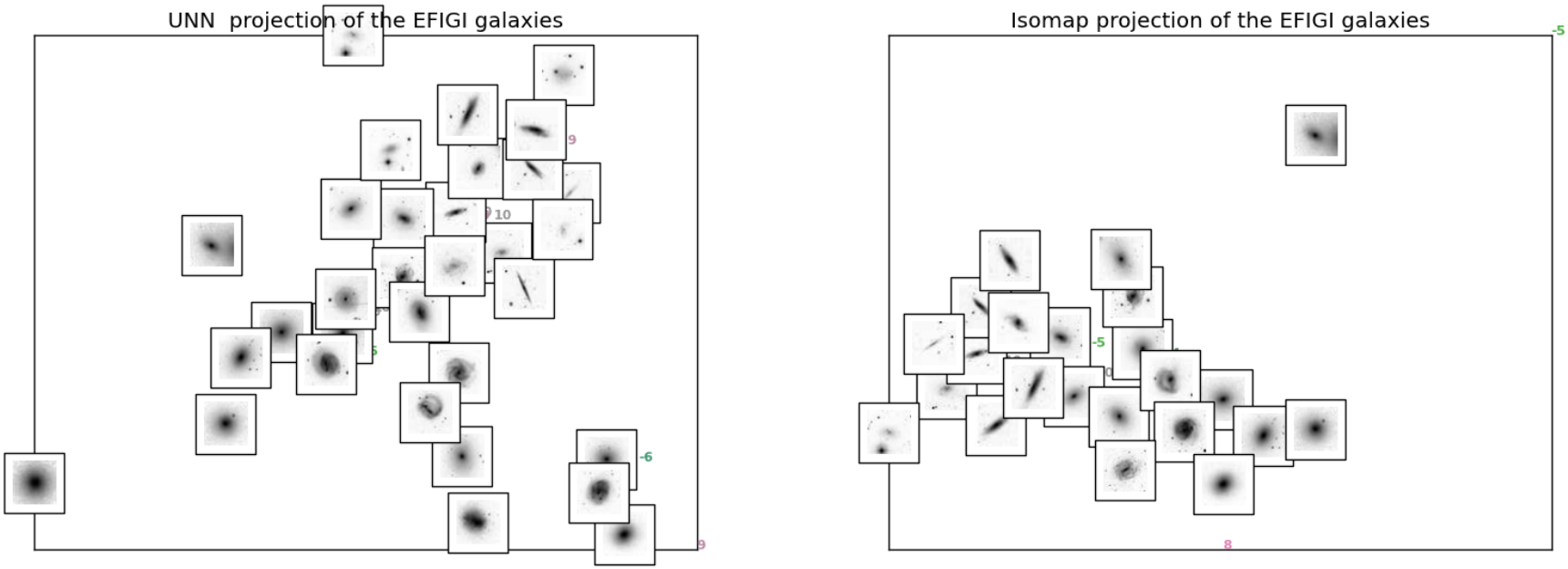


Abbildung : Vergleich von UNN (links) und ISOMAP (rechts) bei Einbettung des EFIGI Galaxie Datensatzes.

Übersicht

Geräteerkennung durch Energieverbrauch-Analyse

Ziele

- Energieberatung
- Gesundheitswesen, Monitoring alter Menschen
- Energieprognose

Idee

- Physikalische Merkmale mit höchstem Informationsgehalt identifizieren
- Trainingsdaten generieren (im Labor oder vor Ort)
- Modell lernen (z.B. mit SVM)

Geräteerkennung, Experimentelle Analyse

Ensemble-Methoden

- 15 Geräte (an/aus)
- K-nächste Nachbarn ($K = 1, 3, 7$), SVM (linear, RBF)
- Ensemble der Klassifikatoren $\mathbf{f}$:

$$f_{\text{ENS}}(\mathbf{x}') = \arg\max_{y \in \mathcal{Y}} \sum_{f_i \in \mathbf{f}} \mathcal{I}(f_i(\mathbf{x}') = y) \tag{5}$$

Trainings-menge	SVM linear	SVM RBF	KNN $K = 1$	KNN $K = 7$	ENS *
Install	**0.0787**	0.4767	0.0883	0.2927	*0.0802*
10^{-1}	*0.0526*	0.0915	0.0652	0.1430	**0.0514**
2^{-1}	0.0457	0.0617	0.0617	*0.0446*	**0.0443**
$2 \cdot 3^{-1}$	0.0490	0.0572	0.0617	**0.0389**	*0.0446*
$\sum$ Punkte	3	0	0	3	5

Suffosionsstabilität

Suffosionsstabilität

- Bewertung der Bodenstabilität für Erdbauwerke
- Feine Kornfraktionen können durch Strömungskraft des Wassers mobilisiert werden (Suffosion)
- Ziel: Bewertung der Suffosionsstabilität aufgrund von Korngrößen d_{10} bis d_{100}
- Bisherige Kriterien: **geometrische** und hydraulische Suffosionskriterien

Suffosionsstabilität (ii)

Tan *et al.* (1987)

- $d_{85}/d_{50} < 5$ und
- $d_{50}/d_{35} < 5$ und
- $d_{35}/d_{15} < 5$: keine Suffosion

Burenkova *et al.* (1993)

- $h'' = d_{90}/d_{15}$, $h' = d_{90}/d_{60}$
- $0,76 \log(h'') + 1 < h' < 1,86 \log(h'') + 1$: keine Suffosion

Suffosionsstabilität mit SVMs

- empirisches geometrisches Modell
- Trainingsdatensatz: 20 Böden mit 10 Merkmalen (d_{10} bis d_{100})
- SVM mit linearem Kernel
- Kreuzvalidierungsfehler (LOO-CV): **0.0**
- Auswertung auf 35 Böden in Bearbeitung (Prof. Witt, Weimar)

Unüberwachte Nächste Nachbarn

Ein effizientes Verfahren zur Dimensionsreduktion

Oliver Kramer

Department für Informatik
Carl von Ossietzky Universität Oldenburg

4. Mai 2012