W4240 Data Mining

Frank Wood

September 6, 2010

Introduction

- Data mining is the search for patterns in large collections of data
  - Learning models
  - Applying models to large quantities of data
- Pattern recognition is concerned with *automatically* finding patterns in data / learning models
- Machine learning is pattern recognition with concern for computational tractability and full automation
- Data mining = Machine Learning = Applied Statistics
  - Scale
  - *Computation*

Example Application: ALARM, expert diagnostic system

Goal: Inference in given/known/hand-specified Bayesian network

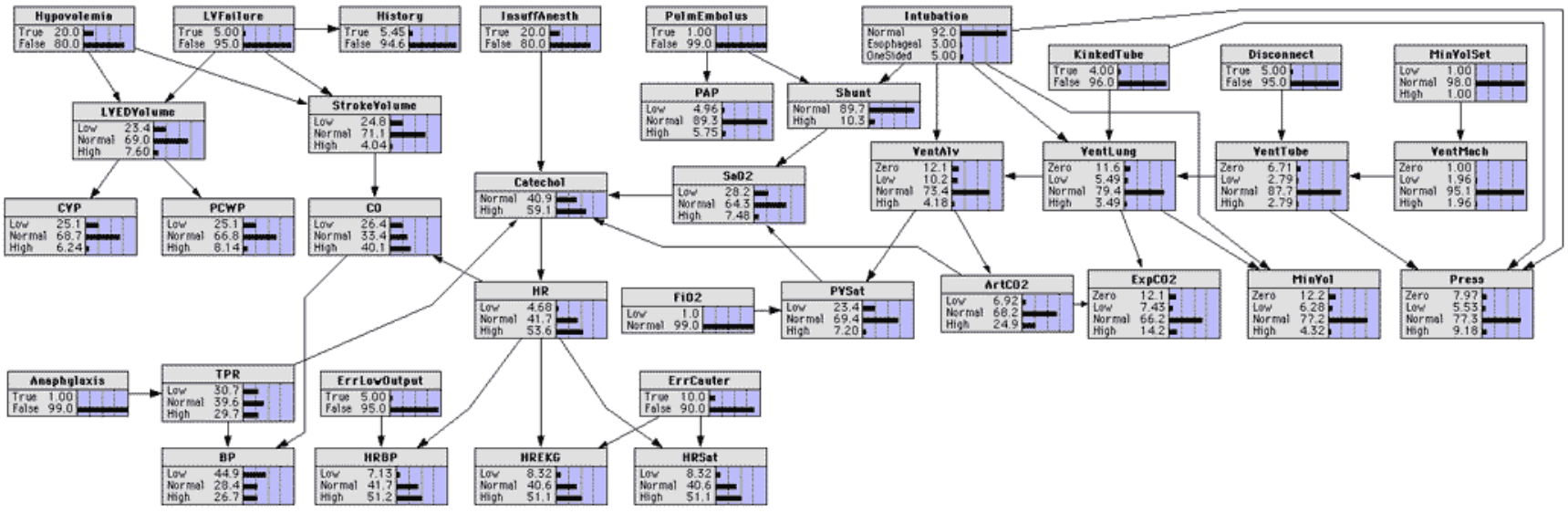

Figure: ALARM stands for 'A Logical Alarm Reduction Mechanism'. This is a medical diagnostic system for patient monitoring. It is a nontrivial belief network with 8 diagnoses, 16 findings and 13 intermediate variables. Described in [2]

Graphical Models

- ALARM network and most other probabilistic models can be expressed in the "language" of graphical models.
- Inference procedures such as the sum-product algorithm and belief propagation are general inference techniques that can be run on *any* discrete or linear-Gaussian graphical model.

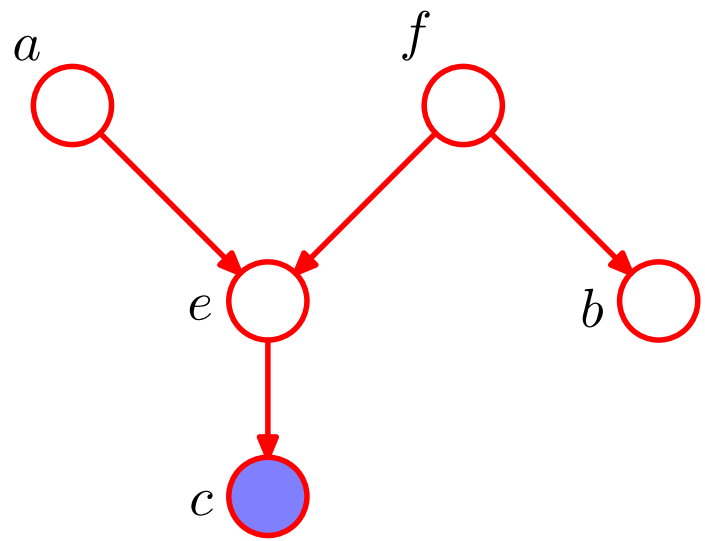

Figure: Directed Graphical Model : Chapter 8, Figure 22a, PRML [3]

Graphical Models Cont.

Results

- Ability to compute marginal distribution of any subset of variable in the graphical model conditioned on any other subset of variables (values observed / fixed)
- Generalizes many, many inference procedures such as Kalman filter, forward-backward, etc.
- Can be used for parameter estimation in the case where all latent, unknown variables are "parameters" and all observations are fixed, known variables.

Another Application: Classification of handwritten digits

Goal

- Build a machine that can identify handwritten digits automatically

Approaches

- Hand craft a set of rules that separate each digit from the next
- Set of rules invariably grows large and unwieldy and requires many "exceptions"
- "Learn" a set of models for each digit automatically from labeled training data, i.e. *mine* a large collection of handwritten digits and produce a model of each
- Use model to do classification

Formalism

- Each digit is 28x28 pixel image
- Vectorized into a 784 entry vector $\mathbf{x}$

Handwritten Digit Recognition Training Data

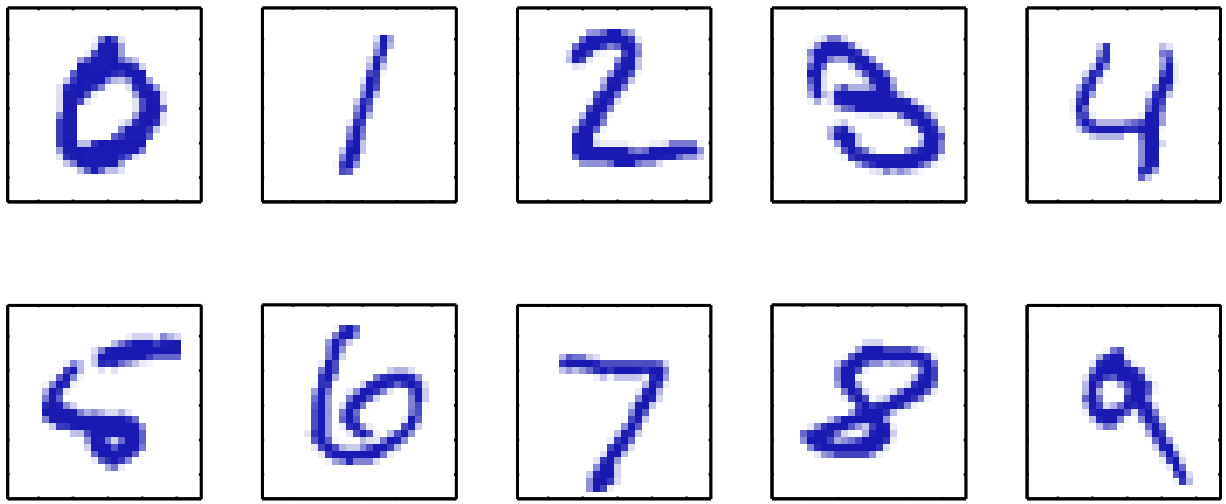

Figure: Hand written digits from the USPS

Machine learning approach to digit recognition

Recipe

- Obtain a of N digits $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ called the *training set*.
- Label (by hand) the training set to produce a label or "target" $\mathbf{t}$ for each digit image $\mathbf{x}$
- Learn a function $\mathbf{y}(\mathbf{x})$ which takes an image $\mathbf{x}$ as input and returns an output in the same "format" as the target vector.

Terminology

- The process of determining the precise shape of the function $\mathbf{y}$ is known as the "training" or "learning" phase.
- After training, the model (function $\mathbf{y}$) can be used to figure out what digit unseen images might be of. The set comprised of such data is called the "test set"

Tools for the handwriting recognition job

Supervised Regression/Classification Models

- Logistic regression
- Neural networks
- Support vector machines
- Naive Bayes classifiers

Unsupervised Clustering

- Gaussian mixture model

Model Parameter Estimation

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

Example Application: Trajectory Inference From Noisy Data

Goal

- Build a machine that can uncover and compute the true trajectory of an indirectly and noisily observed moving target

Approaches

- Hand craft a set of rules that govern the possible movements of said target
- Set of rules invariably grows large and unwieldy and requires many "exceptions"
- "Learn" a model of the kind of movements such a target can make and perform inference in said model

Formalism

- Example observed trajectories $\{\mathbf{x}_n\}_{n=1}^N$
- Unobserved latent trajectories $\{\mathbf{z}_n\}_{n=1}^N$

Latent trajectory Inference

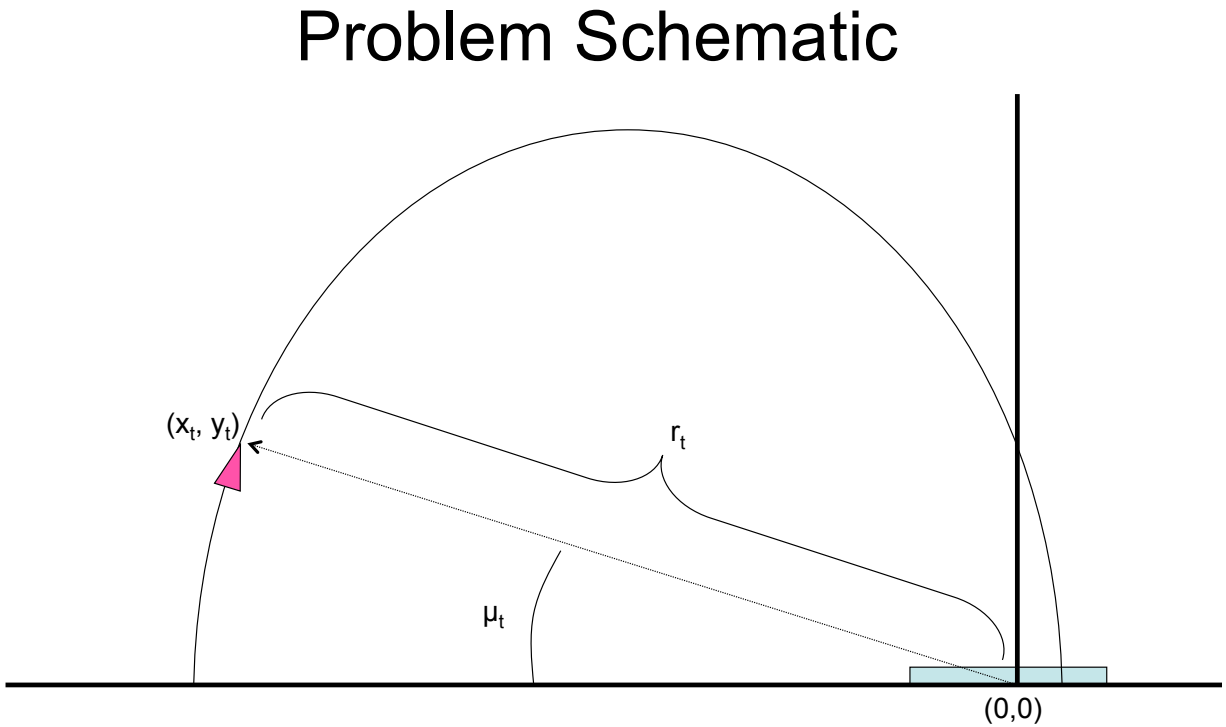

Figure: Schematic of trajectory inference problem

Tools for Latent Trajectory Inference

Known/hand-crafted model, inference only

- Belief propagation
- Kalman filter
- Particle filter
- Switching variants thereof
- Hidden Markov Models

Learning too / Model Parameter Estimation

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

Trajectory need not be "physical," could be an economic indicator, completely abstract, etc.

Cool Trajectory Inference Application : Neural Decoding

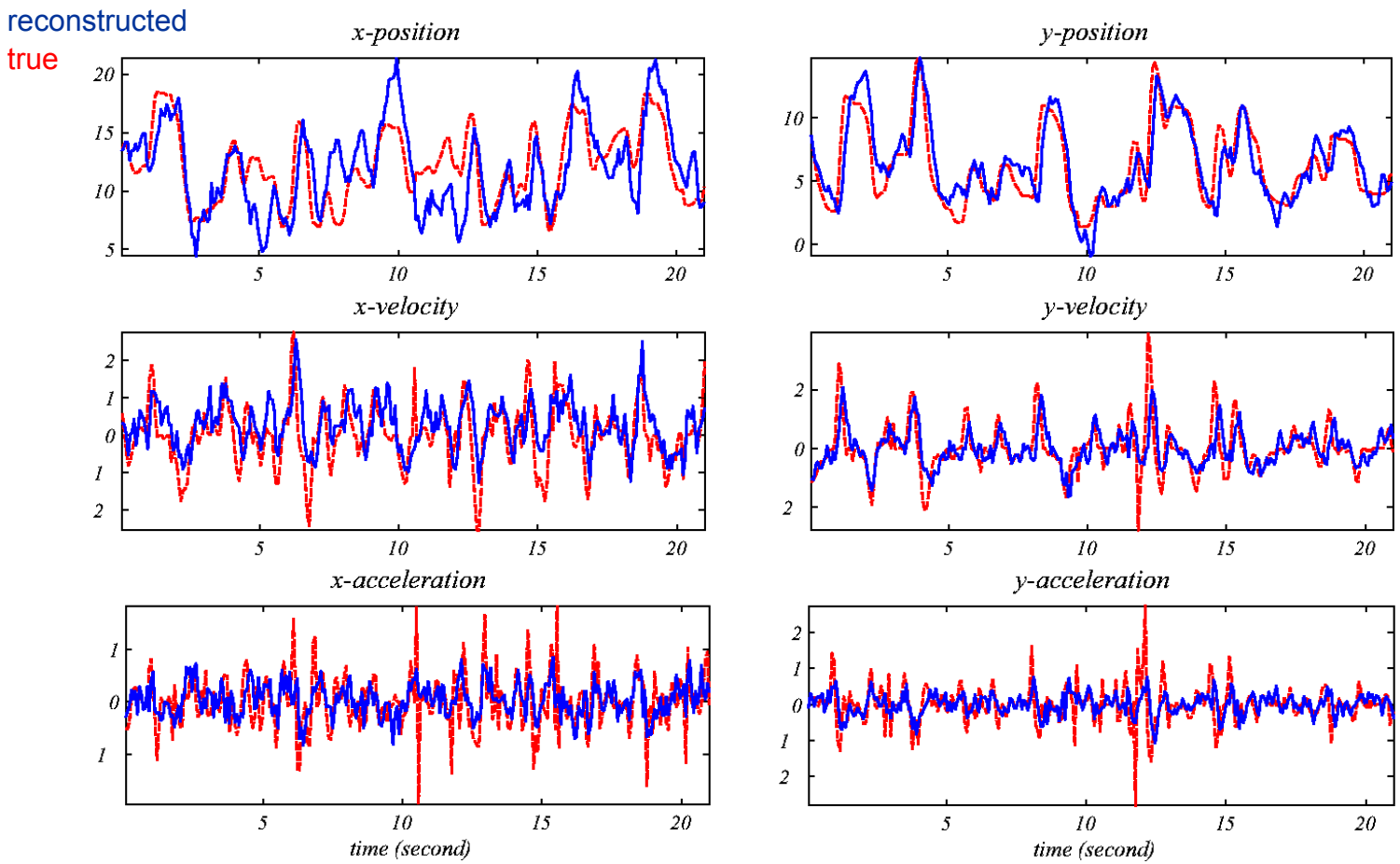

Figure: Actual and predicted hand positions (predicted from neural firing rates alone using a Kalman filter) [5]

Another Application: Unsupervised Clustering

Forensic analysis of printed documents, infer printer used to print document from visual features.

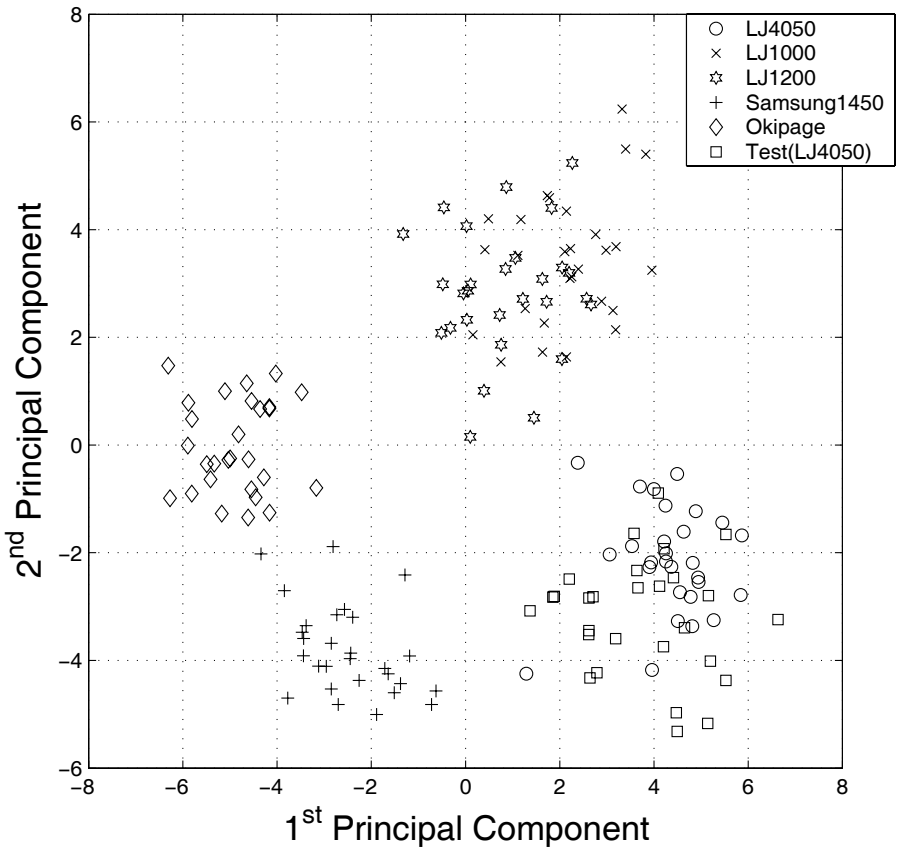

Figure: PCA projection of printer features [1]

Another Unsupervised Clustering Application

Automatic discovery of number of neurons and assignment of waveforms to neurons. Essential to electrophysiological study of the brain.

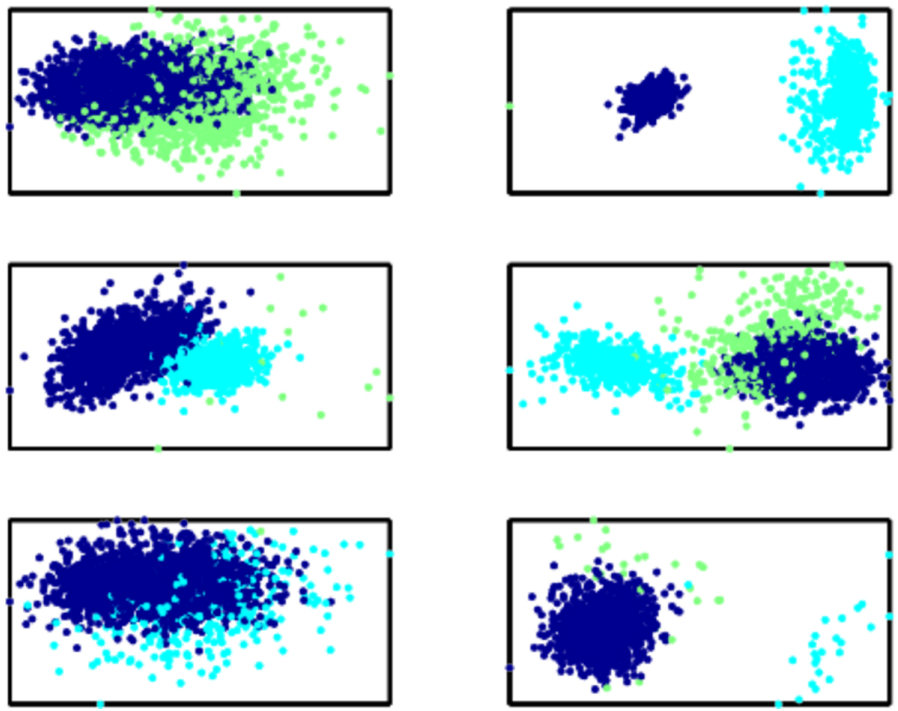

Figure: Automatically sorted action potential PCA projections [4]

A Big Unsupervised Clustering Application

Multinomial mixture model automatic document clustering for information retrieval.

$$\begin{aligned} z_n|\boldsymbol{\pi} &\sim \text{Discrete}(\boldsymbol{\pi}) \\ \mathbf{x}_n|z_n = k, \boldsymbol{\Theta} &\sim \text{Multinomial}(\boldsymbol{\theta}_{z_n}) \end{aligned}$$

where $\mathbf{x}_n$ is a bag of words or feature representation of a document, z_n is a per document class indicator variable, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k\}_{k=1}^K$ is a collection of probability vectors over types (or features) (per cluster k), and $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K], \sum_k \pi_k = 1$ is the class prior.

Such a model can be used to cluster similar documents together for information retrieval (Google, Bing, etc.) purposes.

Tools for Unsupervised Clustering

Known/hand-crafted model, inference only

- K-means
- Gaussian mixture models
- Multinomial mixture models

Learning too / Model Parameter Estimation

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

Tools for All

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

Links and Syllabus

Course home page :
http://www.stat.columbia.edu/ fwood/w4240/
Guest lectures may be sprinkled throughout the course.

Prerequisites

- Linear Algebra
- Multivariate Calculus (Matrix and Vector calculus)
- Probability and Statistics at a Graduate Level
- Programming experience in some language like pascal, matlab, c++, java, c, fortran, scheme, etc.

Good idea to familiarize yourself with PRML [3] Chapter 2 and Appendices B,C,D, and E.
In particular

- Multivariate Gaussian distribution
- Discrete, Multinomial, and Dirichlet distributions
- Lagrange Multipliers
- Matlab

Bibliograpy I

[1] G.N. Ali, P.J. Chiang, A.K. Mikkilineni, G.T.C. Chiu, E.J. Delp, and J.P. Allebach. Application of principal components analysis and gaussian mixture models to printer identification. In *Proceedings of the IS&Ts NIP20: International Conference on Digital Printing Technologies*, volume 20, pages 301–305. Citeseer, 2004.

[2] I. Beinlich, H.J. Suermondt, R. Chavez, G. Cooper, et al. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. 256, 1989.

[3] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.

[4] F. Wood and M. J. Black. A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, page to appear, 2008.

Bibliograpy II

[5] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, and J. P. Donoghue. Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter. In *SAB'02-Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices*, pages 66–73, August 2002.