

3rd Year Project Final Report

SleepAppnea

George Cochrane, Sachin Mylavarapu, Tuan Anh Le, Sophie Louth

March 24, 2014

Abstract

The abstract text goes here.

1 Introduction

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.1 Subsection Heading Here

Write your subsection text here. Add something. Add something else.

2 Medical Information

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3 Rationale

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

4 The app

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5 Design Process

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5.1 Questionnaire selection

5.2 Signal Analysis - simple method

5.3 Machine Learning – Theory

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data [4]. The supervised classification problem in machine learning concerns finding the unknown target function that classifies certain input data into classes based on some set of training examples containing labelled input data.

In our case, given a sampled sound signal of a sleeper, we want to identify apnoea periods. In particular, we represent our input (sampled sound signal) as $\{s_1, s_2, \dots, s_T\} \equiv \{s_i\}_{i=1}^T$, and we want to output the classifiers for every K samples, $\{y_i\}_{i=1}^{T/K}$, where the classifier $y_i \in \{0, 1\}$ corresponds to samples $\{s_j\}_{j=(i-1)K+1}^{iK}$ of the signal. We assume that T is divisible by K , however if this is not the case, we discard an appropriate number of signal samples to make it so.

We have researched three models for our problem which we will discuss in this section. Firstly, we will discuss Support Vector Machines which are one of the most widely used algorithms in Machine Learning today, then we will discuss the State-Space and Hidden Markov Models, which turn out to be more well-suited to our problem due to their temporal nature.

Support Vector Machines

Here, we present the theory for Support Vector Machines (SVMs) based on the lecture notes from Prof. Andrew Ng [3]. SVMs are one of the most widely used and many argue among the best "off-the-shelf" supervised learning algorithms. This is mainly due to the sound theoretical framework, efficiency and good generalisation guarantees even for high-dimensional and linearly non-separable data.

Notation. Having m training examples, where

- $\mathbf{x}^{(i)} \in \mathbb{R}^d$ is the d -dimensional i -th training data.
- $y^{(i)} \in \{-1, 1\}$ is the i -th training label.

we want to find the parameters $\mathbf{w} \in \mathbb{R}^d$ which describe the hyperplane $\mathbf{w}^T \mathbf{x} + b = 0$ that separates our two classes. Thus, we can define our classifier as $h_{\mathbf{w},b}(\mathbf{x}) = g(\mathbf{w}^T \mathbf{x} + b)$, such that $g(z) = 1$ if $z \geq 0$ and $g(z) = -1$ otherwise. Note that this is a non-probabilistic learning model as we are not considering the probability of each class or the data.

Objectives. The essence of SVMs lies in finding the decision boundary (hyperplane) which maximises the gap between the closest training points to it. For the i -th training point $x^{(i)}$, we call the “gap”, geometric margin $\gamma^{(i)}$, which is the distance to the decision hyperplane and can be found by considering a point \mathbf{x} on it, such that $\mathbf{w}^T \mathbf{x} = -b$:

$$\begin{aligned}\gamma^{(i)} &= \left| \frac{\mathbf{w}^T}{\|\mathbf{w}\|} (\mathbf{x}^{(i)} - \mathbf{x}) \right| \\ &= \frac{1}{\|\mathbf{w}\|} y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b)\end{aligned}$$

However, we only consider the closest point $\mathbf{x}^{(i^*)} : i^* = \arg \min_i \gamma^{(i)}$ with the geometric margin of $\gamma = \min_i \gamma^{(i)}$. Since scaling \mathbf{w} and b does not change the output of the classifier, nor the geometric margin $\gamma^{(i)}$, for convenience, we decide to scale \mathbf{w} and b such that $|\mathbf{w}^T \mathbf{x}^{(i^*)} + b| = y^{(i^*)} (\mathbf{w}^T \mathbf{x}^{(i^*)} + b) = 1$. Thus, maximising the geometric margin of the closest point becomes

$$\begin{aligned}\max_{\mathbf{w}, b} \quad & \frac{1}{\|\mathbf{w}\|} \\ \text{s.t.} \quad & y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \dots, m.\end{aligned}$$

which is equivalent to

$$\begin{aligned}\min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \dots, m.\end{aligned} \tag{1}$$

The solution to the optimisation problem (1), which can be found using off-the-shelf Quadratic Programming (QP) packages, gives us the parameters required to do classification.

Lagrange duality. While solving the optimisation problem (1) using QP solves the problem, we move on to derive the dual form of the problem using the principle of Lagrange duality, which will help us to solve the problem using a more efficient algorithm. More importantly, it will also allow us to efficiently transform the data points to high-dimensional spaces using the “kernel trick”, capturing the non-linear nature of the decision boundary without losing the generalisation guarantees. The Lagrangian of our primal minimisation problem (1) (with parameters $\boldsymbol{\alpha} \in \mathbb{R}^m$) can be formed as

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^m \alpha_i \left[y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b) - 1 \right] \tag{2}$$

To find the dual form of the problem, $W(\boldsymbol{\alpha}) = \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})$, we find the gradients of (2) with respect to \mathbf{w} and b and set them to zero to get

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y^{(i)} \mathbf{x}^{(i)} \quad (3)$$

$$\sum_{i=1}^m \alpha_i y^{(i)} = 0 \quad (4)$$

Substituting (3) and (4) back to (2) gives us the dual form, $W(\boldsymbol{\alpha})$

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle \quad (5)$$

Thus, under certain conditions which are in this case fulfilled, our original optimisation problem (1) becomes equivalent to the dual optimisation problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & W(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \dots, m \\ & \sum_{i=1}^m \alpha_i y^{(i)} = 0, \quad i = 1, \dots, m \end{aligned} \quad (6)$$

This problem can be solved using a QP algorithm, however a more efficient Sequential Minimal Optimization (SMO) algorithm can be used. Once the optimal value $\boldsymbol{\alpha}^*$ is obtained, we can find the corresponding parameters of the SVM \mathbf{w}^* using (3) and b^* using

$$b^* = - \frac{\max_{i:y^{(i)}=-1} \mathbf{w}^{*T} \mathbf{x}^{(i)} + \min_{i:y^{(i)}=1} \mathbf{w}^{*T} \mathbf{x}^{(i)}}{2} \quad (7)$$

$$= - \frac{\max_{i:y^{(i)}=-1} \sum_{j=1}^m \alpha_j^* y^{(j)} \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle + \min_{i:y^{(i)}=1} \sum_{j=1}^m \alpha_j^* y^{(j)} \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle}{2} \quad (8)$$

Thus the classification of a test point \mathbf{x} can be done by evaluating the argument of $g(\mathbf{w}^{*T} \mathbf{x} + b^*)$

$$\mathbf{w}^{*T} \mathbf{x} + b^* = \sum_{i=1}^m \alpha_i^* y^{(i)} \langle \mathbf{x}^{(i)}, \mathbf{x} \rangle + b^* \quad (9)$$

We note that due to certain conditions (Karush-Kuhn-Tucker conditions) the α_i^* 's that are non-zero correspond to the points $\mathbf{x}^{(i)}$ that lie on the margin. We call these points the Support Vectors (SVs).

Kernel trick. Note that according to (9) (and (8)), we only need to evaluate the inner products of \mathbf{x} and the support vectors in order to classify the point \mathbf{x} . This fact is used in the “kernel trick”, in which a kernel function $K(\mathbf{x}, \mathbf{z})$ is used as a proxy for the inner product $\langle \mathbf{x}, \mathbf{z} \rangle$. This effectively simulates transforming the data points to another, possibly unknown, space via a transformation function $\phi(\cdot) : \mathbb{R}^d \rightarrow \mathcal{Z}$ as long as there exists such \mathcal{Z} , i.e. $\phi(\cdot)$ such that $K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z})$. It is shown that a kernel $K(\cdot, \cdot)$ is valid if and only if it satisfies Mercer’s conditions. This allows us to transform our input domain into much higher dimensional domains without actually doing so which turns out to improve the time complexity significantly.

$$K_{\text{poly}}(\mathbf{x}, \mathbf{z}) = (a\mathbf{x}^T \mathbf{z} + c)^k \quad (10)$$

$$K_{\text{rbf}}(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \quad (11)$$

In our application, we are going to use the above kernels. The first one, (10), is the polynomial kernel which effectively transforms the input domain space into a feature space whose features are products all possible permutations of the input features of degrees up to k . The second one, (11), is a radial basis kernel (RBF), whose corresponding transformation function transforms the input domain space to an infinite dimensional one. This kernel gives a large inner product to two points if they are close to each other.

Soft margin SVMs. Soft margin SVMs is a slight modification of the original problem that allows small violations of the margin. The purpose is to guarantee that the algorithm doesn’t fail because of the non-separability of the input data even after using the kernel trick. In practice, these two techniques are used simultaneously. It turns out that only minor changes arise and the SMO algorithm can still be used.

Our problem. In our problem, we are going to use the SVM package in MATLAB[®] in order to train our model and classify sleep apnoea. The package includes implementations of both the QP and SMO algorithms, as well as various kernels, including the polynomial kernel and the RBF kernel. Recall that in our problem, we have the signal $\{s_i\}_{i=1}^T$ and the classifiers for every K samples $\{y_i\}_{i=1}^{T/K}$, where $y_i \in \{0, 1\}$ classifies the time period represented by $\{s_j\}_{j=(i-1)K+1}^{iK}$. When using SVMs, the input vectors and the

classifiers become

$$\mathbf{x}^{(i)} = [s_{(i-1)K+1}, \dots, s_{iK}]^T$$

$$y^{(i)} = \begin{cases} 1 & \text{if } y_i = 1 \\ -1 & \text{otherwise.} \end{cases}$$

State-Space Models

Here, we present the State-Space Models (SSMs) (also known as Linear Dynamical Systems) based on Kevin Murphy's book [2, Chapter 18] and Prof. Zoubin Ghahramani's paper [1]. SSMs allow us to model systems that are dynamic. Unlike the SVMs, SSMs will model dependencies of the current state on the previous ones.

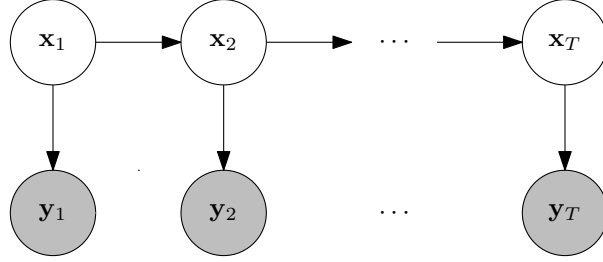


Figure 1: Probabilistic Graphical Model of a SSM

We have two set of random variables, the hidden states $\{\mathbf{x}_t, \mathbf{x}_t \in \mathbb{R}^d\}_{t=1}^T$ and the observed vectors $\{\mathbf{y}_t, \mathbf{y}_t \in \mathbb{R}^k\}_{t=1}^T$. We can represent SSMs graphically in Figure 1, and write out the joint probability of the model as

$$p\left(\{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^T\right) = p(\mathbf{x}_1)p(\mathbf{y}_1|\mathbf{x}_1) \prod_{t=2}^T p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{y}_t|\mathbf{x}_t) \quad (12)$$

In our case, we consider these particular transition and observation models with zero-mean Gaussian noises:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t \quad (13)$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t \quad (14)$$

$$\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \quad (15)$$

$$\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \quad (16)$$

Since we assume that $\pi = p(\mathbf{x}_1)$ is also Gaussian, all our conditional probability distributions will be

Gaussian due to the linearity of the transition and observation models. Once the parameters are obtained, the problem of inference and state estimation consists of

1. **Filtering.** We want to find $p(\mathbf{x}_t|\{\mathbf{y}_t\}_1^t)$. This is solved by the Kalman filtering algorithm.
2. **Smoothing.** We want to find $p(\mathbf{x}_t|\{\mathbf{y}_T\}_1^t)$. This is solved by the Kalman smoothing algorithm.
3. **Prediction.** We want to find $p(\mathbf{x}_{t+\tau}|\{\mathbf{y}_t\}_1^t)$. This is done by solving to 1. and then simulating using the state transition function.

We are mainly interested in filtering and smoothing.

Our problem. Although SSMs are well-suited for time-series data, they are not very well suited for our problem because the hidden state \mathbf{x}_t in this model is continuous, whereas in our problem, the sleeper's state is binary. While there are techniques to map from a continuous domain to a discrete one, we move on to discuss a more promising model.

5.3.1 Hidden Markov Models

5.3.2 Dimensionality Reduction

Spectrograms sdffasdf

Principal Components Analysis fdsafasd

5.4 Machine Learning - Experiments

5.5 Android

6 Conclusion

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

References

- [1] Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.
- [2] Kevin P Murphy. *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.
- [3] Andrew Ng. Cs229 - lecture notes. Course website, 2013.
- [4] Wikipedia. Machine learning — Wikipedia, the free encyclopedia, 2014. [Online; accessed 18-March-2014].