Overview Implementation Details Results Conclusions Section no. 4

Offline Handritting Word Recognition

Thijs Kooi, Davide Modolo

January 27, 2011



Table of contents

- Overview
 - General
 - Dataset
- 2 Implementation Details
 - Pipeline
 - Pre-Processing
 - Feature Extraction
 - Hidden-Markov Model
- Results
- 4 Conclusions
- 5 Section no. 4
 - blocs



Overview of the Project

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

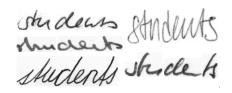


Figure: 'Students' written by different authors



Our Al 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



Dataset

The IAM Handwriting Database 3.0¹

- 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 1 1 2 2

Example of a page of scanned text

Sentence Database

N01-009

"Good lack, Air Marshal," she said geath. "The waiting for you at the Hotel Roma at six this evening - and I shall look forward to meeting you both at midnight." They might have been arranging supper party. Then she rang off. Alastair admitted that never in a not altogether uneventful life had be come across a girl who sounded so charming and appeared to be so efficient.

Good buck, to Marshal, " sur said gently "I'll be Haiting for you at the total Roma at the time the Roma at the total wind you both at midwight." They aright have been alranging a supple pary. Then the long of that a decorate an area of the had we can a give in a sure of the world so chasting and appeared to be so efficient.

Overview Implementation Details Results Conclusions Section no. 4

Pipeline Pre-Processing Feature Extraction Hidden-Markov Model

Implementation Details

Pipeline

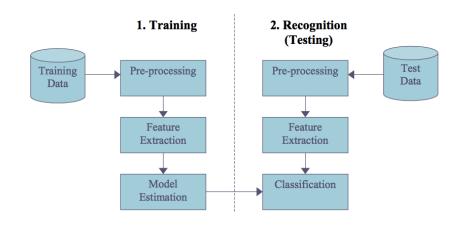


Figure: Pipeline of a word recognition system

Pipeline Pre-Processing Feature Extraction Hidden-Markov Model

Implementation Details

Pre-Processing

Pre-processing

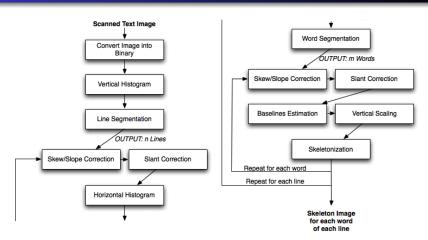
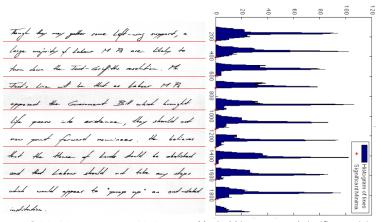


Figure: Pipeline for the pre-processing/normalization step

Line Segmentation



Original image segmented in lines

Vertical histograms and significant minima

Skew and Slope Correction

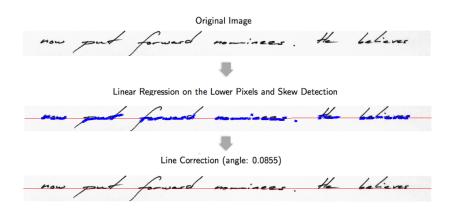


Figure: Skew detection and correction pipeline



Slant detection and Correction

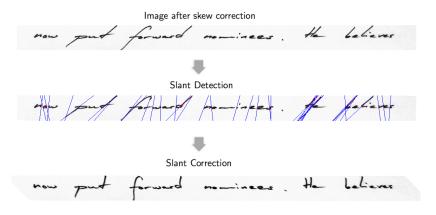
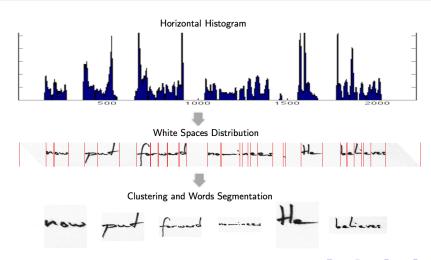


Figure: Slant detection and correction pipeline

Word segmentation

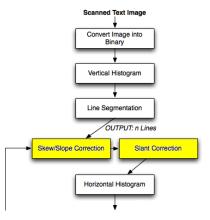


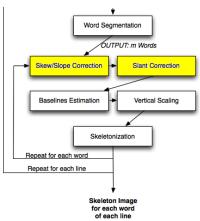
Vertical Scaling

Skeletonization

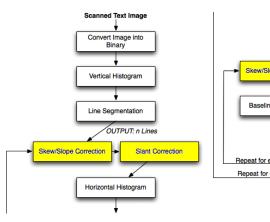
Remembering the entire pipeline....

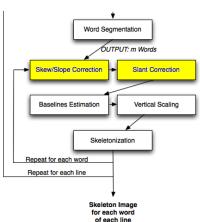
Why do we repeat skew and slant corraction twice?





Because....





Pipeline Pre-Processing Feature Extraction Hidden-Markov Model

Implementation Details

Feature Extraction

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.

HMM

- A set of N states $S = (s_1, s_2, ..., s_N)$, where the state of the system at time t is denoted q_t
- A set of priors $\pi = (\pi_1, \pi_2, \dots, \pi_N)$, providing the probability $P(q_1 = s_i)$.
- A transition function **A**, where $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$.
- An observation function **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 $\mathbf{o}_t = (o_1, o_2, \dots, o_D)$.



HMM

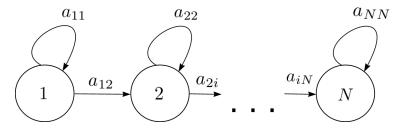


Figure: Left-to-right HMM with N states

Main problems in an HMM

- 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 $X = (\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_M)$.
- **3** The most likely state sequence, underlying a given observation sequence and the model, $Q^* = \max P(Q|\mathbf{0}, \lambda)$.

Main problems in an HMM

• 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 $X = (\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_M)$.

EM-algorithm: Baum-Welch reestimation

3 The most likely state sequence, underlying a given observation sequence and the model, $Q^* = \max P(Q|\mathbf{0}, \lambda)$.

Dynamic programming: Viterbi algorithm



Forward probability

$$\alpha_{t}(i) \equiv$$

$$P(o_{1}, o_{2}, \dots, o_{t} | q_{t} = s_{i}, \lambda) =$$

$$\left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ij}\right] b_{j}(o_{t})$$
(1)

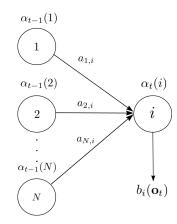


Figure: Computation of forward probability

Forward probability

$$\alpha_{t}(i) \equiv$$

$$P(o_{1}, o_{2}, \dots, o_{t} | q_{t} = s_{i}, \lambda) =$$

$$\left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ij}\right] b_{j}(o_{t})$$

$$(2)$$

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

$$(3)$$

Problem 1 solved.

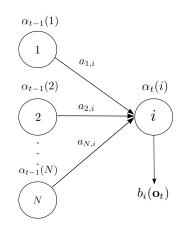


Figure: Computation of forward probability

Updating the parameters

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

$$\beta_{t}(j) \equiv P(o_{t+1}, o_{t+2}, \dots, o_{T} | q_{t} = s_{j}, \lambda) = \sum_{i=1}^{N} a_{ij} b_{i}(o_{t+1}) \beta_{i}(o_{t+1})$$

$$(4)$$

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

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

Where normalisation constant

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

Updating the parameters

Comparison with GMM:

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

Updating parameters

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

Consider the following observation:

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

Singularities. Consider the following observation:

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

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

Singularities

$$\mathbf{O} = \begin{pmatrix} -1 & 0 \\ 2 & 0 \\ 1 & 0 \\ -2 & 0 \end{pmatrix}$$
$$\mathbf{\Sigma} = \begin{pmatrix} 3\frac{1}{3} & 0 \\ 0 & 0 \end{pmatrix}$$
$$|\mathbf{\Sigma}| = \mathbf{0}$$

Possible solution: Add some random noise.

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.

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

Results

blabla



Conclusions

blabla



blocs

title of the bloc

bloc text

title of the bloc

bloc text

title of the bloc

bloc text