On-line Handwriting Recognition with Support Vector Machines— A Kernel Approach

Claus Bahlmann, Bernard Haasdonk and Hans Burkhardt,

Computer Science Department,
Albert-Ludwigs-University Freiburg, Germany

August 6, 2002

Overview

- Review of classification techniques
  - (Generative) Bayesian classification
  - (Discriminative) Support Vector Machine (SVM) classification
- Our new SVM-kernel for sequences:
  the Gaussian dynamic time warping (GDTW) kernel
- Examples, simulations and results with UNIPEN data
  - two-class problems
  - multi-class problems

Overview

- Review of classification techniques
  - (Generative) Bayesian classification
  - (Discriminative) Support Vector Machine (SVM) classification
- Our new SVM-kernel for sequences:
  the Gaussian dynamic time warping (GDTW) kernel
- Examples, simulations and results with UNIPEN data
  - two-class problems
  - multi-class problems

Overview

- Review of classification techniques
  - (Generative) Bayesian classification
  - (Discriminative) Support Vector Machine (SVM) classification
- Our new SVM-kernel for sequences:
  the Gaussian dynamic time warping (GDTW) kernel
- Examples, simulations and results with UNIPEN data
  - two-class problems
  - multi-class problems

Overview

- Review of classification techniques
  - (Generative) Bayesian classification
  - (Discriminative) Support Vector Machine (SVM) classification
- Our new SVM-kernel for sequences:
  the Gaussian dynamic time warping (GDTW) kernel
- Examples, simulations and results with UNIPEN data
  - two-class problems
  - multi-class problems

Overview

- Review of classification techniques
  - (Generative) Bayesian classification
  - (Discriminative) Support Vector Machine (SVM) classification
- Our new SVM-kernel for sequences:
  the Gaussian dynamic time warping (GDTW) kernel
- Examples, simulations and results with UNIPEN data
  - two-class problems
  - multi-class problems

Overview

- Review of classification techniques
  - (Generative) Bayesian classification
  - (Discriminative) Support Vector Machine (SVM) classification
- Our new SVM-kernel for sequences:
  the Gaussian dynamic time warping (GDTW) kernel
- Examples, simulations and results with UNIPEN data
  - two-class problems
  - multi-class problems

Overview

- Review of classification techniques
  - (Generative) Bayesian classification
  - (Discriminative) Support Vector Machine (SVM) classification
- Our new SVM-kernel for sequences:
  the Gaussian dynamic time warping (GDTW) kernel
- Examples, simulations and results with UNIPEN data
  - two-class problems
  - multi-class problems

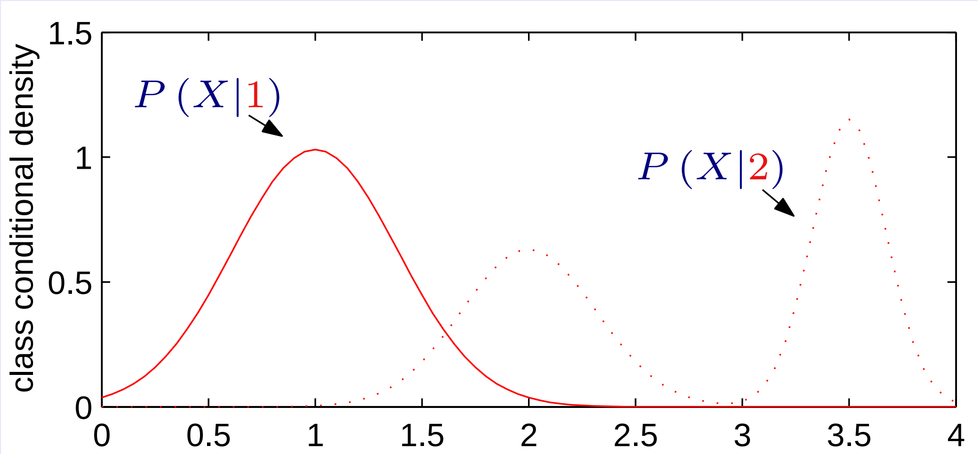
Bayesian Classification

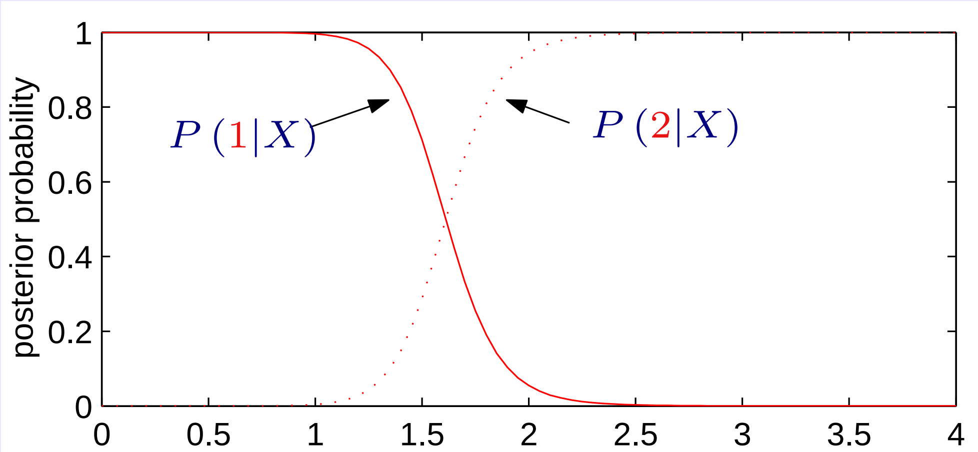
The generative approach

1. Estimate class conditional density models $P(X|l)$ for each class l

2. Choose class with highest posterior probability by Bayes' rule

$$P(l|T) = \frac{P(T|l)\,P(l)}{P(T)}$$





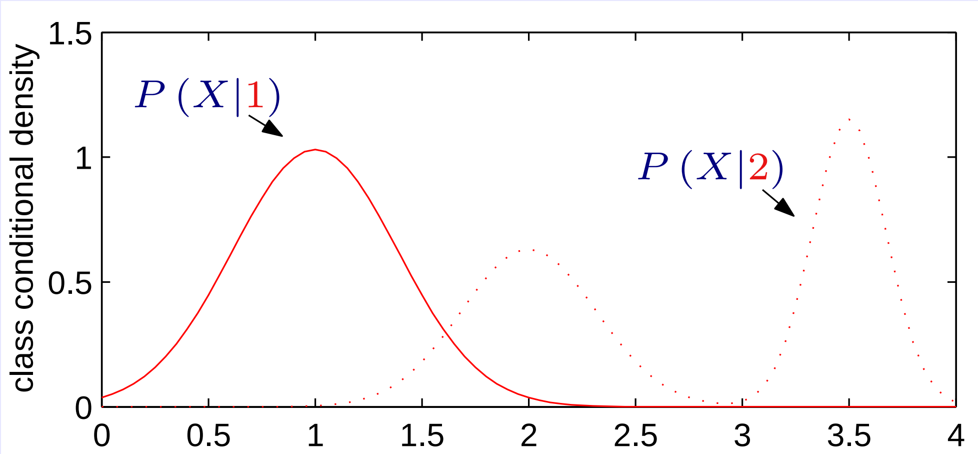
Bayesian Classification

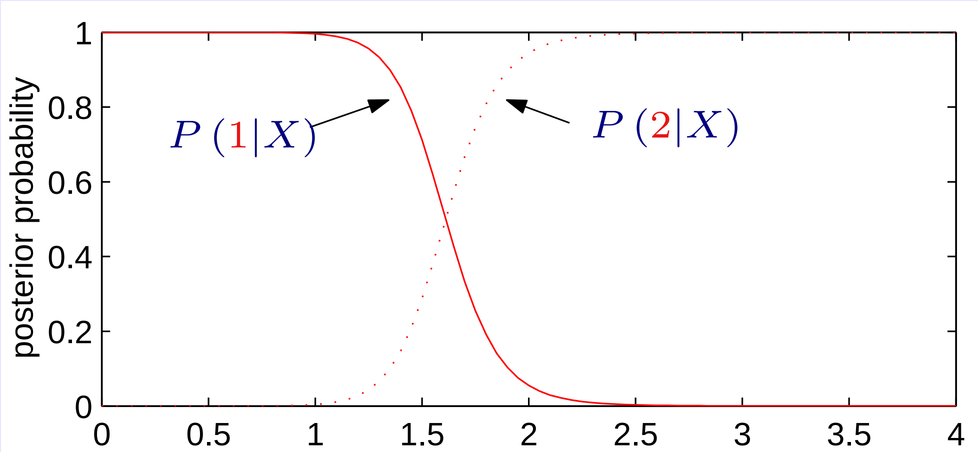
The generative approach

1. Estimate class conditional density models $P(X|l)$ for each class l

2. Choose class with highest posterior probability by Bayes' rule

$$P(l|T) = \frac{P(T|l)\,P(l)}{P(T)}$$



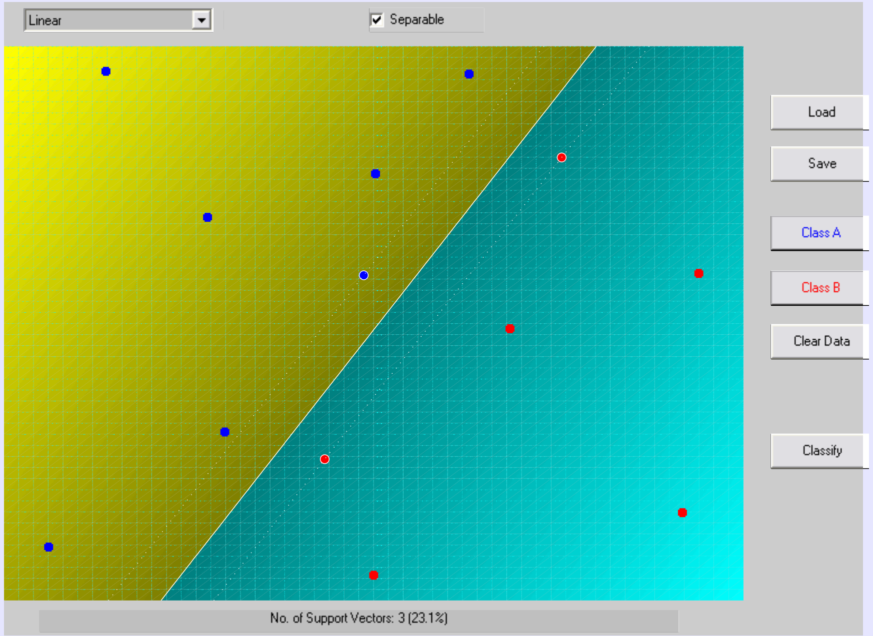


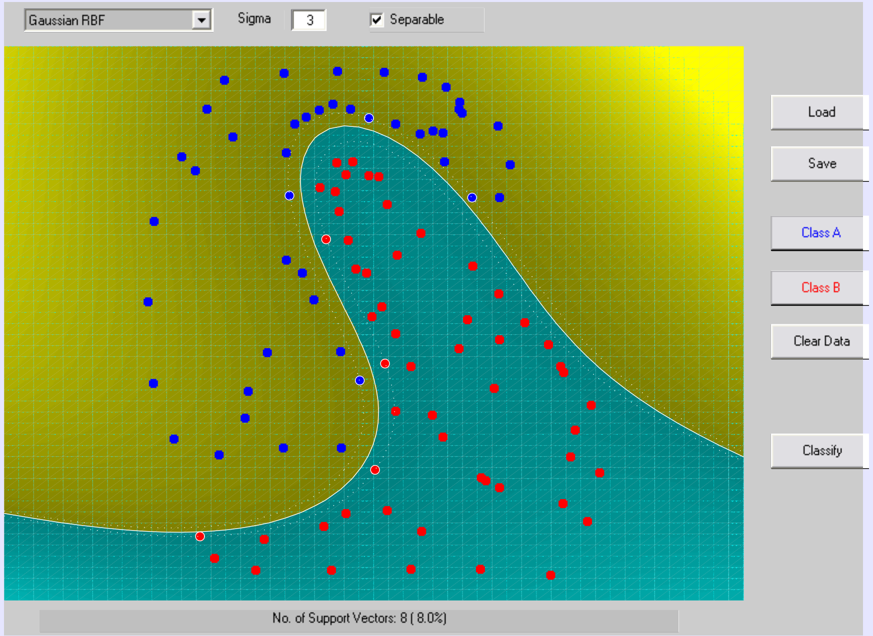
Support Vector Machine (SVM)

The Discriminative Approach (Two-class Case)

- Discrimination boundary has *widest margin* to “closest” training examples (*support vectors*)

- Non-linear extension by implicit problem transformation into higher dimensional space by the “kernel trick”





SVM GUI by (Gunn, 1998)

Support Vector Machine (SVM)

The Discriminative Approach (Two-class Case)

Kernel:

$$K\left(T, P\right)$$

SVM classification:

$$\hat{S}\left(T\right) = \operatorname{sign}\left(\sum_i \alpha_i S_i K\left(T, P_i\right) + b\right)$$

SVM training: Determine α_i, that maximize the objective function

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j S_i S_j K\left(P_i, P_j\right)$$

with the constraints

$$0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_i \alpha_i S_i = 0$$

Support Vector Machine (SVM)

The Discriminative Approach (Two-class Case)

Kernel:

$$K(T, P)$$

SVM classification:

$$\hat{S}(T) = \operatorname{sign}\left(\sum_i \alpha_i S_i K(T, P_i) + b\right)$$

SVM training: Determine α_i, that maximize the objective function

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j S_i S_j K(P_i, P_j)$$

with the constraints

$$0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_i \alpha_i S_i = 0$$

Support Vector Machine (SVM)

The Discriminative Approach (Two-class Case)

Kernel:

$$K\left(T, P\right)$$

SVM classification:

$$\hat{S}\left(T\right) = \operatorname{sign}\left(\sum_i \alpha_i S_i K\left(T, P_i\right) + b\right)$$

SVM training: Determine α_i, that maximize the objective function

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j S_i S_j K\left(P_i, P_j\right)$$

with the constraints

$$0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_i \alpha_i S_i = 0$$

Support Vector Machine (SVM)

The Discriminative Approach (Two-class Case)

Kernel:

$$K(T, P)$$

SVM classification:

$$\hat{S}(T) = \operatorname{sign}\left(\sum_i \alpha_i S_i K(T, P_i) + b\right)$$

SVM training: Determine α_i, that maximize the objective function

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j S_i S_j K(P_i, P_j)$$

with the constraints

$$0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_i \alpha_i S_i = 0$$

Support Vector Machine (SVM)

Kernels

	Vectors	Sequences (On-line handwriting data!)
Pattern examples	$T = (7,5,8)^\tau$ $P = (9,3,4)^\tau$	$\mathcal{T} = [7,5,8]$ $\mathcal{P} = [7,5,5,8]$
Kernel example $K(T,P)$	Gaussian kernel $K(T,P) = \exp\left(-\gamma \left\|T-P\right\|^2\right)$	?

Support Vector Machine (SVM)

Kernels

	Vectors	Sequences (On-line handwriting data!)
Pattern examples	$T = (7,5,8)^\tau$ $P = (9,3,4)^\tau$	$\mathcal{T} = [7,5,8]$ $\mathcal{P} = [7,5,5,8]$
Kernel example $K(T,P)$	Gaussian kernel $K(T,P) = \exp\left(-\gamma \left\|T-P\right\|^2\right)$	Gaussian DTW (GDTW) kernel $K(\mathcal{T},\mathcal{P}) = \exp\left(-\gamma D_{\mathrm{DTW}}(\mathcal{T},\mathcal{P})\right)$

Support Vector Machine (SVM)

Kernels

	Vectors	Sequences (On-line handwriting data!)
Pattern examples	$T = (7,5,8)^\tau$ $P = (9,3,4)^\tau$	$\mathcal{T} = [7,5,8]$ $\mathcal{P} = [7,5,5,8]$
Kernel example $K(T,P)$	Gaussian kernel $K(T,P) = \exp\left(-\gamma \left\|T-P\right\|^2\right)$	Gaussian DTW (GDTW) kernel $K(\mathcal{T},\mathcal{P}) = \exp\left(-\gamma D_{\mathrm{DTW}}(\mathcal{T},\mathcal{P})\right)$ (however, GDTW cannot be proven to be positive definite;

Support Vector Machine (SVM)

Kernels

	Vectors	Sequences (On-line handwriting data!)
Pattern examples	$T = (7,5,8)^\tau$ $P = (9,3,4)^\tau$	$\mathcal{T} = [7,5,8]$ $\mathcal{P} = [7,5,5,8]$
Kernel example $K(T,P)$	Gaussian kernel $K(T,P) = \exp\left(-\gamma \|T-P\|^2\right)$	Gaussian DTW (GDTW) kernel $K(\mathcal{T},\mathcal{P}) = \exp\left(-\gamma D_{\mathrm{DTW}}(\mathcal{T},\mathcal{P})\right)$ (however, GDTW cannot be proven to be positive definite; but, positive definite in (many) practical evaluations)

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

$$\begin{aligned} \mathcal{T} &= \left[\mathbf{t}_1, \ldots, \mathbf{t}_{N_{\mathcal{T}}}\right] \\ \mathcal{R} &= \left[\mathbf{r}_1, \ldots, \mathbf{r}_{N_{\mathcal{R}}}\right] \end{aligned}$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$

Warping path: (for aligning corresponding samples)

$$\boldsymbol{\phi} : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_{\mathcal{T}}\} \times \{1, \ldots, N_{\mathcal{R}}\})$$

DTW distance:

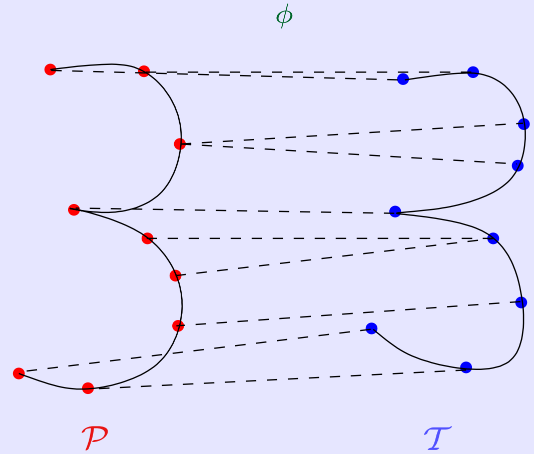
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

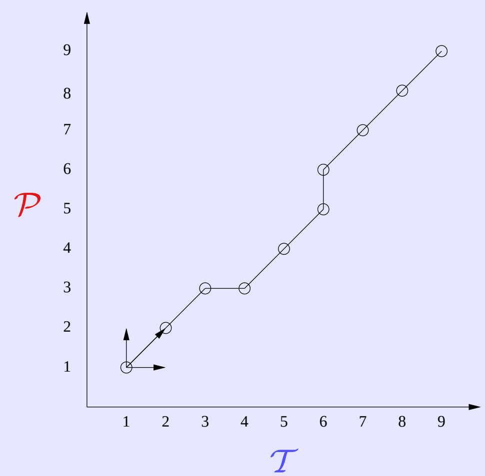
$$\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$
$$\mathcal{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\boldsymbol{\phi} : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

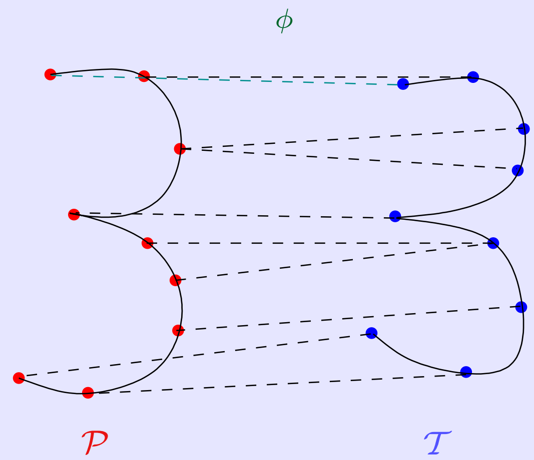
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

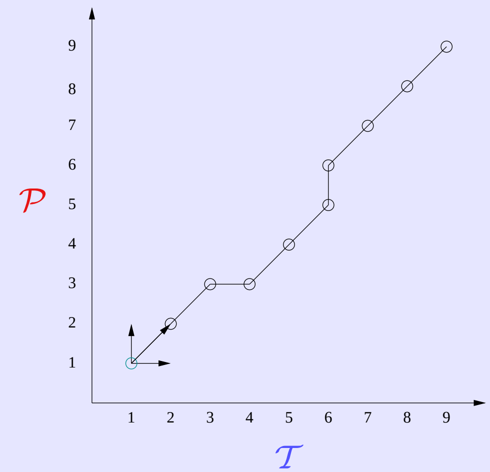
$$\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$
$$\mathcal{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}]$$

and compute a distance measure $D_{\text{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\phi : \{1, \ldots, N\} \to (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

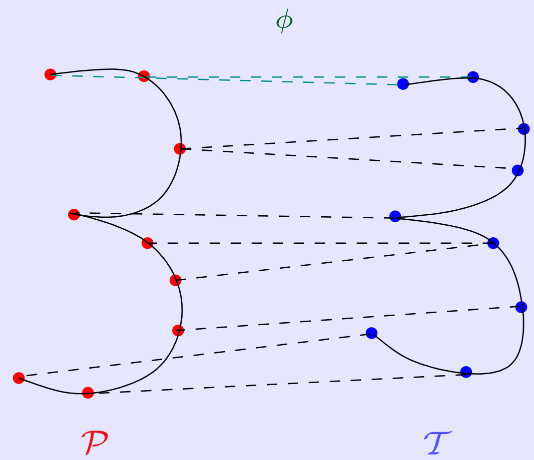
$$D_{\text{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

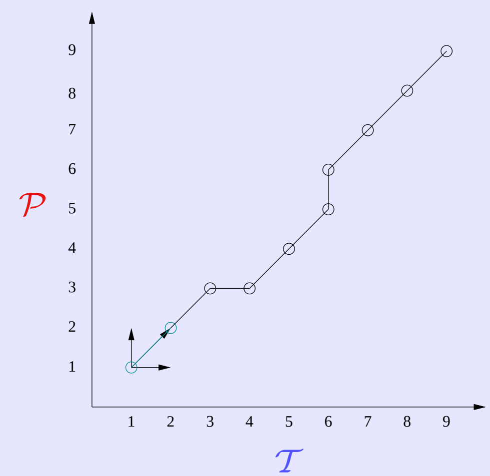
$$\mathcal{T} = \left[\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}\right]$$
$$\mathcal{R} = \left[\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}\right]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\phi : \{1, \ldots, N\} \to (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

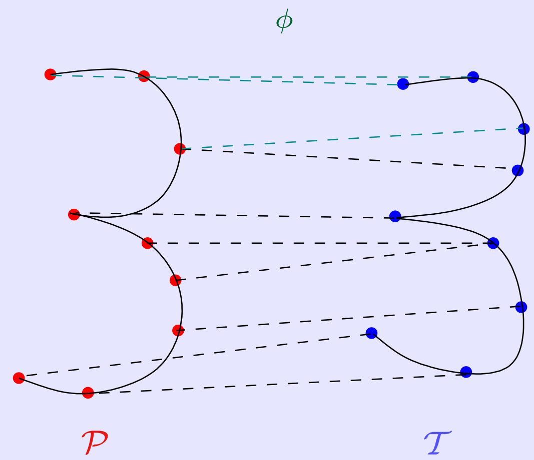
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_\mathcal{T}(n)} - \mathbf{r}_{\phi^*_\mathcal{R}(n)} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

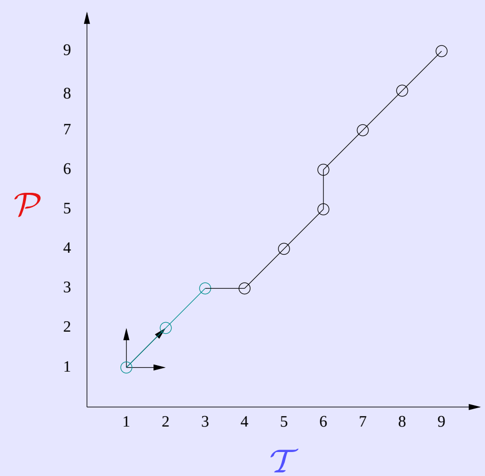
$$\mathcal{T} = \left[\mathbf{t}_1, \ldots, \mathbf{t}_{N_{\mathcal{T}}}\right]$$
$$\mathcal{R} = \left[\mathbf{r}_1, \ldots, \mathbf{r}_{N_{\mathcal{R}}}\right]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\boldsymbol{\phi} : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_{\mathcal{T}}\} \times \{1, \ldots, N_{\mathcal{R}}\})$$



DTW distance:

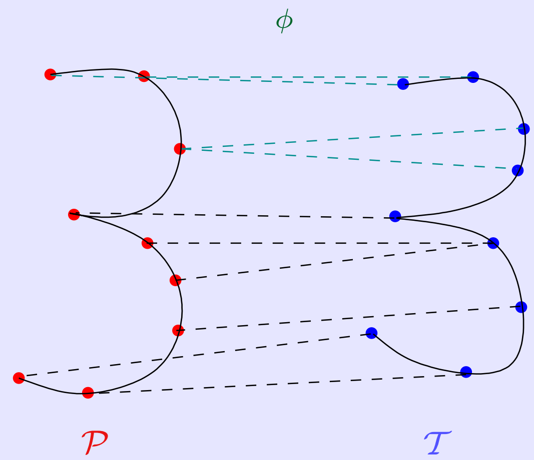
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

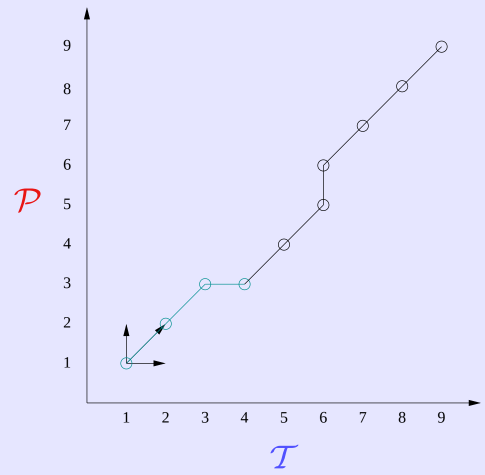
$$\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$
$$\mathcal{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\phi : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

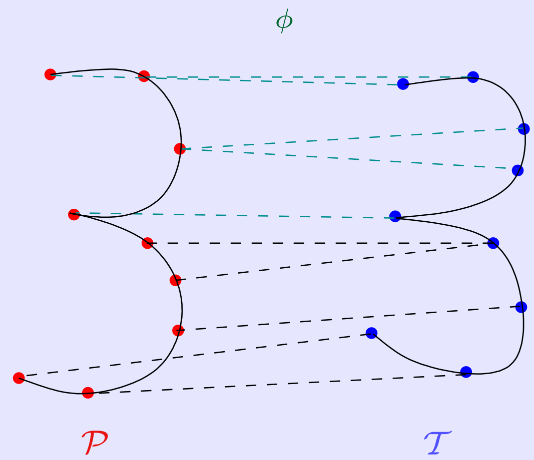
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

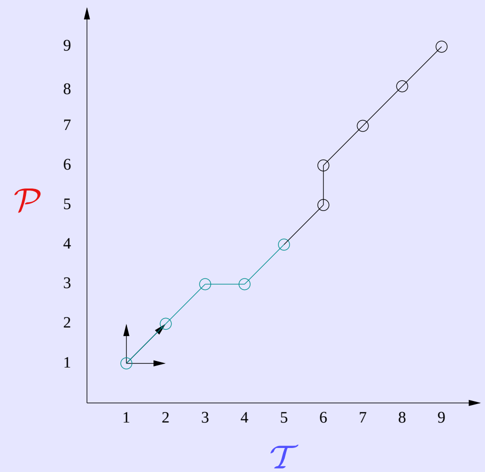
$$\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$
$$\mathcal{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\boldsymbol{\phi} : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

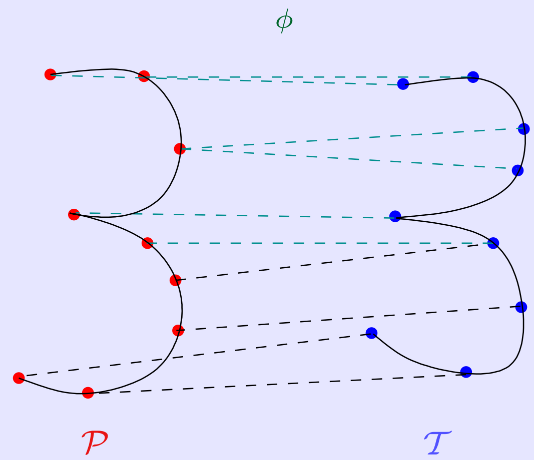
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_\mathcal{T}(n)} - \mathbf{r}_{\phi^*_\mathcal{R}(n)} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

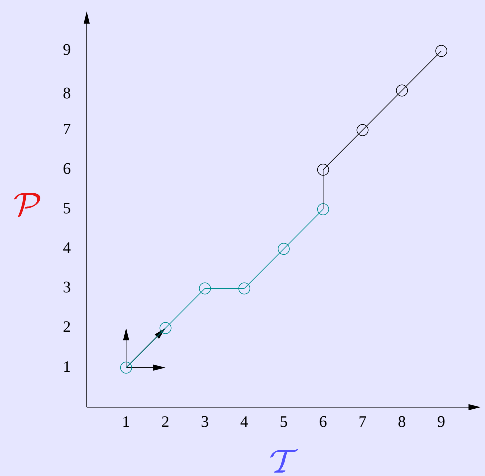
$$\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$
$$\mathcal{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}]$$

and compute a distance measure $D_{\text{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\phi : \{1, \ldots, N\} \to (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

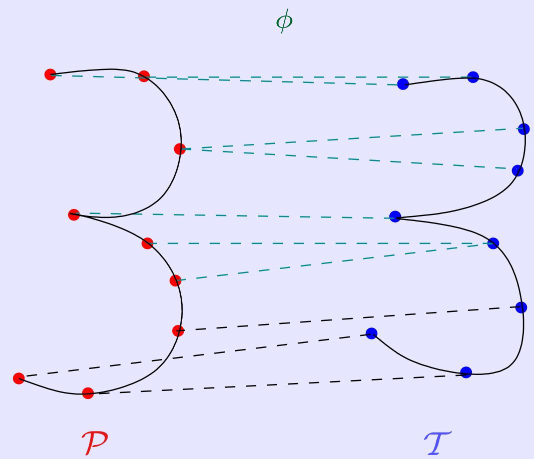
$$D_{\text{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_\mathcal{T}(n)} - \mathbf{r}_{\phi^*_\mathcal{R}(n)} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

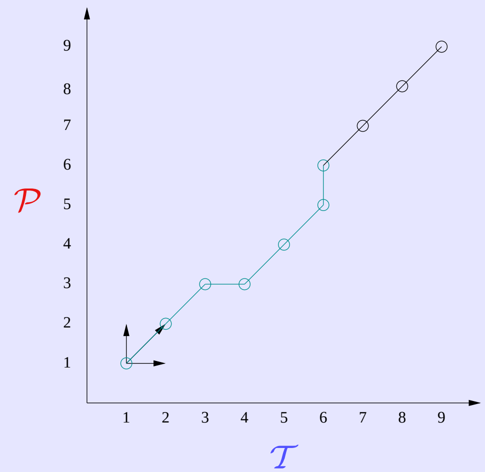
$$\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$
$$\mathcal{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\boldsymbol{\phi} : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

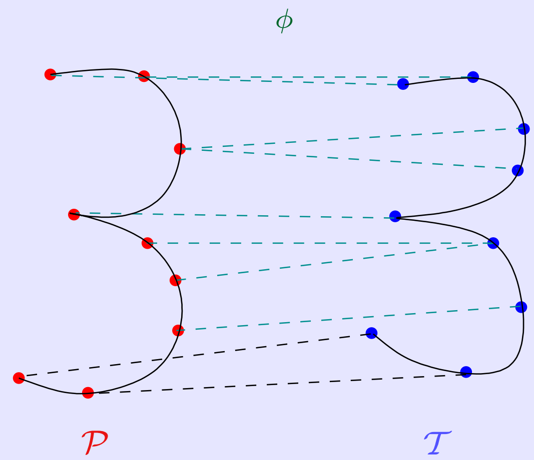
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

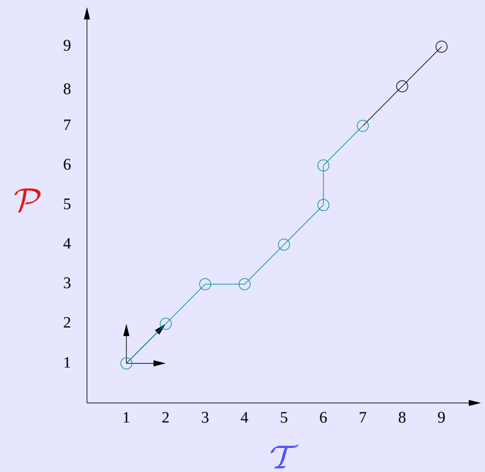
$$\mathcal{T} = \left[\mathbf{t}_1, \ldots, \mathbf{t}_{N_{\mathcal{T}}}\right]$$
$$\mathcal{R} = \left[\mathbf{r}_1, \ldots, \mathbf{r}_{N_{\mathcal{R}}}\right]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\boldsymbol{\phi} : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_{\mathcal{T}}\} \times \{1, \ldots, N_{\mathcal{R}}\})$$



DTW distance:

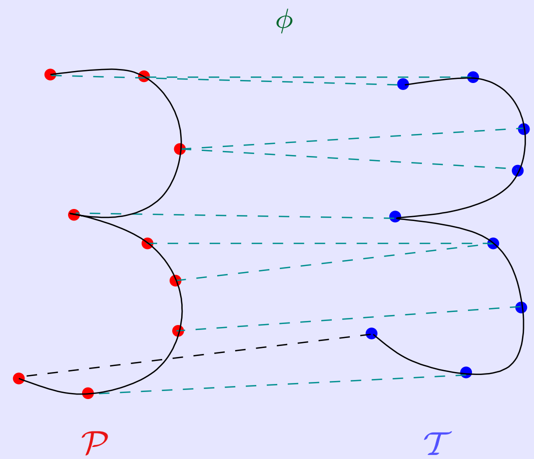
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

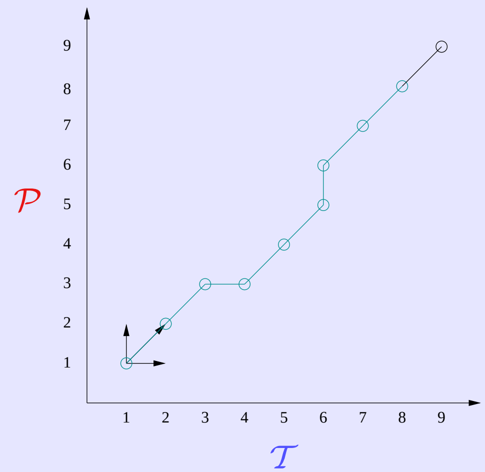
$$\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$
$$\mathcal{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}]$$

and compute a distance measure $D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R})$



Warping path: (for aligning corresponding samples)

$$\phi : \{1, \ldots, N\} \to (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

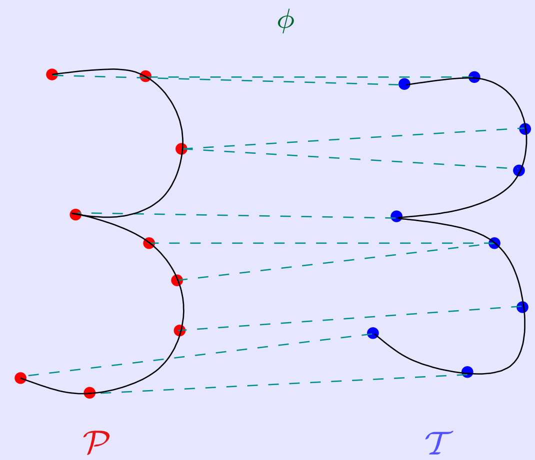
$$D_{\mathrm{DTW}}(\mathcal{T}, \mathcal{R}) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_{\mathcal{T}(n)}} - \mathbf{r}_{\phi^*_{\mathcal{R}(n)}} \right\|^2$$

Dynamic Time Warping

Purpose: Aligning temporally distorted patterns

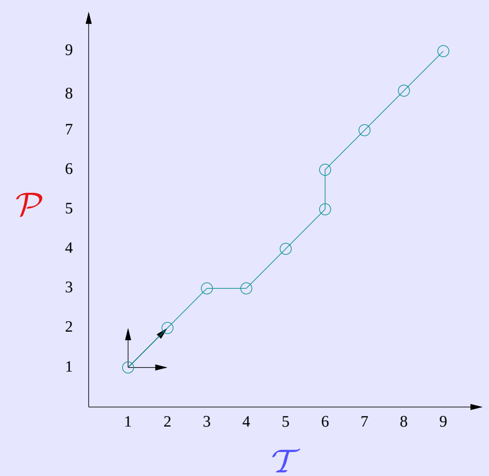
$$\mathcal{T} = \left[\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}\right]$$
$$\mathcal{R} = \left[\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}}\right]$$

and compute a distance measure $D_{\mathrm{DTW}}\left(\mathcal{T}, \mathcal{R}\right)$



Warping path: (for aligning corresponding samples)

$$\boldsymbol{\phi} : \{1, \ldots, N\} \rightarrow (\{1, \ldots, N_\mathcal{T}\} \times \{1, \ldots, N_\mathcal{R}\})$$



DTW distance:

$$D_{\mathrm{DTW}}\left(\mathcal{T}, \mathcal{R}\right) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{t}_{\phi^*_\mathcal{T}(n)} - \mathbf{r}_{\phi^*_\mathcal{R}(n)} \right\|^2$$

Simulations and Results

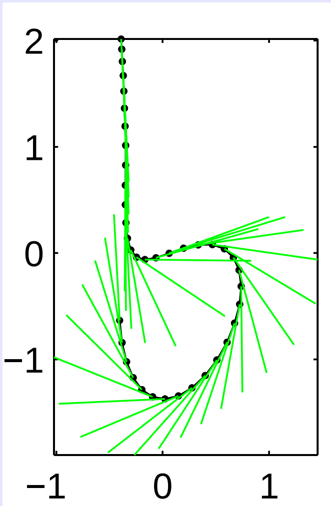
The Database

- UNIPEN Train-R01/V07 corpus
- **no** cleaning from poor quality/mislabeled characters

UNIPEN section	number of samples
1a (digits)	16000
1b (upper case characters)	28000
1c (lower case characters)	61000

Simulations and Results

Feature Selection



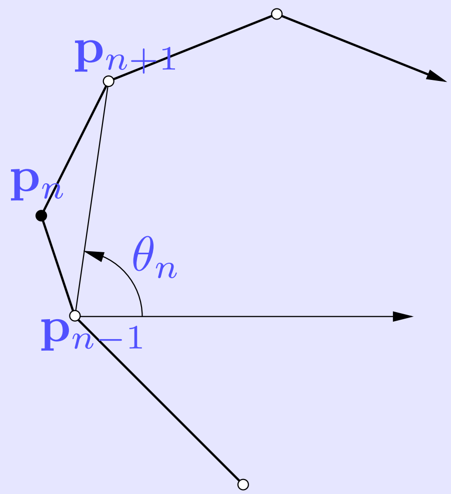
$$\text{Feature vector sequence } \mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}}]$$

$$\text{Feature vector } \mathbf{t}_n = (\tilde{x}_n, \tilde{y}_n, \theta_n)^\tau$$

1. normalized x-coordinate $\tilde{x}_n = \frac{x_n - \mu_x}{\sigma_y}$

2. normalized y-coordinate $\tilde{y}_n = \frac{y_n - \mu_y}{\sigma_y}$

3. tangent slope angle

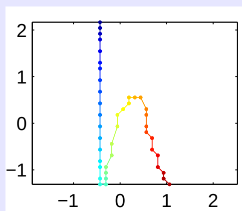
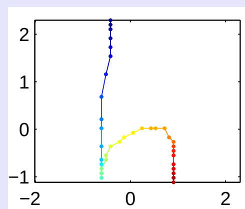
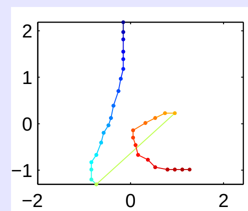
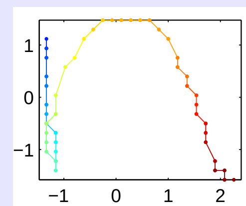
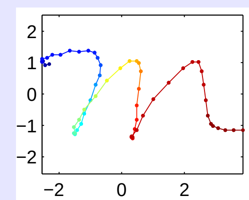
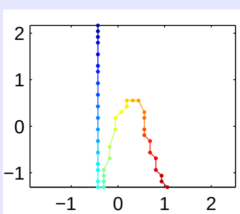
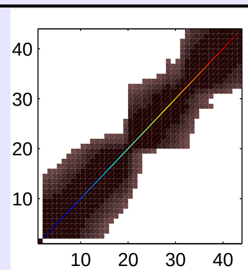
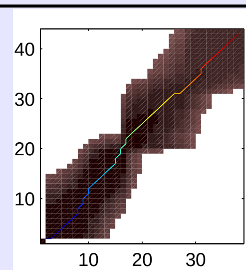
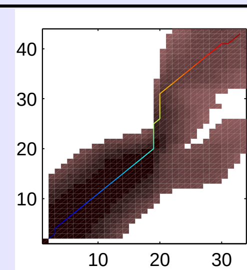
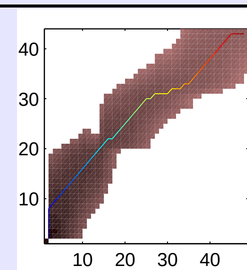
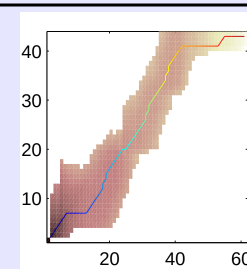


$$\theta_n = \text{ang}\left(\Delta_n x + \sqrt{-1} \cdot \Delta_n y\right)$$

Gaussian DTW (GDTW) Kernel

Examples

$$K\left(\mathcal{T},\mathcal{P}_j\right) = \exp\left(-\gamma D_{\mathrm{DTW}}\left(\mathcal{T},\mathcal{P}_j\right)\right)$$

	$\mathcal{P}_1$ 	$\mathcal{P}_2$ 	$\mathcal{P}_3$ 	$\mathcal{P}_4$ 	$\mathcal{P}_5$ 
$\mathcal{T}$ 					
$D_{DTW}\left(\mathcal{T},\mathcal{P}_j\right)$	0	0.20	0.71	0.99	10.04
$K\left(\mathcal{T},\mathcal{P}_j\right)$	1	0.70	0.28	0.17	0.00

Error Rates of Two-class Problems

1c section (lower case characters),
randomly chosen 67 % Train / 33 % Test set

Difficulty	Character pairs	# Tr.-Expls.	# SVs	$E_{\mathrm{SVM-GDTW}}$	E_{SDTW} [BB01]
easy	a $\leftrightarrow$ b	3540	298	0.5 %	0.8 %
	d $\leftrightarrow$ m	2595	334	0.1 %	0.4 %
difficult	c $\leftrightarrow$ e	5088	351	3.7 %	7.2 %
	u $\leftrightarrow$ v	2214	397	9.2 %	6.8 %
	y $\leftrightarrow$ g	2088	358	11.2 %	7.7 %
	b $\leftrightarrow$ h	2524	275	2.3 %	3.2 %

Error Rates of Two-class Problems

1c section (lower case characters),
randomly chosen 67 % Train / 33 % Test set

Difficulty	Character pairs	# Tr.-Expls.	# SVs	$E_{\mathrm{SVM-GDTW}}$	E_{SDTW} [BB01]
easy	a $\leftrightarrow$ b	3540	298	0.5 %	0.8 %
	d $\leftrightarrow$ m	2595	334	0.1 %	0.4 %
difficult	c $\leftrightarrow$ e	5088	351	3.7 %	7.2 %
	u $\leftrightarrow$ v	2214	397	9.2 %	6.8 %
	y $\leftrightarrow$ g	2088	358	11.2 %	7.7 %
	b $\leftrightarrow$ h	2524	275	2.3 %	3.2 %

Error Rates of Two-class Problems

1c section (lower case characters),
randomly chosen 67 % Train / 33 % Test set

Difficulty	Character pairs	# Tr.-Expls.	# SVs	$E_{\mathrm{SVM-GDTW}}$	E_{SDTW} [BB01]
easy	a $\leftrightarrow$ b	3540	298	0.5 %	0.8 %
	d $\leftrightarrow$ m	2595	334	0.1 %	0.4 %
difficult	c $\leftrightarrow$ e	5088	351	3.7 %	7.2 %
	u $\leftrightarrow$ v	2214	397	9.2 %	6.8 %
	y $\leftrightarrow$ g	2088	358	11.2 %	7.7 %
	b $\leftrightarrow$ h	2524	275	2.3 %	3.2 %

Multi-class SVM

DAG (directed acyclic graph)-SVM:

combining $K \cdot (K-1)/2$ two-class SVMs into **one** K-class-SVM

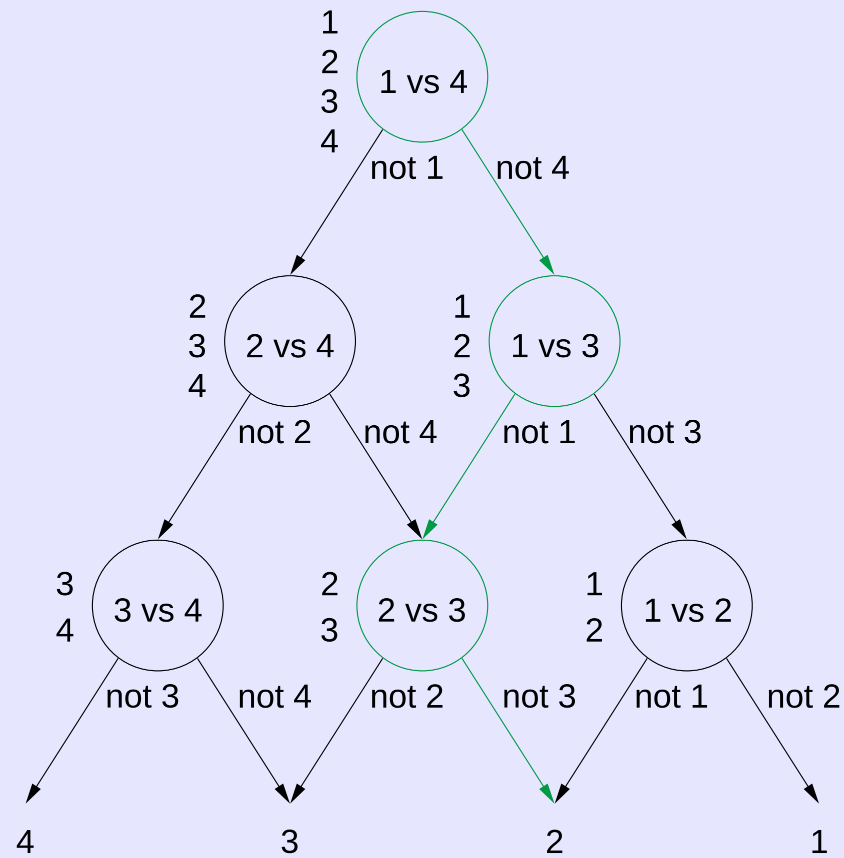


figure taken from (Platt, 2000)

Example of a Multi-class SVM-GDTW

Matlab demo

Error Rates of Multi-class Problems

1c section (lower case characters)

Approach	Error rate E (average of 5 runs)	UNIPEN Database Type
DAG-SVM-GDTW		Train-R01/V07
	11.5 %	rand. chosen 10 %/10 % Train/Test
	12.0 %	rand. chosen 20 %/20 % Train/Test
SDTW [BB01]		Train-R01/V07
	13.0 %	rand. chosen 10 %/10 % Train/Test
	11.4 %	rand. chosen 20 %/20 % Train/Test
	9.7 %	rand. chosen 67 %/33 % Train/Test
MLP [PLG01]	14.4 %	DevTest-R02/V02
HMM-NN hybrid [GADG01]	13,2 %	Train-R01/V07
HMM [HLB00]	14,1 %	Train-R01/V06 4 % "bad characters" removed

Error Rates of Multi-class Problems

1c section (lower case characters)

Approach	Error rate E (average of 5 runs)	UNIPEN Database Type
DAG-SVM-GDTW		Train-R01/V07
	11.5 %	rand. chosen 10 %/10 % Train/Test
	12.0 %	rand. chosen 20 %/20 % Train/Test
SDTW [BB01]		Train-R01/V07
	13.0 %	rand. chosen 10 %/10 % Train/Test
	11.4 %	rand. chosen 20 %/20 % Train/Test
	9.7 %	rand. chosen 67 %/33 % Train/Test
MLP [PLG01]	14.4 %	DevTest-R02/V02
HMM-NN hybrid [GADG01]	13,2 %	Train-R01/V07
HMM [HLB00]	14,1 %	Train-R01/V06 4 % ”bad characters” removed

Error Rates of Multi-class Problems

1c section (lower case characters)

Approach	Error rate E (average of 5 runs)	UNIPEN Database Type
DAG-SVM-GDTW	**11.5 %** **12.0 %**	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test
SDTW [BB01]	**13.0 %** **11.4 %** 9.7 %	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test rand. chosen 67 %/33 % Train/Test
MLP [PLG01]	14.4 %	DevTest-R02/V02
HMM-NN hybrid [GADG01]	13,2 %	Train-R01/V07
HMM [HLB00]	14,1 %	Train-R01/V06 4 % ”bad characters” removed

Complexity

multi-class, 1c section (lower case characters),
randomly chosen 10 % Train / 10 % Test set

	Order	Experiments on AMD Athlon 1200 MHz
Time		
Training	$\mathcal{O}\left(M^2 \cdot T_{\text{kernel}}\right)$	81 h
Classification	$(K-1) \cdot M_s \cdot T_{\text{kernel}}$	2.5 sec
Memory	$\frac{K \cdot (K-1)}{2} \cdot M_S \cdot \tilde{N} \cdot F \cdot \text{sizeof}(\text{float})$	17.5 MByte

M: total number of training samples
K: number of classes
M_s: average number of support vectors
$\tilde{N}$: average sequence length
F: number of features

Conclusion

- A discriminative classifier for sequences:
  SVM with a Gaussian DTW kernel (SVM-GDTW)
- Examples, simulations and results
  - Small training sets: Significant decrease of error rate
  - Large training sets: Comparable error rates
- Remaining potential for improvement
- Just a small number of model parameters have to be adjusted
- Complexity of SVM-GDTW quite high
- Kernel is **not** positive definite and thus global optimality of the training cannot be guaranteed.
- Suitable for all problems with sequences (speech, genome processing, ...)

Conclusion

- A discriminative classifier for sequences:
  SVM with a Gaussian DTW kernel (SVM-GDTW)

- Examples, simulations and results
  - Small training sets: Significant decrease of error rate
  - Large training sets: Comparable error rates

- Remaining potential for improvement

- Just a small number of model parameters have to be adjusted

- Complexity of SVM-GDTW quite high

- Kernel is **not** positive definite and thus global optimality of the training cannot be guaranteed.

- Suitable for all problems with sequences (speech, genome processing, ...)

Conclusion

- A discriminative classifier for sequences:
  SVM with a Gaussian DTW kernel (SVM-GDTW)
- Examples, simulations and results
  - Small training sets: Significant decrease of error rate
  - Large training sets: Comparable error rates
- Remaining potential for improvement
- Just a small number of model parameters have to be adjusted
- Complexity of SVM-GDTW quite high
- Kernel is **not** positive definite and thus global optimality of the training cannot be guaranteed.
- Suitable for all problems with sequences (speech, genome processing, ...)

Conclusion

- A discriminative classifier for sequences:
  SVM with a Gaussian DTW kernel (SVM-GDTW)
- Examples, simulations and results
  - Small training sets: Significant decrease of error rate
  - Large training sets: Comparable error rates
- Remaining potential for improvement
- Just a small number of model parameters have to be adjusted
- Complexity of SVM-GDTW quite high
- Kernel is **not** positive definite and thus global optimality of the training cannot be guaranteed.
- Suitable for all problems with sequences (speech, genome processing, ...)

Conclusion

- A discriminative classifier for sequences:
  SVM with a Gaussian DTW kernel (SVM-GDTW)
- Examples, simulations and results
  - Small training sets: Significant decrease of error rate
  - Large training sets: Comparable error rates
- Remaining potential for improvement
- Just a small number of model parameters have to be adjusted
- Complexity of SVM-GDTW quite high
- Kernel is **not** positive definite and thus global optimality of the training cannot be guaranteed.
- Suitable for all problems with sequences (speech, genome processing, ...)

Conclusion

- A discriminative classifier for sequences:
  SVM with a Gaussian DTW kernel (SVM-GDTW)
- Examples, simulations and results
  - Small training sets: Significant decrease of error rate
  - Large training sets: Comparable error rates
- Remaining potential for improvement
- Just a small number of model parameters have to be adjusted
- Complexity of SVM-GDTW quite high
- Kernel is **not** positive definite and thus global optimality of the training cannot be guaranteed.
- Suitable for all problems with sequences (speech, genome processing, ...)

Conclusion

- A discriminative classifier for sequences:
  SVM with a Gaussian DTW kernel (SVM-GDTW)
- Examples, simulations and results
  - Small training sets: Significant decrease of error rate
  - Large training sets: Comparable error rates
- Remaining potential for improvement
- Just a small number of model parameters have to be adjusted
- Complexity of SVM-GDTW quite high
- Kernel is **not** positive definite and thus global optimality of the training cannot be guaranteed.
- Suitable for all problems with sequences (speech, genome processing, ...)

Future Work

- Character recognition $\longrightarrow$ word recognition
- Improving computational speed
- Investigating non-positive definiteness
- Investigating additional kernels
- Hybrid of generative / discriminative classifier

Future Work

- Character recognition $\longrightarrow$ word recognition
- Improving computational speed
- Investigating non-positive definiteness
- Investigating additional kernels
- Hybrid of generative / discriminative classifier

Future Work

- Character recognition $\longrightarrow$ word recognition
- Improving computational speed
- Investigating non-positive definiteness
- Investigating additional kernels
- Hybrid of generative / discriminative classifier

Future Work

- Character recognition $\longrightarrow$ word recognition
- Improving computational speed
- Investigating non-positive definiteness
- Investigating additional kernels
- Hybrid of generative / discriminative classifier

Future Work

- Character recognition $\longrightarrow$ word recognition
- Improving computational speed
- Investigating non-positive definiteness
- Investigating additional kernels
- Hybrid of generative / discriminative classifier

References

[BB01] Claus Bahlmann and Hans Burkhardt. Measuring HMM similarity with the Bayes probability of error and its application to online handwriting recognition. In *Proc. of the 6th ICDAR*, pages 406–411, 2001.

[GADG01] N. Gauthier, T. Artères, B. Dorizzi, and P. Gallinari. Strategies for combining on-line and off-line information in an on-line handwriting recognition system. In *Proc. of the 6th ICDAR*, pages 412–416, 2001.

[HLB00] Jianying Hu, Sok Gek Lim, and Michael K. Brown. Writer independent on-line handwriting recognition using an HMM approach. *Pattern Recognition*, 33:133–147, January 2000.

[PCST00] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGS for multiclass classification. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2000.

[PLG01] Marc Parizeau, Alexandre Lemieux, and Christian Gagné. Character recognition experiments using UNIPEN data. In *Proc. of the 6th ICDAR*, pages 481–485, 2001.

Error Rates of Multi-class Problems

UNIPEN section	Approach	Error rate E	UNIPEN Database Type
1a (digits)	DAG-SVM-GDTW	**3.8 %** **3.7 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	SDTW [BB01]	**4.5 %** **3.2 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	MLP [PLG01]	3.0 %	DevTest-R02/V02
	HMM [HLB00]	3.2 %	Train-R01/V06 4 % "bad characters" removed
1b (upper case)	DAG-SVM-GDTW	**7.4 %** **7.3 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	SDTW [BB01]	**10.0 %** **8.0 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	HMM [HLB00]	6.4 %	Train-R01/V06 4 % "bad characters" removed
1c (lower case)	DAG-SVM-GDTW	**11.5 %** **12.0 %**	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test
	SDTW [BB01]	**13.0 %** **11.4 %** 9.7 %	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test rand. chosen 67 %/33 % Train/Test
	MLP [PLG01]	14.4 %	DevTest-R02/V02
	HMM-NN hybrid [GADG01]	13,2 %	Train-R01/V07
	HMM [HLB00]	14,1 %	Train-R01/V06 4 % "bad characters" removed

Error Rates of Multi-class Problems

UNIPEN section	Approach	Error rate E	UNIPEN Database Type
1a (digits)	DAG-SVM-GDTW	**3.8 %** **3.7 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	SDTW [BB01]	**4.5 %** **3.2 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	MLP [PLG01]	3.0 %	DevTest-R02/V02
	HMM [HLB00]	3.2 %	Train-R01/V06 4 % "bad characters" removed
1b (upper case)	DAG-SVM-GDTW	**7.4 %** **7.3 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	SDTW [BB01]	**10.0 %** **8.0 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	HMM [HLB00]	6.4 %	Train-R01/V06 4 % "bad characters" removed
1c (lower case)	DAG-SVM-GDTW	**11.5 %** **12.0 %**	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test
	SDTW [BB01]	**13.0 %** **11.4 %** 9.7 %	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test rand. chosen 67 %/33 % Train/Test
	MLP [PLG01]	14.4 %	DevTest-R02/V02
	HMM-NN hybrid [GADG01]	13,2 %	Train-R01/V07
	HMM [HLB00]	14,1 %	Train-R01/V06 4 % "bad characters" removed

Error Rates of Multi-class Problems

UNIPEN section	Approach	Error rate E	UNIPEN Database Type
1a (digits)	DAG-SVM-GDTW	**3.8 %** **3.7 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	SDTW [BB01]	**4.5 %** **3.2 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	MLP [PLG01]	3.0 %	DevTest-R02/V02
	HMM [HLB00]	3.2 %	Train-R01/V06 4 % "bad characters" removed
1b (upper case)	DAG-SVM-GDTW	**7.4 %** **7.3 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	SDTW [BB01]	**10.0 %** **8.0 %**	Train-R01/V07 rand. chosen 20 %/20 % Train/Test rand. chosen 40 %/40 % Train/Test
	HMM [HLB00]	6.4 %	Train-R01/V06 4 % "bad characters" removed
1c (lower case)	DAG-SVM-GDTW	**11.5 %** **12.0 %**	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test
	SDTW [BB01]	**13.0 %** **11.4 %** 9.7 %	Train-R01/V07 rand. chosen 10 %/10 % Train/Test rand. chosen 20 %/20 % Train/Test rand. chosen 67 %/33 % Train/Test
	MLP [PLG01]	14.4 %	DevTest-R02/V02
	HMM-NN hybrid [GADG01]	13,2 %	Train-R01/V07
	HMM [HLB00]	14,1 %	Train-R01/V06 4 % "bad characters" removed