PREDICTION OF RELEVANT AREAS FOR DETECTION OF ABNORMALITIES IN X-RAY CHEST IMAGES

BY

IRIS LORENTE GOMEZ

Advisor: Dr. Jovan G. Brankov

Chicago, Illinois
2012-2013
ACKNOWLEDGMENT

This work could not have been done without the guidance of Dr. Brankov, who has given me advice and assistance every time I needed it. I also want to thank Centre de Formació Interdisciplinària Superior and the Medical Imaging Research Center for the financial support and confidence in me. I also want to acknowledge my coworkers and friends Oriol Caudevilla, Carles Figuerola, Francesc Massanés, Felipe Parages and Marc Lain for make me feel at home. Finally, I wish to thank Juan and my family members, especially my parents and my sister who always supported me in each decision.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>RESUM</td>
<td>viii</td>
</tr>
<tr>
<td>RESUMEN</td>
<td>ix</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>x</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>1. PRESENTATION</td>
<td>1</td>
</tr>
<tr>
<td>1.1. Basics about X-ray and projectional radiography</td>
<td>1</td>
</tr>
<tr>
<td>1.2. Problem definition and Objectives</td>
<td>5</td>
</tr>
<tr>
<td>1.3. Structure of the algorithm</td>
<td>7</td>
</tr>
<tr>
<td>2. X-RAY IMAGES TREATMENT: CONVERGENCE INDEX FILTERS AND WATERSHED SEGMENTATION</td>
<td>8</td>
</tr>
<tr>
<td>2.1. Convergence Index Filter</td>
<td>8</td>
</tr>
<tr>
<td>2.2. Using watershed transformation in filtered X-ray images</td>
<td>19</td>
</tr>
<tr>
<td>3. MACHINE LEARNING TECHNIQUES. RELEVANCE VECTOR MACHINE</td>
<td>22</td>
</tr>
<tr>
<td>3.2. Preliminars</td>
<td>23</td>
</tr>
<tr>
<td>3.3. Relevance Vector Machine for Regression</td>
<td>26</td>
</tr>
<tr>
<td>3.4. Advantages of RVM over SVM</td>
<td>31</td>
</tr>
<tr>
<td>4. FEATURE EXTRACTION AND SELECTION</td>
<td>33</td>
</tr>
<tr>
<td>4.1. Choosing features</td>
<td>36</td>
</tr>
<tr>
<td>4.2. Using Principal Components Analysis to reduce dimension of the feature space</td>
<td>39</td>
</tr>
<tr>
<td>5. RESULTS</td>
<td>42</td>
</tr>
<tr>
<td>5.1. Summary of the experiment design</td>
<td>42</td>
</tr>
</tbody>
</table>
5.2. Results and error estimation .......................... 43

6. CONCLUSIONS ............................................. 48

BIBLIOGRAPHY ................................................. 49
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Summary of the selected features</td>
<td>39</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Diagnosis of pneumonia in X-Ray chest image. Author: James Heilman, MD</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Diagnosis of tuberculosis in X-Ray chest image</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>Pleural effusion in X-Ray chest image. Author: James Heilman, MD</td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>Atelectasis in X-Ray chest image.</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>Pulmonary edema in X-Ray chest image. Author: James Heilman, MD</td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>Gazed points of observer1 in an image from the data set</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Region of support of the COIN filter.</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>Simplified region of support.</td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>Substantial region of support of the Iris Filter.</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>Image “Lena” filtered by IF with different values of $R_{min}$ and $R_{max}$.</td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>Chest X-ray image filtered by IF with different values of $R_{min}$ and $R_{max}$.</td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>Substantial region of support of the SBF.</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>Image “Lena” filtered by SBF with different values of $R_{min}$, $R_{max}$ and $d$.</td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>Chest X-ray image from the database filtered by SBF with different values of $R_{min}$ and $R_{max}$.</td>
<td></td>
</tr>
<tr>
<td>2.9</td>
<td>Chest X-ray image from the database filtered by SBF and segmented with watershed method.</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Example of SVM classification problem solution in 2D.</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Original and IF filtered images for feature extraction</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>SBF filtered images for feature extraction</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison of targets and predictions. Left: Target. Right: Prediction</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Comparison of targets and predictions. Left: Target. Right: Prediction</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>Confidence interval</td>
<td></td>
</tr>
</tbody>
</table>
RESUM

Aquest projecte pertany al camp d’imatge mèdica i està motivat per la idea d’elaborar una primera aproximació a la diagnosi mèdica automàtica. L’objectiu principal de l’estudi és, donada una radiografia de pit, predir quanta estona estaria un doctor observant parts específiques de la imatge. Aquesta predició ens proporciona una estimació sobre quines d’aquestes parts considera un doctor més relevant durant la diagnosi, les parts importants probablement estaran correlades amb el fet que hi hagi una anomalia.

Les prediccions les obtenim entrenant un model mitjançant l’algoritme de machine learning Relevance Vector Machine (RVM). S’ha escollit aquesta tècnica degut a les seves propietats bayesianes intrínseques, que ens proporcionen la distribució de probabilitat de la predicció. Conèixer la distribució ens és útil per determinar l’exactitud d’aquesta estimació.

El dataset usat per entrenar el model s’obté aplicant tècniques de seguiment de la mirada a doctors durant la diagnosi. Està format per les radiografies, les coordenades dels punts observats i la durada de la observació a casa punt.

Per obtenir bones features per entrenar el model es necessari segmentar i preprocessar les imatges. El preprocés es duu a terme aplicant fíltres d’índex de convergència (COIN) a les imatges. Això ens permet remarcar els nòduls clars detectant la seva vorera. A continuació, s’aplica una segmentació watershed per dividir els pulmons en regions d’intensitat similar i es tracten tots els pixels d’una regió com una única mostra. De cada regió, n’extraïm un conjunt de features i en reduim la dimensió del seu espai mitjançant un anàlisi de components principals. El RVM s’entrena amb aquest nou conjunt i les durades de les observacions.

Els resultats d’aquest estudi indiquen que el model es comporta prou bé en predir quines són les regions relevantes a les radiografies.
RESUMEN

Este proyecto pertenece al campo de la imagen médica y tiene la motivación de crear una primera aproximación al diagnóstico médico automático. El principal objetivo de este estudio es, dada una radiografía del pecho, predecir cuánto rato estaría mirando un doctor a determinadas partes de la imagen. Esta predicción da una estimación de cuál de esas partes es más relevante para el médico durante el diagnóstico, y puede estar correlacionado con la probabilidad de existencia de una anomalía.

Las predicciones se obtienen entrenando un modelo con un algoritmo de “machine learning” llamado Relevant Vector Machine (RVM). La elección de esta técnica se debe a las propiedades bayesianas intrínsecas al método que permiten obtener la distribución de probabilidad de la solución. Conocer esta distribución se usará para poder determinar la precisión de la estimación.

El dataset usado para enseñar al modelo se obtiene aplicando técnicas de seguimiento de la mirada a médicos durante el diagnóstico. Está compuesto por las radiografías, las coordenadas de los puntos observados y la duración de la mirada en cada punto.

Para obtener buenas “features” con las que entrenar el modelo es necesario segmentar las imágenes y someterlas a un preproceso. Durante el preproceso se aplican filtros de índice de convergencia (COIN) a las imágenes. Esto permite resaltar nódulos claros detectando sus bordes y diferenciándolos claramente de los píxeles a su alrededor. Después, se aplica segmentación “watershed” a las imágenes para dividir los pulmones en regiones de intensidad similar y tratar todos los píxeles de una región como una única muestra. De cada región se extraen un conjunto de features, que posteriormente es reducido por un análisis de componentes principales. Este resultado se usa junto al tiempo de observación para enseñar al RVM.

Los resultados indican que los métodos predicen cuáles son las zonas más relevantes en las radiografías.
ABSTRACT

This project belongs to the field of medical imaging and was motivated by the idea of elaborating a first approximation to automatic medical diagnosis. The principal objective of this study is, given an X-ray chest image, to predict how long a doctor would be looking to specific different parts of the image. This prediction provides an estimation of which of these parts are more relevant for a doctor during diagnosis, and might be correlated with the probability of having an abnormality.

Predictions are obtained by training a model with a machine learning algorithm called Relevant Vector Machine (RVM). The choice of this technique is due to its inner bayesian properties that provide the probability distribution of prediction. The knowledge of the distribution is used to determine the accuracy of the estimation.

The data set used to learn the model is obtained by applying eye tracking techniques to doctors during diagnosis. It is formed by X-ray images, coordinates of gazed points and gaze duration at each point.

In order to obtain good features to learn the model it is necessary to segment and preprocess the images. Preprocessing is done by applying Convergence Index Filters (COIN) to the images. This allows to remark bright nodules by enhancing their boundary and making it clearly different from neighborhood pixels. Then, watershed segmentation is applied to the images so as to divide the lungs in regions of similar intensity and treat all pixels in one region as a unique sample. From each region, a set of features is extracted. The size of this set is reduced by analysis of principal components and the result is used along with gaze durations to learn RVM.

Results indicate that the learned model has good performance in predicting which the relevant regions are in X-ray images.
CHAPTER 1
PRESENTATION

In this first chapter, we are going to define the problem and explain some basics of projectional radiography and X-Ray chest images interpretation. This chapter also contains the structure of the algorithm which will be detailed in next chapters focusing on its theoretical and practical principal characteristics.

1.1 Basics about X-ray and projectional radiography

X-rays are electromagnetic radiation with wavelength ranging between 0.01 to 10 nanometers and energies in the range of 100 eV to 100 keV. Due to their penetrating ability, X-rays are widely used to get an image from inside objects. The resulting image is called projectional radiography. As they are a form of ionizing radiation, they have enough energy to potentially remove electrons from an atom and have sufficient energy to penetrate human tissue.

Diagnostic X-ray are useful to detect abnormalities within the body because they are a painless, non-invasive way to help diagnose problems such as broken bones or tumors.

Projection radiography is capable to obtain an inside image of the human body by locating the human between an X-ray emissor and an X-ray receptor. These rays pass through the body and arrive to the receptor attenuated according to the crosswised part of the body. For example, those rays passing through the bones will reach the receptor weaker than rays crossing the heart. Dense objects absorb more energy and are laboriously crossed by X-ray. The remain beam is converted to light using a fluorescent screen which is then captured on photographic film.

To comprehend and interpret in a correct way an X-ray chest image, we must have awareness of how this attenuation is expressed in the film. As bones are the densest parts of the body, they will appear in color white. Tissue and air will appear in grey and black
respectively.

1.1.1 X-ray chest images for diagnosis.

In this section, we are going to present some diseases that can be detected in an X-ray chest image.

One of the most common lung’s diseases is pneumonia. It is an inflammatory condition of the lung usually caused by infection with viruses or drugs. In figure 1.1, we can appreciate the bright part caused by this disease.

![Figure 1.1: Diagnosis of pneumonia in X-Ray chest image. Author: James Heilman, MD](image)

Tuberculosis is another of the abnormalities that can be detected in chest radiographies. It is an infectious disease caused by *Mycobacterium tuberculosis*. Figure 1.2 shows its repercussion. Other diseases which need an X-ray chest image to be detected are: neoplasm, atelectasis, pulmonary edema, pulmonary hemorrhage, pulmonary embolism, interstitial pulmonary fibrosis... Some of them are represented in figures 1.3, 1.4 and 1.5.
Figure 1.2: Diagnosis of tuberculosis in X-Ray chest image

Figure 1.3: Pleural effusion in X-Ray chest image. Author: James Heilman, MD
Figure 1.4: Atelectasis in X-Ray chest image.

Figure 1.5: Pulmonary edema in X-Ray chest image. Author: James Heilman, MD
1.2 Problem definition and Objectives

This study was motivated by the idea of elaborating a first approximation to automatic medical diagnosis. The aim of this work is to learn a model able to imitate human behaviour by learning from them. Particularly, the main objective of this study is to predict how long a doctor would be looking to a specific part of a random X-ray chest image. From another point of view, to predict relevant areas in a X-ray chest image.

In order to achieve this goal, 8 doctors participated in an eye gaze study. In this study, they were shown 97 X-ray chest images and their gaze was recorded with eye tracking technologies. This gazing information was then processed and linked to the corresponding location in every image according to the size of the display and its distance from the observer.

Our data set is form, thus, by 97 X-ray chest images. Each one is accompanied by a list of coordinates of the points that have been gazed by the observers, and we also have the duration of this gazing. This data will be used to train a model able to predict the duration of gazed points in other images.

Another objective of this study is to find image processing algorithms that facilitate the detection of abnormalities in X-ray chest images and also to analyse the behavior and reliability of machine learning techniques applied to this field. Figure 1.6 show an image from the data set marked by the gazed points of one observer.
1.2.1 Image alignment.

If we pretend to predict the relevant gazed areas of a lungs radiography, we must have in consideration the position within the lungs of these gazed areas. However, each image belongs to different people and was taken by different radiologist. These facts caused the lungs not to be located at the same place in each radiography.

To solve this problem and facilitate the comparison of images using computer vision techniques, all the images were previously aligned and deformed using a thin plate splines algorithm.
1.3 Structure of the algorithm

The algorithm and also this report is structured in two stages:

1. Processing of images.

2. Extraction of features from the images and training a machine learning model.

Processing the images is applying them computer vision techniques so as to obtain new images with concrete characteristics. This first stage contains lungs segmentation and filtering of the images.

The next stage consists of extracting features from these processed images and learn a model using a learning machine technique called Relevance Vector machine.

Finally, the model is tested with new gazed images to analyse the performance of the algorithm.
CHAPTER 2
X-RAY IMAGES TREATMENT: CONVERGENCE INDEX FILTERS AND
WATERSHED SEGMENTATION

The detection of pulmonary nodules in chest radiography is one of the most studied problems in X-ray image analysis. Most of the time, the nodules are not easily visible, there are opacities and some of them have very weak contrast to their backgrounds, hence an effective method to enhance vague boundaries and edge detection techniques are required.

Along this chapter, there is an introduction to convergence index filters which evaluate the degree of convergence of gradient vectors within its region of support toward a pixel of interest, specifically the Iris Filter and the Sliding Band Filter are used to treat these X-ray chest images. Once this procedure has been accomplished, the image is processed with a watershed segmentation operator aiming at dividing the lungs in a set of nonoverlapping regions with similar intensity [1].

2.1 Convergence Index Filter

The convergence index of a gradient vector in a given pixel is a measure of how strongly this gradient vector point toward the pixel of interest in a neighborhood of it denoted by $R$. This strength is measured by computing the cosine of the gradient vector orientation with respect to the line connecting the pixel and the pixel of interest. As the information extracted is the cosine, the degree of convergence of the gradient vector is not related to the magnitude of the gradient vector but to the distribution of its directions.

The output of this type of filters is the average of convergence indices within its region of support, thus it belongs to the range $[-1, 1]$.

From now on, we are going to refer to Convergence Index Filter as COIN Filter.
2.1.1 Formal definition.

Let us denote the intensity of an image and its gradient vector at the point \((x, y)\) as 
\(I(x, y)\) and \(\mathbf{g}(x, y)\) respectively. If we denote the row component and the column component 
of the gradient at this point by \(G_R(x, y)\) and \(G_C(x, y)\), the magnitude of \(\mathbf{g}\) is given by 
\[
|\mathbf{g}(x, y)| = \sqrt{G_R(x, y)^2 + G_C(x, y)^2}
\] (2.1)

There are multiples methods to compute the gradient of an image in each point. In 
this study, \(G_R(x, y)\) and \(G_C(x, y)\) are obtained using a Prewitt-type \(3 \times 3\) operator, which 
is a discrete differentiation operator based on convolving the image with two \(3 \times 3\) kernels: 
one to measure approximations of horizontal derivatives and another for vertical derivatives. 
Thus, \(G_R(x, y)\) and \(G_C(x, y)\) are obtained as follows:
\[
G_R = I \ast \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad G_C = I \ast \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}
\]

Let us define the orientation of \(\mathbf{g}\) with respect to the row axis by 
\[
\phi(x, y) = \arctan \frac{G_R(x, y)}{G_C(x, y)}
\] (2.2)

As we are working with digital images, the two-dimensional space is discrete, hence 
each pixel is going to be referred by \((i, j)\) instead of \((x, y)\) from now on.

The convergence index is evaluated in a neighborhood of the pixel of interest and 
denoted by \(R\), also known as the region of support of the filter. \(R\) is a circle of radius \(r\) 
whose center is at the pixel of interest \(P\). Let \(Q\) be an arbitrary pixel in \(R\) with coordinates 
\((k, l)\). The angle \(\theta(k, l)\) is the orientation of the gradient vector \(\mathbf{g}(k, l)\) with respect to the 
direction of the line \(\overline{PQ}\). This angle \(\theta(k, l)\) becomes the convergence index of the gradient 
vector at pixel \((k, l)\). Fig. 2.1 shows the region of support of the convergence index filter 
and the angle \(\theta\).
The output of the COIN filter at the point \((k,l)\) is defined as follows:

\[
C(i,j) = \frac{1}{M} \sum_{(k,l) \in R} \cos \theta(k,l)
\]  

where \(M\) is the number of pixels in the region of support \(R\).

If \(C(i,j)\) reaches the maximum value +1, it means that all the gradient vectors in \(R\) point toward the pixel of interest \(P\), i.e. \(P\) is rounded by concentric circumferences of same intensity in the region \(R\).

In general, images are redundant, therefore we can simplify the region of support \(R\) as the union of \(N\) half-lines radiating from the pixel of interest and assume that the average of the indices in these half-lines is a good representation of \(C(i,j)\) if \(N\) is sufficiently large.

Thus the region of support becomes:

\[
R = \bigcup_{i=0}^{N-1} L_i
\]  

where the orientation of each radial half-line \(L_i\) with respect to the abscissa is \(\frac{2\pi i}{N}\).
Figure 2.2 shows the simplified region of support.

Iris Filter and Sliding Band Filter are two examples of COIN filters. In this work, the X-ray images are processed separately with these two filters using difference sizes of $R$ for each one in order to obtain some variations of the each X-ray image and be able to extract enough features.

2.1.2 Using Iris Filter in X-ray images.

The basic COIN filter is expected to work well if the objective is to detect a rounded convex region with a fixed size. However, the size of lungs tumors changes in a wide range; in fact, X-ray images can show tumors from different sizes starting at 1 square centimeter. For this reason, it is better to introduce the concept of adaptative COIN filter, as Iris Filter (IF). The main difference between IF and the basic COIN filter is that IF automatically adjusts the length of each radial half-line in the region of support as the iris of the eye changes its size adaptively to the brightness of the field of vision. This adjust aims to maximize the average of convergence indices in each radial direction and causes each pixel to have a different region of support [2]. Figure 2.3 shows
More specifically, let $P$ be the pixel of interest and let us define the possible region of support of $P$, $R_P$ as the region within a ring which radius are limited by $R_{\text{min}}$ and $R_{\text{max}}$, i.e., $R_P$ is formed by the pixels $Q$ that verify $R_{\text{min}} < ||P - Q|| < R_{\text{max}}$. $R_P$ is not the final region of support of pixel $P$ but it is necessary considering all the points of $R_P$ in order to decide the length of each half-line. In figure 2.3 we can observe the substantial region of support of IF.

Let us denote coordinates of the $m$th pixel from the pixel of interest on the $i$th half-line by $([x_{im}], [y_{im}])$. \[ \begin{align*}
  x_{im} &= k + m \cos \frac{2\pi}{N} i \\
  y_{im} &= l + m \sin \frac{2\pi}{N} i
\end{align*} \]

where $(k, l)$ are the coordinates of the pixel of interest $P$. [3]

For each half-line $L_i$ we define $C_i(n)$ as the average of the convergence indices from

\footnote{The symbol $[\cdot]$ refers to the floor function, i.e. it represents the maximum integer less than or equal to the real number.}
the 1st to the $n$th pixel on the $i$th half-line

$$C_i(n) = \frac{1}{n} \sum_{m=1}^{n} \cos \theta_{im}$$

(2.7)

and $C_{i\text{max}}$ as the maximum value of these averages

$$C_{i\text{max}} = \max_{R_{\text{min}} \leq n \leq R_{\text{max}}} C_i(n)$$

(2.8)

Finally, the output of the Iris Filter in discrete space is the average of the maximal convergence indices for the $N$ half radial directions

$$C(k, l) = \frac{1}{N} \sum_{i=0}^{N-1} C_{i\text{max}}$$

(2.9)

The angle $\theta_{im}$ in equation (2.7) is the orientation of the gradient vector with respect to the $i$th half-line at the $m$th pixel from the pixel of interest $P$. An aspect to be considered on implementation is the computation time of the convergence index, $\cos \theta_{im}$. It is time consuming so in this work, the orientation $\theta_{im}$ is quantized into one of eight levels and its cosine values are tabulated. This quantization does not cause huge errors because the number of pixels in the region of support is large and the output of the filter is given by averaging many convergences indices.

### 2.1.2.1 How to choose $R_{\text{min}}$ and $R_{\text{max}}$?

The election of parameters $R_{\text{min}}$ and $R_{\text{max}}$ depends on the size range of the nodules that we want to highlight. A larger value of $R_{\text{max}}$ will allow the filter to detect bigger shadows more accurately and will less likely detect small perturbations, whereas a small value of $R_{\text{max}}$ will emphasize small nodules. However, a value greater than 1 is chosen for $R_{\text{min}}$ to minimize errors caused by the noise of the image. In this way, pixels very closed to the pixel of interest are not considered and all averages are done at least with a number of pixels equal to $R_{\text{min}}$. Thus, a larger value of $R_{\text{min}}$ also collaborates to detect bigger nodules. Fig. 2.4 shows the behaviour of IF with different values of $R_{\text{min}}$ and $R_{\text{max}}$. 
(a) Original gray scale image

(b) $R_{\text{min}} = 5$, $R_{\text{max}} = 20$ and $N = 32$

(c) $R_{\text{min}} = 25$, $R_{\text{max}} = 60$ and $N = 32$

(d) $R_{\text{min}} = 1$, $R_{\text{max}} = 60$ and $N = 32$

Figure 2.4: Image “Lena” filtered by IF with different values of $R_{\text{min}}$ and $R_{\text{max}}$.

Fig. 2.5 shows the result from applying the iris filter with different parameters to an image from the data set. An accurate vision of nodules is shown on filtered images.
2.1.3 Using Sliding Band Filter in X-ray images.

Sliding Band Filter (SBF) also belongs to the family of convergence index filters, thus its output is an average of some convergence indices. The main difference between this filter and IF is that SBF searches in each radial direction the band of fixed width that corresponds to the maximum degree of convergence, while the region of support of an IF
always starts at the \( R_{\text{min}} \)th pixel in each half-line.

\[
\begin{align*}
C(k,l) &= \frac{1}{N} \sum_{i=0}^{N-1} \left( \max_{R_{\text{min}} \leq n \leq R_{\text{max}}} \left( \frac{1}{d} \sum_{m=n}^{n+d} \cos \theta_{im} \right) \right) \\
\end{align*}
\]

\( (2.10) \)

where \( d \) refers to the widthband of the substantial region of support. This width is fixed in each half-line but its position is variable depending on the convergence indices of each pixel on the half-line. Fig. 2.6 represents an example of the substantial region of support for a SBF.

If we compare the outputs of IF and SBF, the difference is that SBF has a more selective response for those nodules whose central region has a more random degree of convergence because it only considers the band of the nodule with the highest convergence indices.

As with the IF, different values of the parameters allow to enhance larger or smaller nodules. Figures 2.7 and 2.8 show that effect.

Figure 2.6: Substantial region of support of the SBF.
Figure 2.7: Image “Lena” filtered by SBF with different values of $R_{\text{min}}$, $R_{\text{max}}$ and $d$. 

(a) Original gray scale image

(b) $R_{\text{min}} = 10$, $R_{\text{max}} = 30$, $d = 8$ and $N = 32$

(c) $R_{\text{min}} = 5$, $R_{\text{max}} = 15$, $d = 8$ and $N = 32$

(d) $R_{\text{min}} = 5$, $R_{\text{max}} = 15$, $d = 4$ and $N = 32$
(a) Original X-ray image

(b) $R_{\text{min}} = 10$, $R_{\text{max}} = 30$, $d = 8$ and $N = 32$

(c) $R_{\text{min}} = 25$, $R_{\text{max}} = 60$, $d = 8$ and $N = 32$

Figure 2.8: Chest X-ray image from the database filtered by SBF with different values of $R_{\text{min}}$ and $R_{\text{max}}$. 
2.2 Using watershed transformation in filtered X-ray images

Watershed transformation is the most used method for image segmentation in the field of mathematical morphology. Specifically, it can be classified as a region-based segmentation approach. Let us consider a gray scale image as a topographic surface where the height in each point corresponds to its intensity. This method is an idea based on geography: it is that of a landscape or topographic relief which is flooded by water, watersheds being the divide lines of the domains of attraction of rain falling over the region. Another intuitive approach is to imagine the landscape being immersed in a lake, with holes pierced in local minimum. Basins will fill up with water starting at this local minimum, and, at points where water coming from different basins would meet, dams are built. The process stops when the water level has reached the highest peak in the landscape. The result of this segmentation is the image partitioned into basins separated by dams, called watershed lines.

First of all, watershed segmentation is going to be formally defined in the continuous case. The application of watershed segmentation in this work will be detailed next.

2.2.1 Watershed definition: continuous case.

The following definition of watershed for the continuous case is based on distance functions, specifically the topographical distance.

Let \( f \) be an element of the space \( C(D) \) of real twice continuously differentiable functions on a connected domain \( D \) with only isolated critical points. The topographical distance between points \( p \) and \( q \) in \( D \) is defined as

\[
T_f(p, q) = \inf_{\gamma} \int_{\gamma} \| \nabla f(\gamma(s)) \| \, ds
\]

(2.11)

where the infimum is over all paths \( \gamma \subset D \) with \( \gamma(0) = p \) and \( \gamma(1) = q \).

Let \( f \in C(D) \) have minimum \( \{m_k\}_{k \in I} \), for some index set \( I \). We define de catchment basin \( CB(m_i) \) of a minimum \( m_i \) as the set of points \( x \in D \) which are topographically closer
to $m_i$ than to any other regional minimum $m_j$

$$CB(m_i) = \{ x \in D : \forall j \in I \setminus \{i\}, f(m_i) + T_f(x,m_i) < f(m_j) + T_f(x,m_j) \}$$  \hspace{1cm} (2.12)

The watershed of $f$ is defined as the set of points which not belong to any catchment basin:

$$W(f) = D \cap \left( \bigcup_{i \in I} CB(m_i) \right)^c$$  \hspace{1cm} (2.13)

The watershed transform of $f$ assigns labels to the points of $D$, such that different catchment basins are uniquely labelled, and a special label $W$ is assigned to all points of the watershed of $f$.[4]

\subsection*{2.2.2 Meyer’s flooding algorithm: watershed transformation in discrete case.}

In this section, an algorithm to obtain the watershed transformation of a gray-scale image is explained. The first step is to label the regional minima with different colors and then repeat these three procedures until no such pixel exists:

- Select a pixel $p$, not colored, not watershed, adjacent to some colored pixels, and having the lowest possible gray level.
- If $p$ is adjacent to exactly one color then label $p$ with this color.
- If $p$ is adjacent to more than one color then label $p$ as watershed.

In this work, watershed transformation has been applied to all the X-ray chest images of the data set previously filtered with a Sliding Band Filter. The goal was to obtain a partition of each image by separating it in regions with similar intensity. Then, one can consider each region as a sample itself instead of working with each pixel by separate.

Fig. 2.9 shows an example of X-ray image filtered with SBF and then segmented with watershed transformation.
Figure 2.9: Chest X-ray image from the database filtered by SBF and segmented with watershed method.
CHAPTER 3
MACHINE LEARNING TECHNIQUES. RELEVANCE VECTOR MACHINE

At this point, we have each X-ray image segmented in regions with similar intensity. The aim of this work is to estimate how long will be a doctor observing each region. In general, we can suppose that if a doctor spends more time observing one region, means that the region has something that makes it special, that differs it of the usual pattern. In other words, given a segmented X-ray image, we want to determine which regions are more relevant to the observer.

We are going to approach this estimation by using machine learning techniques, particularly, a bayesian method called Relevance Vector Machine.

3.1 Machine Learning: Introduction

Machine learning is a branch of artificial intelligence based on the study of systems that can learn from data. It is the development of computer algorithms that learn empirically patterns and relationships between data and use these patterns for prediction. For example, if most of the regions of the X-ray image considered “relevants” have similar properties, it is probable that another region with similar characteristics is considered relevant by the observer.

The data set used to learn is called training data. In this work, the training data is formed by each region of the training set of segmented images. Each region is represented by an input vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top \in \mathbb{R}^n$ and a label (target) $t$. The components of $\mathbf{x}$ are features extracted from the image relatives to the pertinent region (in chapter 4 we will explain which features we have extracted from each region of the image), $n$ is the number of features (the dimension of the space of features), and $t$ the time spent by a doctor looking at this region. In general, there are two types of problem depending on $t$: regression and classification. In classification problems $t$ represents a class label while in regression $t$ is not
necessarily a category, it could be a real number, integer...

The goal is to find a function \( y(x) \) defined over the input space that well approximates known labels \( t \), so that given a feature vector \( x \) not belonging to the training set we could obtain its \( t \) value making an accurate prediction. In this chapter, we are going to explain how to find the parameters of this function \( y(x) \).

### 3.2 Preliminaries

Let be \( \{x_k\}_{k=1}^N \) a set of examples of input vectors and \( \{t_k\}_{k=1}^N \) its corresponding targets. We want to learn a model of the dependency of the targets on the inputs. The construction of \( y(x) \) is not unique, the simplest way is choosing \( y(x) \) to be a linear function:

\[
y(x; w) = \sum_{i=1}^{M} \omega_i x_i + \omega_0 = w^\top x + \omega_0 \quad (3.1)
\]

However, in order to achieve better performance we have chosen \( y(x) \) as:

\[
y(x; w) = \sum_{i=1}^{M} \omega_i \phi_i(x) = w^\top \Phi(x) + \omega_0 \quad (3.2)
\]

where the output is a weighted sum of basis functions \( \Phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_M(x))^\top \) and \( w = (\omega_1, \omega_2, \ldots, \omega_M)^\top \) the parameters to be determined. This basis function map the data from the original feature space to a higher dimensional feature-space. The advantage of this mapping is that we can transform the initial feature-space into a new space in which a linear regression is good enough to approximate our function.

In next sections we are going to present two methods **Support Vector Machine** and **Relevance Vector Machine** to approach our regression problem.

#### 3.2.1 Support Vector Machine (SVM)

First of all, we consider a supervised learning regression model called **Support Vector Machine** (SVM). In classification problems, the main idea of this method is to find parameters \( w \) such that the hyperplane defined by them in the mapped space is the one that represents the largest separation between the two classes. Similarly, in regression problems,
SVM penalizes errors that are greater than a threshold. We will see that this fact makes outliers less relevant and avoid overfitting problem.

The optimum values for \( w \) and \( \omega_0 \) are determined through a structural risk minimization procedure form:

\[
[w^*, \omega_0^*] = \arg \min_{w, \omega_0} \left( \frac{1}{2} w^\top w + C \sum_{n=1}^{N} L_\varepsilon(x_n) \right)
\]

(3.3)

where \( L_\varepsilon(\cdot) \) is called \( \varepsilon \)-insensitive loss function and is defined as:

\[
L_\varepsilon(x_n) = \begin{cases} 
|y_n - y(x_n; w)| - \varepsilon, & |y_n - y(x_n; w)| \geq \varepsilon \\
0, & \text{otherwise}
\end{cases}
\]

(3.4)

The principal characteristic of \( L_\varepsilon(\cdot) \) is that it does not penalize prediction error smaller than \( \varepsilon \). The constant \( C \) determines a trade-off between fidelity and model complexity. A high value of \( C \) gives accuracy to the model while a low \( C \) tries to fit the training data labels as much as possible [5].

The optimal solution to this problem is

\[
w^* = \sum_{i=1}^{l_s} \gamma_i \Phi(s_i)
\]

(3.5)

We can observe in eq. (3.5) that the optimal solution only depends on a few samples of the training set. These samples, \( \{s_n\}^{l_s}_{n=1} \) are called support vectors. The choice of vectors is not random, all of them have an issue in common: they are the critical elements of the training set, the ones that would be most difficult to classify in a classification problem. This result suggests that it does not matter how are the samples whose distance to the hyperplane is large and also proves that including new training samples that are out of this margin does not change the solution. Figure 3.1 shows an example of SVM classification problem.
Replacing the solution $\mathbf{w}^*$ in eq. (3.2) we obtain the next regression:

$$y(x; \mathbf{w}) = \sum_{i=1}^{l_s} \gamma_i \Phi(s_i)^\top \Phi(x) + \omega^*_0$$  \hspace{1cm} (3.6)

We can see in eq. (3.6) that there is no need to know explicitly $\Phi(x)$. It is enough to know the inner product $\Phi(x)^\top \Phi(x')$ that is usually described as a kernel function $k(x, x'; \sigma)$. The election of the kernel function depends specifically on the kind of problem we are working on (gaussian, polynomial, sigmoid functions...). In this work, we are going to use a Gaussian kernel:

$$k(x, x'; \sigma) = \Phi(x)^\top \Phi(x') = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{1}{2} \frac{||x - x'||^2}{\sigma^2}\right)$$  \hspace{1cm} (3.7)

where $\sigma$, Gaussian kernel’s width, is a free parameter of the SVM’s formulation [6].

At this point, we have the model clearly defined, in the next section we will explain how kernel width, $\sigma$, fidelity penalization, $C$, and insensitive tube, $\varepsilon$ are selected.
3.2.1.1 Choosing free parameters.

At this moment, we have to choose optimal values for the free parameters defined earlier. These optimal values are expected to be the ones that minimize the function defined in equation (3.3).

One method to approach it is called $K$-fold cross-validation. This method consists of repeating for every possible value of parameters $\sigma$, $C$ and $\varepsilon$ the next procedure: first of all, training data has to be split into $K$-folds. $K - 1$ are used to optimize the model, finding $w^*$ and $\omega^*_0$. Then, the remaining fold is used for testing. This is repeated and averaged over every fold. The metric used to measure error is defined in equation (3.8).

$$MSE = \frac{1}{\text{#1ftd} - \text{old}} \sum_{n \in \text{1ftd} - \text{old}} (y_n - f(x_n; w^*))^2 \quad (3.8)$$

Finally, after repeating this procedure for every possible value of the parameters, we choose the combination that minimizes eq. (3.8).

Once we have chosen values for the parameters, we can train our model using all the training data set and find the optimal values for $w^*$ and $\omega^*_0$.

As soon as we have the regression, we can extract features from any gazed region of the testing data set and evaluate it in the regression. This should show an estimation of how long the observer would spend looking at this concrete region of the X-ray image.

3.3 Relevance Vector Machine for Regression

The key feature of this bayesian method is that as well as offering good generalization performance, inferred predictors are exceedingly sparse in that they contain relatively few non-zero $\omega_i$ parameters. The majority of parameters are automatically set to zero during learning process, giving a procedure that is extremely effective at discerning those basis functions which are “relevant” for making good predictions.
3.3.1 Model formulation.

Given a training data set formed by feature vectors and targets \( \{ x_n, t_n \}_{n=1}^{N} \) (in next chapter we are going to explain which features have been extract from images in each gazed point), let \( y(x_n; w) \) be the non-linear (linear in the mapped space) regression that we want to find. As the regression will not fit all the samples of the training data set, we can assume that targets are samples from the model with a random additive mean-zero gaussian noise with variance \( \sigma^2 \).

\[
t_n = y(x_n; w) + \epsilon_n \tag{3.9}
\]

If we examine the previous equation we can observe that \( y(x_n; w) \) is deterministic, therefore the distribution of probability of \( t_n \) is also gaussian with variance \( \sigma^2 \).

\[
p(t_n|x) = \mathcal{N}(t_n|y(x_n), \sigma^2) \tag{3.10}
\]

We can consider that target values \( t_n \) are independent. Therefore, using the same notation as in SVM formulation, the likelihood of the complete data set corresponds to

\[
p(t|w, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp \left[-\frac{1}{2\sigma^2} ||t - \Phi w||^2 \right] \tag{3.11}
\]

where \( t = (t_1, \ldots, t_N)^\top \), \( w = (\omega_0, \ldots, \omega_N)^\top \) and \( \Phi \) is the \( N \times (N+1) \) design matrix with \( \Phi = [\phi(x_1), \ldots, \phi(x_N)]^\top \) where \( \phi(x_n) = [1, K(x_n, x_1), K(x_n, x_2), \ldots, K(x_n, x_N)]^\top \). In this work, the kernel used is defined by (3.7).

As we have as many parameters in the model as training samples, we need to impose some additional constraint on the parameters to avoid over-fitting. For example, we can add a penalty term to the likelihood or error function. In SVM this constraint was introduced by the \( \varepsilon \)-intensitive loss functions (eq. (3.3)). However, in RVM the restriction is based on a Bayesian perspective. An explicit prior probability distribution is defined over the parameters [?, tipping]
We define a zero-mean Gaussian prior distribution over \( w \):

\[
p(w|\alpha) = \prod_{i=0}^{N} \mathcal{N}(\omega_i|0, \alpha_i^{-1})
\]  

(3.12)

where \( \alpha \) is a vector of \( N+1 \) hyperparameters. Basically, there is an hyperparameter associated with every \( \omega_i \) which is responsible of moderating it.

At this point, we must define hyperpriors over \( \alpha \) and over the noise variance \( \sigma^2 \) as well. These quantities are examples of scale parameters and suitable priors are Gamma distributions:

\[
p(\alpha) = \prod_{i=0}^{N} \text{Gamma}(\alpha_i|a, b)
\]  

(3.13)

\[
p(\beta) = \text{Gamma}(\beta|c, d)
\]  

(3.14)

with \( \beta \equiv \sigma^{-2} \).

Reminder of Gamma function:

\[
\text{Gamma}(\alpha|a, b) = \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-b\alpha}
\]  

(3.15)

with \( \Gamma(a) = \int_0^\infty t^{a-1} e^{-t} \, dt \), the “gamma function”. To make the priors flat we might fix their parameters to a small values, for example \( 10^{-4} \). In this work, we use RVM with \( a = b = c = d = 0 \) for simplicity, thus we obtain uniform hyperpriors.

The assignment of an individual hyperparameter to each weight is the key feature of the RVM because it is responsible ultimately for its sparsity properties.

### 3.3.2 Bayesian Inference.

At this point, given a new test point, \( x_* \), predictions are made for the corresponding target \( t_* \) in terms of the predictive distribution:

\[
p(t_*|t) = \int p(t_*|w, \alpha, \sigma^2)p(w, \alpha, \sigma^2|t) \, dw \, d\alpha \, d\sigma^2
\]  

(3.16)

In order to solve this integral, first of all we are going to focus on the second term, the posterior distribution over all unknowns given the data. Note that we can decompose
it as:

\[ p(w, \alpha, \sigma^2 | t) = p(w | t, \alpha, \sigma^2) p(\alpha, \sigma^2 | t) \] (3.17)

Using Bayes rule, we can decompose the first factor of eq. (3.17) by doing:

\[ p(w | t, \alpha, \sigma^2) = \frac{p(t | w, \sigma^2) p(w | \alpha)}{p(t | \alpha, \sigma^2)} \] (3.18)

At this point, we can compute analytically the posterior distribution over the weights since its normalising integral, \( p(t | \alpha, \sigma^2) = \int p(t | w, \sigma^2) p(w | \alpha) \), is a convolution of Gaussians.

Therefore, the posterior distribution over the weights is given by:

\[ p(w | t, \alpha, \sigma^2) = (2\pi)^{-\frac{(N+1)}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2}(w - \mu)^\top \Sigma^{-1} (w - \mu) \right] \] (3.19)

where the posterior covariance and mean are:

\[ \Sigma = (\sigma^{-2} \Phi^\top \Phi + A)^{-1} \] (3.20)

\[ \mu = \sigma^{-2} \Sigma \Phi^\top t \] (3.21)

with \( A = \text{diag}(\alpha_0, \alpha_1, \ldots, \alpha_N) \).

To compute the second term in eq. (3.17) it is necessary to turn to an effective approximation. We are going to approximate the hyperparameter posterior, \( p(\alpha, \sigma^2 | t) \), by a delta function at its most probable values \( \alpha_{\text{MP}}, \sigma_{\text{MP}}^2 \). This approximation is based on the fact that this point-estimate is representative of the posterior in the sense that functions generated using the posterior most-probable values are near-identical to those obtain by sampling from the full posterior distribution. Besides, we do not need \( p(\alpha, \sigma^2 | t) \approx \delta(\alpha_{\text{MP}}, \sigma_{\text{MP}}^2) \), we only want

\[ \int p(t_* | \alpha, \sigma^2) \delta(\alpha_{\text{MP}}, \sigma_{\text{MP}}^2) d\alpha d\sigma^2 \approx \int p(t_* | \alpha, \sigma^2) p(\alpha, \sigma^2 | t) d\alpha d\sigma^2 \] (3.22)

to be a good approximation.
The search of the hyperparameters most-probable values, $\alpha_{MP}$, $\sigma_{MP}^2$ is thus the principal focus in relevance vector ‘learning’. In other words, the problem consists of maximising $p(\alpha, \sigma^2|t) \propto p(t|\alpha, \sigma^2)p(\alpha)p(\sigma^2)$ with respect to $\alpha$ and $\beta$.

In this work, we are going to consider the case of uniform hyperpriors so we only need to maximize the marginal likelihood, $p(t|\alpha, \sigma^2)$, which is computable and given by:

$$p(t|\alpha, \sigma^2) = \int p(t|w, \sigma^2)p(w|\alpha)dw$$

$$= (2\pi)^{-\frac{N}{2}}\sigma^2 I + \Phi A^{-1}\Phi^\top|^{-\frac{1}{2}} \exp \left[\frac{1}{2}t^\top(\sigma^2 I + \Phi A^{-1}\Phi^\top)^{-1}t\right]$$

(3.23)

In the next section, we show how to find the optimal values for $\alpha_{MP}$ and $\sigma_{MP}^2$.

### 3.3.3 Optimization of the hyperparameters.

The values of $\alpha$ and $\sigma^2$ that maximizes (3.23) cannot be determined explicitly so we are going to approach the optimization by an iterative method.

For $\alpha$, if we differentiate, equate to zero and rearrange (3.23) we obtain:

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{\mu_i^2}, \quad \forall i = 1, \ldots, N$$

(3.24)

with $\mu_i$ the $i$-th posterior mean weight from (3.21) and $\gamma_i$ are defined as follows:

$$\gamma_i = 1 - \alpha_i \Sigma_{ii}$$

(3.25)

where $\Sigma_{ii}$ is the $i$-th diagonal element of the posterior weight covariance from (3.20) computed with the current $\alpha$ and $\sigma^2$ values. Each $\gamma_i$ gives values ranging from 0 to 1 and can be interpreted as a measure of how “well-determined” its corresponding weight $\omega_i$ is by the data.

For $\sigma^2$, we can also differentiate, equate to zero and rearrange (3.23) and obtain:

$$(\sigma^2)^{\text{new}} = \frac{||t - \Phi \mu||^2}{N - \Sigma_{ii} \gamma_i}$$

(3.26)

The algorithm consists of choosing initial values for $\alpha$ and $\sigma^2$ and then repeat application of (3.24) and (3.26) until we reach a convergence criteria.
Once we have found $\alpha_{MP}$ and $\sigma^2_{MP}$, we just have to substitute them in (3.22) and then we have everything we need to compute (3.16).

### 3.3.4 Predictive distribution.

Estimating the hyperparameters, we make predictions based on the posterior distribution over the weights, conditioned on the optimal values $\alpha_{MP}$ and $\sigma^2_{MP}$. Therefore, we can compute the predictive distribution, from (3.16), for a new point $x_*$ using (3.19):

$$p(t_*|t, \alpha_{MP}, \sigma^2_{MP}) = \int p(t_*|w, \sigma^2_{MP})p(w|t, \alpha_{MP}, \sigma^2_{MP}) \, dw \quad (3.27)$$

Since both terms in the integral are Gaussians, we have:

$$p(t_*|t, \alpha_{MP}, \sigma^2_{MP}) = \mathcal{N}(t_*|y_*, \sigma_*^2), \quad (3.28)$$

with

$$y_* = \mu \phi(x_*) \quad (3.29)$$

$$\sigma_*^2 = \sigma^2_{MP} + \phi(x_*)^T \Sigma \phi(x_*). \quad (3.30)$$

The predictive mean is $y(x_*; \mu)$, or the basis functions weighted by the posterior mean weights, many of which will typically be zero. The predictive variance is formed by the sum of two variance components: the estimated noise on the data and that due to the uncertainty in the prediction of the weights.

So the solution of the weights, $w$, finding problem is the predictive mean $\mu$ and $\Sigma$ gives us information about errors bars.

### 3.4 Advantages of RVM over SVM

1. The SVM method is relatively sparse but makes an unnecessarily generous use of basis functions since the number of support vectors grows linearly with the number of training samples. This results in a hard and complex problem which often requires some form of post-processing.
2. In RVM’s formulation the number of free parameters to be obtained by cross-validation is lower than in SVM’s (only the choice of $\sigma$-kernel width against the choices of $\sigma$, $C$, and $\varepsilon$). This fact makes SVM highly time consuming in comparison to RVM method.

3. SVM’s predictions are not probabilistic. The result, either in classification and regression problems, is a binary decision or a point estimate. However, RVM’s formulation estimates the conditional distribution $p(t|x)$ which gives us information about how accurate is the obtained prediction.

4. Finally, this probabilistic view facilitates computing marginal likelihoods that is very useful in parameters comparison and selection, instead of using cross-validation.
CHAPTER 4
FEATURE EXTRACTION AND SELECTION

In last chapter, we explained a way to obtain the linear regression in a feature space. Now, we are going to focus on the extraction of features from the filtered and watershed-segmented images.

In chapter 2, we saw different algorithms to compute convergence index filters, specifically Iris Filter (IF) and Sliding Band Filter (SBF). Both of them evaluate the degree of convergence of the gradient vectors within its region of support towards a pixel of interest.

In order to have variety of features of a concrete region (delimited by watershed segmentation), we have obtained for each image from the data set:

- New image obtained by filtering the original one with an Iris Filter with parameters $R_{\text{min}} = 25$, $R_{\text{max}} = 60$, $N = 32$.
- New image obtained by filtering the original one with a Sliding Band Filter with parameters $R_{\text{min}} = 25$, $R_{\text{max}} = 60$, $N = 32$, $d = 8$.
- New image obtained by filtering the original one with an average filter of size $4 \times 4$ and filtering the result with a Sliding Band Filter with parameters $R_{\text{min}} = 10$, $R_{\text{max}} = 30$, $N = 32$, $d = 8$.

At this point we have, thus, for each original image, three more images from which we are going to extract features. We can see them in figures 4.1 and 4.2.
Figure 4.1: Original and IF filtered images for feature extraction
(a) SBF image with $R_{\text{min}} = 25$, $R_{\text{max}} = 60$, $N = 32$ and $d = 8$

(b) $4 \times 4$ averaged and SBF image with $R_{\text{min}} = 10$, $R_{\text{max}} = 30$, $N = 32$ and $d = 8$

Figure 4.2: SBF filtered images for feature extraction
4.1 Choosing features

We can classify the features into two groups: geometrical features and intensity related features.

In next sections we are going to explain, given a watershed region, figure (2.9), which features have been extracted and processed through a Principal Component Analysis (PCA).

4.1.1 Geometrical features.

This first group of features is related to the geometry of the region and, therefore, it does not change between the original and a filtered one.

Let \((m_0, n_0)\) be the first gazed point given one region. We have compute and used as feature:

- Horizontal and vertical position of \((m_0, n_0)\).
- Area and perimeter of the region. It might be some relations between the sizes of similar anomalies.
- Orientation: this refers to the angle (in degrees ranging from \(-90\) to \(90\) degrees) between the \(x\)-axis and the major axis of the ellipse that has the same second-moments as the region.
- Eccentricity: It specifies the eccentricity of the ellipse that has the same second-moments as the region. The eccentricity of an ellipse is the ratio of the distance between the focus of the ellipse and its major axis length. Its value is ranging between 0 and 1, meaning 0 that the ellipse is a circle and 1 that the ellipse is a segment.
- Horizontal distance between \((m_0, n_0)\) and the vertical axis located at the center of the image. This axis corresponds to the center of the vertebral column. This feature was motivated by the observation of this fact: there were more gazed-points in regions
that were away from the vertebral column. Therefore, the time the observer spent looking at closed to the vertebral column points used to be smaller.

- Number of gazed points within a region. Obviously, this fact gives more relevance to the region in question.

4.1.2 Intensity related features.

The features explained in this section are extracted from each of the four images that we have obtained filtering the original one.

4.1.2.1 Local means and variances.

Given a region and its first gazed point \( (m_0, n_0) \), we have considered a square \( N \times N \) square pixels neighborhood centered in \( (m_0, n_0) \) and we have compute its local mean.

The best way to compute \( N \)-local means, consists of convolving the full image with a \( N \times N \) matrix full of 1’s. This convolution gives a new matrix that has in each position its \( N \)-local mean.

As the anomalies can have different sizes, we have repeated this procedure for different values of \( N \): 15, 31, 51, 91 and 151.

Another selected feature related to mean computation is the difference of intensity between the image in question and an image obtained by averaging all the images. In particular, the difference between the intensity of \( (m_0, n_0) \) and the same point location in the “mean image”.

Finally, we have also considered the local mean and variance of all the points belonging to the region.

4.1.2.2 Derivative related features.

We also have include derivative features in the feature space, specifically, the gradient and laplacian of the first gazed point in a region, \( (m_0, n_0) \).
The gradients have been obtained applying a Prewitt operators to the images and the laplacians applying “laplacian of gaussian operators”.

We have compute horizontal and vertical gradients, and also its norm.

4.1.2.3 Features based on Mutual Information.

We are going to start defining the Mutual Information and explaining what is it for. Finally, it is explained the adopted criteria to obtain the mutual information between the pixel \((m_0, n_0)\) and its neighbors.

First, we need to define Entropy and Joint Entropy for two random variables \(x\) and \(y\):

\[
h(x) = -\int p(x) \ln p(x) \, dx
\]

(4.1)

and

\[
h(x,y) = \int \int p(x,y) \ln p(x,y) \, dx \, dy
\]

(4.2)

respectively.

The Mutual Information (MI) is defined in terms of the entropy and joint entropy as follows:

\[
I(x,y) = h(x) + h(y) - h(x,y)
\]

(4.3)

It is a quantity that measures the mutual dependence of the two random variables, \(I(x,y) = 0\) if and only if \(x\) and \(y\) are independent random variables. After knowing that, we can think that the mutual information between two neighborhoods of \((m_0, n_0)\) might be a good feature to be considered.

In particular, we have considered these two neighborhoods of \((m_0, n_0)\):

- \(U = 3 \times 3\) square neighborhood where \((m_0, n_0)\) is located on the center of the right column of \(U\).
- \(V = 3 \times 3\) square neighborhood where \((m_0, n_0)\) is located on the center of the left
column of $V$.

Finally, we have compute and incorporated to the features group $I(U, V)$.

Identically, we have compute the mutual information of two larger neighborhoods of $(m_0, n_0)$, especially 31 × 31 and also added the result as a variable of the feature space.

Table 4.1: Summary of the selected features

<table>
<thead>
<tr>
<th>Feature type</th>
<th>Num. features</th>
<th>Num. Images</th>
<th>Subtotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometrical</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Means and variances</td>
<td>8</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>Derivatives</td>
<td>6</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>Mutual Information</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>72</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2 Using Principal Components Analysis to reduce dimension of the feature space

As the number of features used in this work is appreciably large, it is probable that groups of variables move together. One reason to think this is that more than one variable might be measuring the same driving principle governing the behavior of the system. In many machine learning problems, there are only a few such driving forces but an abundance of resources permits you measuring lots of features. When this occurs, we can take advantage of this redundancy by simplifying the problem replacing a group of variables with a single new variable.

A good method to achieve this simplification is called Principal Component Analysis. It generates a new set of variables called Principal Components which are linear combinations of the original variables. All principal components are orthogonal to each other so there is no redundant information. They are, as a whole form, an orthogonal basis of the feature space.

There are multiple ways to construct an orthogonal basis of the space of data but
The principal components have some special characteristics:

- The first principal component is a single axis in space. When each observation is projected on that axis, the resulting values form a new variable, which its variance is the maximum among all possible choices of the first axis.

- The second principal component is another axis in space, perpendicular to the first with the same characteristic: when a new variable is projected, its variance is the maximum among all possible choices of the second axis.

- The full set of principal components is as large as the set of original variables, but the sum of the variances of the first few principal components usually exceeds 80% of the total variance of the original data.

4.2.1 Background and PCA algorithm.

Let \( x \in \mathbb{R}^n \) be a random vector with zero mean and covariance matrix \( \Sigma_x \). And let’s consider a linear transformation of the vector to a lower dimension random vector \( y \in \mathbb{R}^q \), \( q < n \):

\[
y = A_q^\top x
\]

with \( A_q^\top A_q = I_q \).

In PCA, \( A_q \) is a \( n \times q \) matrix whose columns are the \( q \) orthonormal eigenvectors corresponding to the first \( q \) largest eigenvalues of the covariance matrix \( \Sigma_x \). One important property to be considered for the choice of \( A_q \) is the maximization of the “spread” of the points in the lower dimensional space. This means that the points in the transformed space are kept as far apart as possible, retaining the variation in the original space. And also the minimization of the mean square error between the predicted to the original data.

Suppose we want to choose a subset of the original features of the random vector \( x \). This can be viewed as a linear transformation of \( x \) using a transformation matrix
\[
A_q = \begin{pmatrix}
I_q \\
0_{(n-q)\times q}
\end{pmatrix}
\]

or any matrix that is permutation of the rows of \(A_q\). Without loss of generality let’s consider the transformation matrix \(A_q\) and rewrite the corresponding covariance matrix as:

\[
\Sigma = \begin{pmatrix}
\{\Sigma_{11}\}_{q\times q} & \{\Sigma_{12}\}_{q\times(n-q)} \\
\{\Sigma_{21}\}_{(n-q)\times q} & \{\Sigma_{22}\}_{(n-q)\times(n-q)}
\end{pmatrix}
\]

It is not possible to satisfy all of the optimality properties of PCA using the same subset. Finding the subset that maximizes \(|\Sigma_y| = |\Sigma_{11}|\) is equivalent to maximize the “spread” of the points in the lower dimensional space, thus retaining the variation of the original data. Minimizing the mean square prediction error is equivalent to minimizing the trace of \(\Sigma_{22|1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\).

Therefore, to find an optimal subset of \(q\) features, one of the two quantities above is computed for all possible combinations of \(q\) features [8].
CHAPTER 5

RESULTS

In chapters before we have explained how to process and segment the lung radiographies in order to extract as many different features as it is possible with the objective of training a numerical observer. In the next section, we are going to evaluate the performance of this numerical observer and its accuracy in predicting how long a doctor would be looking at a region previously marked as “gazed”. First of all, there is an explanation of the experiment design followed by the obtained results and an estimation of the error. These results will determine whether the features are good enough or not and the reliability of the RVM algorithm.

5.1 Summary of the experiment design

The initial data set is formed by 68 X-Ray chest images previously aligned. For each radiography, we have a list of points where a human observer has gazed and the duration of this gazing. To evaluate the performance of our numerical observer, we have prepared a model trained by an RVM algorithm with 40 images and tested with 28. The next few lines are a brief summary of all the steps that we have followed in this experiment and explained in this report:

1. We have applied to each image:

   • an Iris Filter with parameters $R_{min} = 25$, $R_{max} = 60$ and $N = 32$. From now on, “IF images”.

   • a Sliding Band Filter with parameters $R_{min} = 25$, $R_{max} = 60$, $N = 32$ and $d = 8$. From now on, “SBF” images.

   • a $4 \times 4$ average filter followed by a Sliding Band Filter with parameters $R_{min} = 10$, $R_{max} = 30$, $N = 32$ and $d = 8$. From now on “ASBF images”.

   • a $4 \times 4$ average filter followed by a Sliding Band Filter with parameters $R_{min} = 10$, $R_{max} = 30$, $N = 32$ and $d = 8$. From now on “ASBF images”.
2. Perform the Watershed segmentation of SBF images.

3. Selection of the gazed points, \((x, y)\), that belong to the inside of the lungs. Discard the others.

4. Group all the gazed points that belong to the same watershed region and sum their gazed times. Each region is treated as a sample and the data set is now formed by gazed regions and their gazed times.

5. For each gazed region extract features from 4 different images, the original and the ones obtained in point 1.

6. Use the gazed regions of 40 different images and their targets to train a RVM and create a model.

7. Use the samples that form the other 28 images to test the performance of the model.

5.2 Results and error estimation

To evaluate the performance of our model, we have plotted some test samples target and its prediction. Each plot consists of a test sample image segmented in watershed regions. The intensity (in gray scale) of each region is directly related its target and its prediction, i.e. the duration of the gazing real and predicted in the region in question. Figures 5.1 and 5.2 show these result plots. The images have been rescaled to facilitate the comparison of relevant areas. The brighter regions correspond to samples whose duration is large. On the other hand, black or dark regions refer to non-gazed or few gazed regions.
Figure 5.1: Comparison of targets and predictions. Left: Target. Right: Prediction
Figure 5.2: Comparison of targets and predictions. Left: Target. Right: Prediction
To interpret the results correctly, it is necessary to have knowledge of how many watershed regions form an image and how many of them were gazed by our human observer. An image is constituted for about 150 watershed regions, of which approximately the 50% were gazed, i.e. around 75 (example in figure 2.9).

Figures 5.1 and 5.2 show that the algorithm is predicting considerably well which of the gazed regions are of most relevance. If we observe the images on the left column, we realize that few regions (from about 75) are visibly distinguishable. We can refer at them as “the relevant ones”. Now, we can see on the predictions column that most of them are also distinguishable and match with “the relevant ones”. However, the error increases if we compare the exact value of the prediction with the target.

As we know that the prediction obtained by RVM method follows a Gaussian distribution with mean the target and variance $\sigma^2 = \sigma^2_{MP} + \phi(x_*)^\top \Sigma \phi(x_*)$ (explained in chapter 3), we can carry out a more specific examination of the error. For each sample, we compute the error $e = |y_* - t|$ and we compare it to $1.96\sigma_*$ which corresponds to a confidence interval of 95%. Figure 5.3 shows the result. The horizontal axis corresponds to the samples while the vertical is the logarithm of the error.

The error between the desired target and the prediction is large, but also is $1.96\sigma_*$. It is not difficult to see that only few of the samples exceed $1.96\sigma_*$. This fact means that RVM method is working well even though it cannot make accurately predictions with these features.
Figure 5.3: Confidence interval
CHAPTER 6
CONCLUSIONS

As a general conclusion, we could say that the model successes in some occasions as a predictor of the gazed areas considered relevant by the observer. However, it fails in predicting the absolute duration of the gazing. Analysing all the steps of the procedure we can conclude that the failing might be related to the features choice, which is one of the most difficult problems in all machine learning field. In this work, the feature selection is especially based on searching nodules, opacities and sharp changes of intensity in X-Ray images. Nevertheless, it pretends to imitate human behavior, who will not only look for nodules but also might examine all the radiography and maybe keep more attention to those uncertain regions.

Despite not obtaining as good results as desired, we have point out several methods of image treatment and machine learning that have proved to be accurate and reliable.

In image processing field, it has been shown that convergence index filters success in identifying and making more noticeable the presence of nodules or others abnormalities. Besides, the size of abnormalities that we want to find can be selected by choosing appropriate values for the filters parameters. In this way, the searching could be filtered to find specified disease symptoms. Therefore, we think that COIN filters could be very useful to start designing automatic diagnosis models.

We can also point that RVM method is a good choice in application of machine learning techniques to problems aiming automatic medical diagnosis. This affirmation is based on the fact that RVM provides not only the predicted value but also its distribution. In medical field, a failure in the diagnosis could have very negative repercussions, thus it is of great importance to know the reliability of the predictions.
BIBLIOGRAPHY


[6] Jovan G. Brankov, Senior Member IEEE, Yongyi Yang, Senior Member, IEEE, Liyang Wei, Student Member, IEEE, Issam El Naqa, Member, IEEE, and Miles N. Wernick, Senior Member, IEEE. Learning a Channelized Observer for Image Quality Assessment.
