Offline Handritting Word Recognition

Thijs Kooi, Davide Modolo

January 27, 2011

Table of contents

Overview
Implementation Details
Results
Conclusions
Section no. 4

General
Dataset

Overview of the Project

Overview
Implementation Details
Results
Conclusions
Section no. 4

General
Dataset

Off-line handwriting recognition

- It involves the automatic conversion of text in an image into letter codes which are usable within computer and text-processing applications
- Off-line handwriting recognition is comparatively difficult, as different people have different handwriting styles
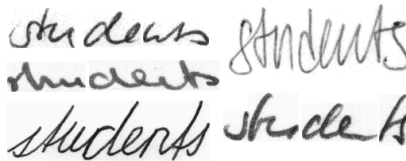

Figure: 'Students' written by different authors

Overview
Implementation Details
Results
Conclusions
Section no. 4

General
Dataset

Our AI Project

A lot of research has been done over the past years.

We explored the topic and implemented a full pipeline for the task. The research touched different fields:

- Data Collection
- Image Processing
- Features extraction
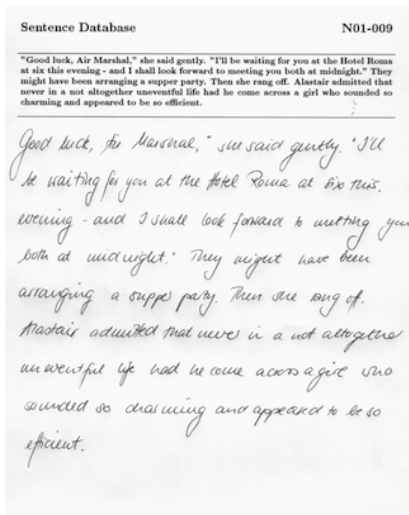- Machine Learning
- Word Recognition

Overview
Implementation Details
Results
Conclusions
Section no. 4

General
Dataset

Dataset

The IAM Handwriting Database 3.0[1]

- Unconstrained handwritten text (scanned at a resolution of 300dpi and saved as PNG images with 256 gray levels)
- 1'539 pages of scanned text of 657 writers
- 13'353 isolated and labeled text lines
- 115'320 isolated and labeled words

---

[1]http://www.iam.unibe.ch/fki/databases/iam-handwriting-database

Overview
Implementation Details
Results
Conclusions
Section no. 4

General
Dataset

Example of a page of scanned text

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Implementation Details

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model
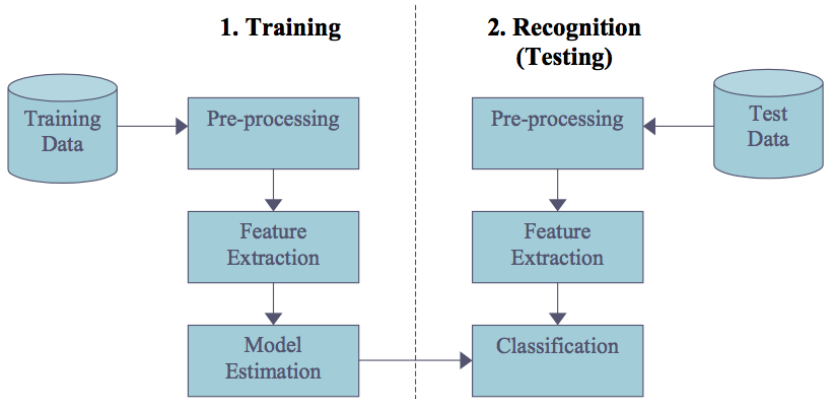
Pipeline

Figure: Pipeline of a word recognition system

Implementation Details

Pre-Processing

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model
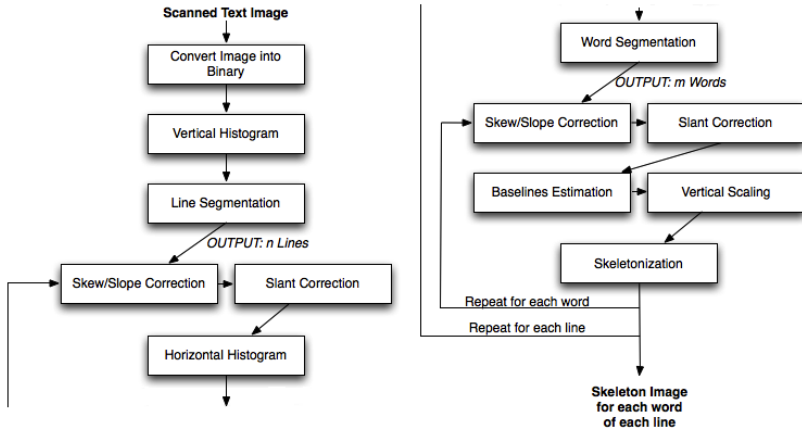
Pre-processing

Figure: Pipeline for the pre-processing/normalization step

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Line Segmentation

Original image segmented in lines   Vertical histograms and significant minima
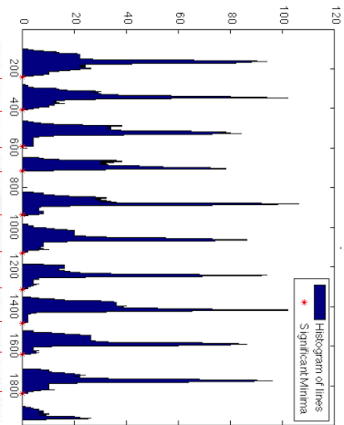
Figure: Example of line segmentation

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Skew and Slope Correction

Original Image

now put forward nominees. He believes

Linear Regression on the Lower Pixels and Skew Detection

Line Correction (angle: 0.0855)

now put forward nominees. He believes
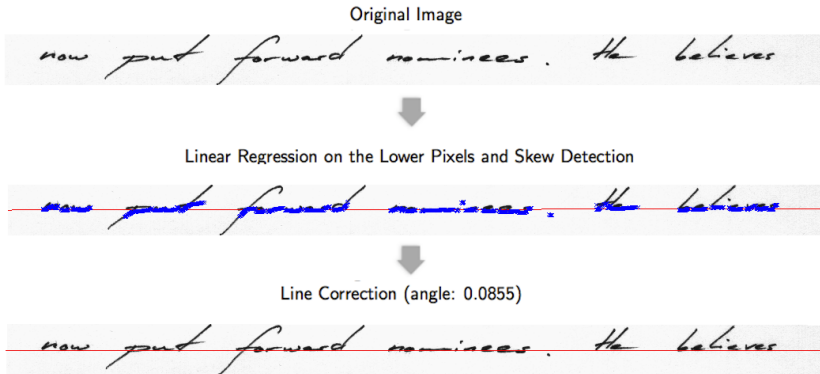
Figure: Skew detection and correction pipeline

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Slant Correction

Image after skew correction

Slant Detection

Slant Correction

Figure: Slant detection and correction pipeline

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Word segmentation

Horizontal Histogram
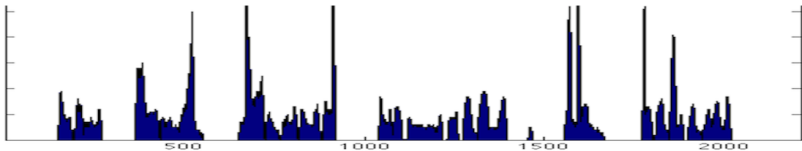
White Spaces Distribution
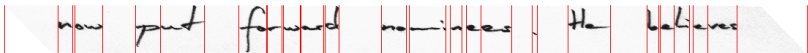
Clustering and Words Segmentation

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model
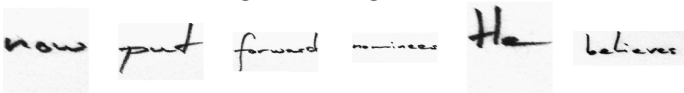
Baseline Estimation
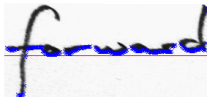
Lower Baseline

Upper Baseline
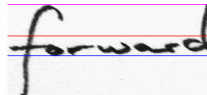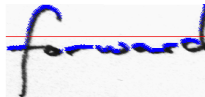
Ascender and Descender Baselines
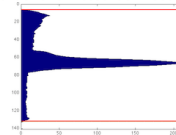
Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model
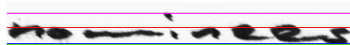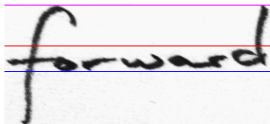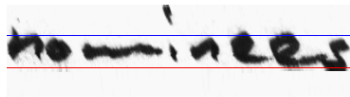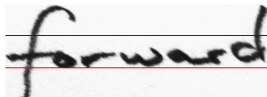
Vertical Scaling

Figure: Examples of vertical scaling process

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model
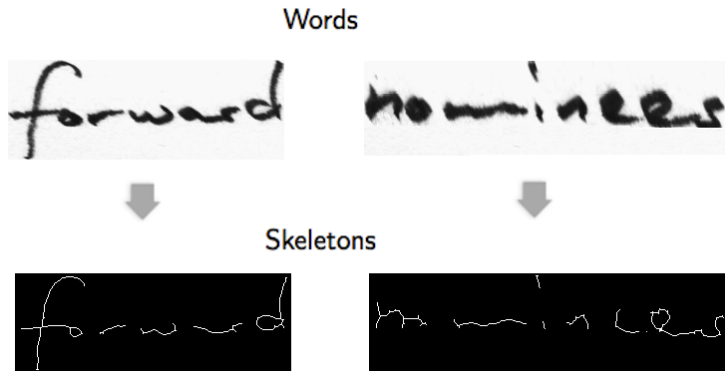
Skeletonization

Figure: Skeletonization process

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Remembering the entire pipeline....

Why repeating skew and slant correction twice?

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model
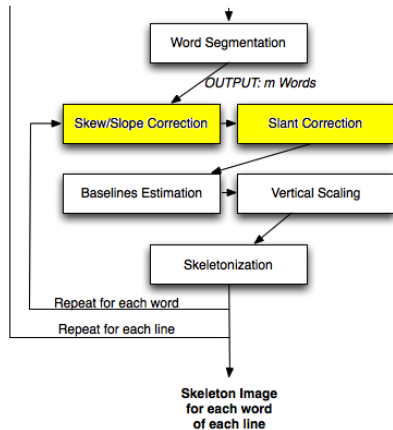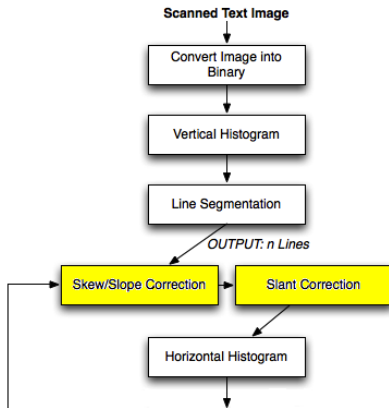
Because....

We need it before

and after

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Implementation Details

Feature Extraction

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Features

We want to find features that minimise the within-class variability and maximise the between class variability. On top of this, the features should be robust againts distortions caused by different handwriting styles. Moreover, we want to find low dimensional featur vectors and would therefore like features to be highly descriptive. The selection of features depends both on the pre-processing and the classifier to use. If all characters are assumed to have the same oriention, we need rotation variant features to distinguish between for instance a 6 and a 9 and a b and an p, etc.

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

HMM

- A set of N states $S = (s_1, s_2, \ldots, s_N)$, where the state of the system at time t is denoted q_t
- A set of priors $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$, providing the probability $P(q_1 = s_i)$.
- A transition function $\mathbf{A}$, where $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$.
- An observation function $\mathbf{B}$, mapping each observation at every state to a probability $b_i(\mathbf{o}_t) = P(\mathbf{o}_t | q_t = s_i, \lambda)$, where λ denotes the model parameters.

The model is trained to estimate the posterior probability $P(\mathbf{O}|\lambda)$ of an observation sequence $\mathbf{O}$, with D-dimensional observation vectors $\mathbf{o}_t = (o_1, o_2, \ldots, o_D)$.

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

HMM
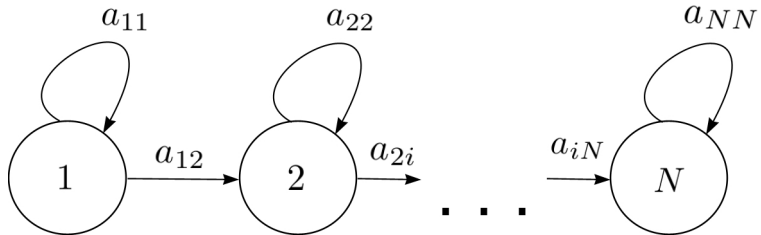
Figure: Left-to-right HMM with N states

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Main problems in an HMM

1. The probability of an observation sequence, given the model, $P(\mathbf{O}|\lambda)$.

2. The most likely parameters of the model $\lambda^* = \max P(X|\lambda)$, given a training set of M observation sequences $X = (\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_M)$.

3. The most likely state sequence, underlying a given observation sequence and the model, $Q^* = \max P(Q|\mathbf{O}, \lambda)$.

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Main problems in an HMM

1. The probability of an observation sequence, given the model, $P(\mathbf{O}|\lambda)$.

   Sum-product algorithm: forward-backward algorithm

2. The most likely parameters of the model $\lambda^* = \max P(X|\lambda)$, given a training set of M observation sequences $X = (\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_M)$.

   EM-algorithm: Baum-Welch reestimation

3. The most likely state sequence, underlying a given observation sequence and the model, $Q^* = \max P(Q|\mathbf{O}, \lambda)$.

   Dynamic programming: Viterbi algorithm
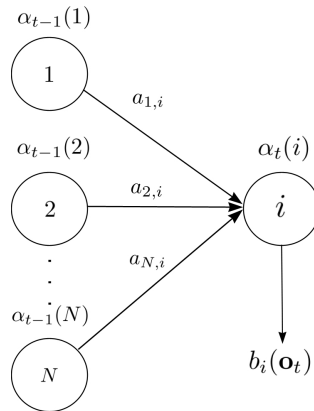
Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Forward probability

$$\alpha_t(i) \equiv$$
$$P(o_1, o_2, \ldots, o_t | q_t = s_i, \lambda) =$$
$$\Big[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ij} \Big] b_j(o_t)$$
(1)

Figure: Computation of forward probability

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Forward probability

$$\alpha_t(i) \equiv$$

$$P(o_1, o_2, \ldots, o_t | q_t = s_i, \lambda) =$$

$$\Big[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ij} \Big] b_j(o_t)$$

$$(2)$$

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

$$(3)$$

Problem 1 solved.

Figure: Computation of forward probability

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model
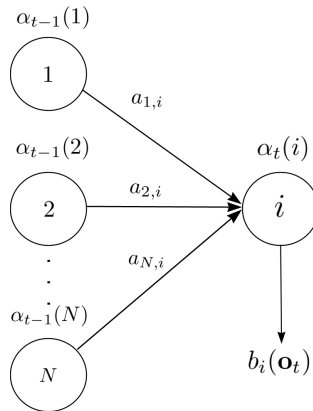
Learning the parameters

The forward-backward algorithm also commes with a backward probability:

$$\beta_t(j) \equiv P(o_{t+1}, o_{t+2}, \ldots, o_T | q_t = s_j, \lambda) =$$

$$\sum_{i=1}^{N} a_{ij} b_i(o_{t+1}) \beta_i(o_{t+1}) \tag{4}$$

With this we can define the probability of being in a state at a timestep as:

$$\gamma_t(i) \equiv (P(\mathbf{O}|\lambda))^{-1} \alpha_t(i)\beta_t(i) \tag{5}$$

Where normalisation constant
$P(\mathbf{O}|\lambda) = \sum_{j=1}^{N} \alpha_t(i)\beta_t(i) = \sum_{j=1}^{N} \alpha_T(j)$.

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Learning the parameters

$\hat{a}_{ij} = frac$Prob. of being in i and transfering to jProb. of begin in i $=$

$$\hat{a}_{ij} = \frac{\displaystyle\sum_{t=1}^{T-1} \xi_t(i,j)}{\displaystyle\sum_{t=1}^{T-1} \gamma_t(i)} \qquad (6)$$

The priors remain fixed, for the left-to-right model.

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Updating the parameters

Comparison with GMM:

Model:	**GMM**	**HMM**
Model parameters:	$\lambda = \pi, \mu, \Sigma$	$\lambda = \pi, \mathbf{A}, \mathbf{B}$
Hyper parameters:	Number of components	Topology (states, transitions), observation function
Observed variables:	Data points	Observations
Latent variables:	Priors of a component	State sequence

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Updating parameters

Model:	**GMM**	**HMM**
E-step:	Estimate the probability of a component, given the data and current parameters.	Estimate the probability of being in a state at a timestep and the probability of transfering from a state to another state.
M-step:	Maximise π, μ and Σ.	Maximise π, **A** and **B**

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Problems we ran into

Consider the following observation:

$$\mathbf{O} = \begin{pmatrix} -1 & 0 \\ 2 & 0 \\ 1 & 0 \\ -2 & 0 \end{pmatrix}$$

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Problems we ran into

Singularities. Consider the following observation:

$$\mathbf{O} = \begin{pmatrix} -1 & 0 \\ 2 & 0 \\ 1 & 0 \\ -2 & 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 3\frac{1}{3} & 0 \\ 0 & 0 \end{pmatrix}$$

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Problems we ran into

Singularities

$$\mathbf{O} = \begin{pmatrix} -1 & 0 \\ 2 & 0 \\ 1 & 0 \\ -2 & 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 3\frac{1}{3} & 0 \\ 0 & 0 \end{pmatrix}$$

$$|\Sigma| = 0$$

Possible solution: Add some random noise.

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Problems we ran into

Singularities

$$\mathbf{O} = \begin{pmatrix} -1 & 0 \\ 2 & 0 \\ 1 & 0 \\ -2 & 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 3\frac{1}{3} & 0 \\ 0 & 0 \end{pmatrix}$$

-¿ Add some random noise, also to prevent variance from collapsing.

Overview
Implementation Details
Results
Conclusions
Section no. 4

Pipeline
Pre-Processing
Feature Extraction
Hidden-Markov Model

Problems we ran into

Short words have less likelihood.
Harder to recognise, more subject to writer variations.
tried to solve this by using MOG.

Results

blabla

Conclusions

blabla

Overview
Implementation Details
Results
Conclusions
Section no. 4

blocs

blocs

title of the bloc

bloc text

title of the bloc

bloc text

title of the bloc

bloc text