Foot Pathologies Classification From Pressure Distribution Over The Foot Plant [⋆]

José García-Hernández[1], Roberto Paredes[1], David Garrido[2] and Carlos Soler[2]

[1] Instituto Tecnológico de Informática
[2] Instituto de Biomecánica de Valencia
Universidad Politécnica de Valencia
Camino de Vera s/n, 46071 Valencia (Spain)

Abstract. Nowadays, approximately 35% of the population suffers problems in the feet. In many cases, some foot pathologies are treated by means of especial insoles. These insoles require customisation for the success of their treatment. However, there is a great lack of knowledge in this sector. As an initial step it is essential to develop a new objective tool capable to classify among the different foot pathologies. In this paper we present a system to classify different foot pathologies which uses the Nearest Neighbor (NN) classification rule based on weighted metrics and prototypes selection. The system uses as input data the pressure distribution over the foot plant.

1 Introduction

In the developed countries, approximately 35% of the population suffers problems in the feet enough serious to need medical attention. The illnesses of the feet are recognised as a great social problem in most of these countries, being most of the people affected women. A high percentage of the population of the European Union over 40 years old has some type of problem in their feet (above 90% in Spain, United Kingdom, Germany or Holland).

In many cases, some foot pathologies are treated by means of especial insoles whose principal aim is to protect the feet against external aggressions (overpressures, shear forces, etc.) and to maintain the articulations in a suitable anatomical position. These insoles require customisation for the success of their treatment. However, there is a great lack of knowledge in this sector. On the one hand, there is not information regarding the best orthetic solution depending on the pathology. On the other hand, in the health area the final adaptation of the insole to the patient highly depends on the specialist experience and on the capability of the patient to transmit the problems to the orthotics use. This lack of knowledge, together with a low technology in the design process, causes that the orthotics customisation process is performed at present without objective criteria and in a traditional way which implies a process with low efficiency.

[⋆] Work supported by the "Agencia Valenciana de Ciencia y Tecnología (AVCiT)" under grant GRUPOS03/031 and the Spanish Project DPI2004-08279-C02-02

Therefore, as an initial step and taking into account the information presented above, it is essential to develop a new objective tool capable to classify among the different foot pathologies, disorders or problems in order to optimise and focus the strategies of design to improve treatments and footwear functionality.

In this paper we present a system to classify different foot pathologies from pressure distribution over the foot plant during walking. For this purpose, we compare the k-Nearest Neighbor (kNN) classification rule using the well known *Euclidean Distance* with the 1-NN classification rule improved by the *Learning Prototypes and Distance (LPD)* method [1, 2]. While at section 2 we present the LPD method, at section 3 is shown the database used in the experiments presented at section 4.

2 Methodology

The Nearest neighbor (NN) classification rule has successfully been used in many pattern recognition applications. The good behaviour of this rule with unbounded number of prototypes is well known. However, in many practical classification problems only a small number of prototypes is usually available.

The NN performance can be improved by using appropriately trained distance measures of metrics. Improvements can be particularly significant for small prototype sets. Trained metrics can be *global* (the same for all the prototypes [3–5], *class-dependent* (shared by all the prototypes of the same class) [6–8] and/or *locally-dependent* (the distance measure depends on the particular position of the prototypes) in the feature space [9–12].

More concretely, in this work we have used (comparing with the well known *Euclidean Distance*) the approach called *Learning Prototypes and distances (LPD)*, proposed in [1, 2]. It stars with an initial selection of a small number of randomly selected prototypes from the training set. Then it iteratively adjusts both the position (features) of the prototypes themselves and the corresponding local-metric weights, so that the resulting combination of prototypes and metric minimises a suitable estimation of the probability of classification error. The adjustment rules are derived by solving the minimisation problem through gradient descent.

2.1 Approach

Let $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a *training set*; i.e., a collection of *training vectors* or *class-labelled* points $\mathbf{x}_i \in E, 1 \leq i \leq N$ in a suitable *representation space*. In our case, each training vector (prototype) $\mathbf{x}_i$ represents each subject. The components of this vector are the pressure values of the subject's plant foot acquired from a specific sensor. Being M the number of available plant foot sensors, the training vector $\mathbf{x}_i \in \mathbb{R}^M$, that is the representation space $E = \mathbb{R}^M$. Each prototype belongs to a particular class, among C different classes, being each class a different foot pathology.

The goal of the LPD algorithm is to use T to obtain a reduced set of *prototypes*, $P = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \subset E$, $n \ll N$, and a suitable *weighted distance* $d : E \times P \to \mathbb{R}$ associated to P, which optimise the Nearest Neighbor (NN) classification performance. Initially P is a subset of T randomly selected.

The *weighted distance* from an arbritrary vector $\mathbf{x} \in E$ to a prototype $\mathbf{y}_i \in P$ is defined as:

$$d_W(\mathbf{x}, \mathbf{y}_i) = \sqrt{\sum_{j=1}^{M} w_{ij}^2 (x_j - y_{ij})^2} \tag{1}$$

where w_{ij} is a weight associated with the feature j of the prototype $\mathbf{y}_i$. That is, in our problem, w_{ij} is the weight associated with the plant foot sensor j. These weights can be represented as an $N \times M$ weight matrix $W = \{w_{ij},\ 1 \le i \le N, 1 \le j \le M\}$. In what follows, whenever the weight matrix W can be understood from the context, $d_W(\mathbf{x}, \mathbf{y}_i)$ will be simply denoted as $d(\mathbf{x}, \mathbf{y}_i)$.

Note that this definition assigns separate weights to the different dimensions or *features* of the representation space. Note also that it is asymmetric and *local* in that it depends on the particular position of each $\mathbf{y}_i$.

Learning the Prototypes and their Weights In order to find both a matrix W and a suitable reduced set of prototypes P that results in a low error rate of the NN classifier, we propose minimise a criterion index which is an approximation to the NN classification error of T using P and $d(\cdot, \cdot)$. This NN error estimate can be written as:

$$J(P, W) = \frac{1}{n} \sum_{\mathbf{x} \in T} \textit{step}\left(\frac{d(\mathbf{x}, \mathbf{y}_{\mathbf{x}}^{=})}{d(\mathbf{x}, \mathbf{y}_{\mathbf{x}}^{\neq})} \right) \tag{2}$$

where the *step* function is defined as

$$step(z) = \begin{cases} 0 & \text{if } z \le 1 \\ 1 & \text{if } z > 1 \end{cases}$$

and $\mathbf{y}_{\mathbf{x}}^{=}$, $\mathbf{y}_{\mathbf{x}}^{\neq} \in P$ are, respectively, the *same-class* and *different-class* NNs of $\mathbf{x}$. Note that each term of the above sum (2) involves two different prototypes $\mathbf{y}_{\mathbf{x}}^{=}$, $\mathbf{y}_{\mathbf{x}}^{\neq}$, and their associated weights.

As in previous work [13, 14, 8, 15], a gradient descent procedure is used to minimise this index. This requires J to be differentiable with respect to all the parameters to be optimised; i.e.: y_{ij} and w_{ij}, $1 \le i \le k$, $1 \le j \le M$. Therefore some approximations are needed. First, the *step* function will be approximated by using a *sigmoid* function, defined as:

$$\mathcal{S}_\beta(z) = \frac{1}{1 + e^{\beta(1-z)}} \tag{3}$$

With this approximation, the proposed index becomes:

$$J(P, W) \ \approx\ \frac{1}{n} \sum_{\mathbf{x} \in T} \mathcal{S}_\beta(r(\mathbf{x})) \tag{4}$$

$$\text{where} \quad r(\mathbf{x}) = \frac{d(\mathbf{x}, \mathbf{y}_{\mathbf{x}}^{=})}{d(\mathbf{x}, \mathbf{y}_{\mathbf{x}}^{\neq})} \tag{5}$$

Clearly, if β is large $\mathcal{S}_\beta(z) \approx step(z)$, $\forall z \in \mathbb{R}$, and this approximation is very accurate. On the other hand, if β is not so large, the contribution of each NN classification error (or success) to the index J depends upon the corresponding quotient of the distances responsible for the error (or the success). As will be discussed later, in some cases this can be a desirable property which may make the *sigmoid* approximation preferable to the exact *step* function. The derivative of $\mathcal{S}_\beta(\cdot)$ will be needed throughout the paper:

$$\mathcal{S}'_\beta(z) = \frac{d\,\mathcal{S}_\beta(z)}{d\,z} = \frac{\beta e^{\beta(1-z)}}{(1+e^{\beta(1-z)})^2} \tag{6}$$

$\mathcal{S}'_\beta(z)$ is a "windowing" function which has its maximum for $z = 1$ and vanishes for $|z-1| >> 0$. Note that if β is large $\mathcal{S}'_\beta(z)$ approaches the Dirac delta function, while if it is small $\mathcal{S}'_\beta(z)$ is approximately constant for a wide range of values of z.

To obtain the partial derivatives from (4-5), required for gradient descent, it should be noted that J depends on P and W through the distances $d(\cdot,\cdot)$ in two different ways. First, it depends directly through the prototypes and weights involved in the definition of $d(\cdot,\cdot)$ (1). The second, more subtle dependence is due to the fact that, for some $\mathbf{x} \in T$, $\mathbf{y}^{=}_{\mathbf{x}}$ and $\mathbf{y}^{\neq}_{\mathbf{x}}$ may be different as prototype positions and their associated weights are varied.

While the derivatives due to the first, direct dependence can be developed from (4-5), the secondary dependence is non-linear and is thus more problematic. Therefore a simple approximation will be followed here by assuming that the secondary dependence is not significant compared to the direct one. In other words, we will assume that, for sufficiently small variations of the positions and weights, the prototype neighborhoods remain unchanged. Correspondingly, we can derive from (1) and (4-5):

$$\frac{\partial J}{\partial y_{ij}} \approx \frac{1}{n} \sum_{\substack{\forall \mathbf{x}\in T: \\ index(\mathbf{y}^{=}_{\mathbf{x}})=i}} \mathcal{S}'_\beta(r(\mathbf{x}))\, r(\mathbf{x})\, \frac{(y^{=}_{\mathbf{x}j} - x_j)}{d^2(\mathbf{x},\mathbf{y}^{=}_{\mathbf{x}})}\, w_{ij}^2 - \frac{1}{n} \sum_{\substack{\forall \mathbf{x}\in T: \\ index(\mathbf{y}^{\neq}_{\mathbf{x}})=i}} \mathcal{S}'_\beta(r(\mathbf{x}))\, r(\mathbf{x})\, \frac{(y^{\neq}_{\mathbf{x}j} - x_j)}{d^2(\mathbf{x},\mathbf{y}^{\neq}_{\mathbf{x}})}\, w_{ij}^2 \tag{7}$$

$$\frac{\partial J}{\partial w_{ij}} \approx \frac{1}{n} \sum_{\substack{\forall \mathbf{x}\in T: \\ index(\mathbf{y}^{=}_{\mathbf{x}})=i}} \mathcal{S}'_\beta(r(\mathbf{x}))\, r(\mathbf{x})\, \frac{(y^{=}_{\mathbf{x}j} - x_j)^2}{d^2(\mathbf{x},\mathbf{y}^{=}_{\mathbf{x}})}\, w_{ij} - \frac{1}{n} \sum_{\substack{\forall \mathbf{x}\in T: \\ index(\mathbf{y}^{\neq}_{\mathbf{x}})=i}} \mathcal{S}'_\beta(r(\mathbf{x}))\, r(\mathbf{x})\, \frac{(y^{\neq}_{\mathbf{x}j} - x_j)^2}{d^2(\mathbf{x},\mathbf{y}^{\neq}_{\mathbf{x}})}\, w_{ij} \tag{8}$$

where $r(\mathbf{x})$ and $\mathcal{S}'_\beta(\cdot)$ are as in (5) and (6), respectively. Using these derivatives leads to the corresponding gradient descent update equations. A simple manner to implement these equations is by visiting each prototype $\mathbf{x}$ in T and updating the positions and the weights associated with the *same-class* and *different-class* NNs of $\mathbf{x}$.

The effects of the update equations in the *LPD* algorithm are intuitively clear. For each training vector $\mathbf{x}$, its *same-class* NN, $\mathbf{y}_i = \mathbf{y}^{=}_{\mathbf{x}}$, is moved towards $\mathbf{x}$, while its *different-class* NN, $\mathbf{y}_k = \mathbf{y}^{\neq}_{\mathbf{x}}$, is moved away from $\mathbf{x}$. Similarly, the feature-dependent weights associated with $\mathbf{y}_{\mathbf{x}}^{=}$ are modified so as to make it appear closer to $\mathbf{x}$ in a feature-dependent manner, while those of $\mathbf{y}_{\mathbf{x}}^{\neq}$ are modified so that it will similarly appear farther from $\mathbf{x}$. Since these update steps are weighted by the distance ratio, $r(\mathbf{x})$, their importance depends upon the relative proximity of $\mathbf{x}$ to $\mathbf{y}_{\mathbf{x}}^{=}$ or $\mathbf{y}_{\mathbf{x}}^{\neq}$. This is further divided by the corresponding squared distance, thereby reducing the update importance for large distances. Finally, the resulting steps are *windowed* by the derivative of the sigmoid function applied to the distance ratio, $r(\mathbf{x})$. This way, only those prototypes (and their weights) which are sufficiently close to the decision boundaries are actually updated.

3 The Database Description

For the classification of the foot problems, pressure distribution over the foot plant during walking was used (figure 1). To acquire the plantar pressure distribution, the Biofoot®IBV pressure insoles were used (figure 1). The structure of the pressure files (plain text) exported by the application Biofoot®IBV is based on a matrix with as many rows as the sample frequency and the duration of the test define. The number of columns will depend on the number of sensor of the instrumented insole, in our case 64. So, for each subject (each training vector) n and for each foot, we have a matrix in the form:

$$\begin{pmatrix} x_{t_1 1} & x_{t_1 2} & \ldots & x_{t_1 64} \\ x_{t_2 1} & x_{t_2 2} & \ldots & x_{t_2 64} \\ \ldots & & & \\ x_{t_F 1} & x_{t_F 2} & \ldots & x_{t_F 64} \end{pmatrix}$$

being $x_{t_i j}$ the registered pressure by the sensor j in the time t_i. ¿From the matrixes we compute the average pressure for each sensor from each foot, that is:

$$x_s = \frac{\sum_{i=1}^{F} x_{t_i s}}{F}$$

As there is a matrix for each foot, we built each training vector as:

$$\mathbf{x}_{subject} = \begin{pmatrix} x_1 & x_2 & \ldots & x_{64} & x_{65} & \ldots & x_{128} \end{pmatrix}$$

Being x_i from right foot when $1 \leq i \leq 64$ and from the left one when $65 \leq i \leq 128$. Note that $E = \mathbb{R}^{128}$. Note also that $\mathbf{x}_{subject_A}$ is the training vector acquired from the subject A and the database can be represented as: $T = \{\mathbf{x}_{subject_1}, \ldots, \mathbf{x}_{subject_N}\}$.

As it is said above, the database is acquired by the Biofoot®IBV. It is a tool that has been developed by the *Institute of Biomechanics of Valencia* and it consists of a flexible insole with up to 64 piezoelectric ceramics (figure 1) distributed according to foot physiology in such a way that a greater density of sensors is placed under bony areas where the pressure uses to reach the highest values, mainly on the heel and the metatarsal heads.

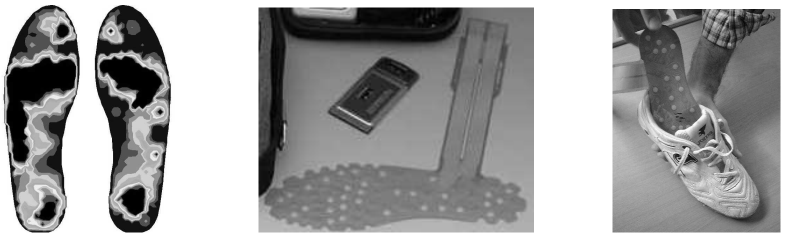

Fig. 1. Example of pressures (left) and Biofoot®IBV Insole (center and right).

Biofoot®IBV allows the study of the foot pressures during the normal conditions, walking and wearing shoes. Besides, by means of a telemetry system the signal is transmitted wireless to a PC, where it is saved (figure 2). On the other hand, and in order to avoid the effect in pressure values of the walking speed, two photocell barriers placed at a known distance were used. This procedure permits the researchers to remove the trials out of the interval [1,1.5] m/s, which defines the normal speed gait.

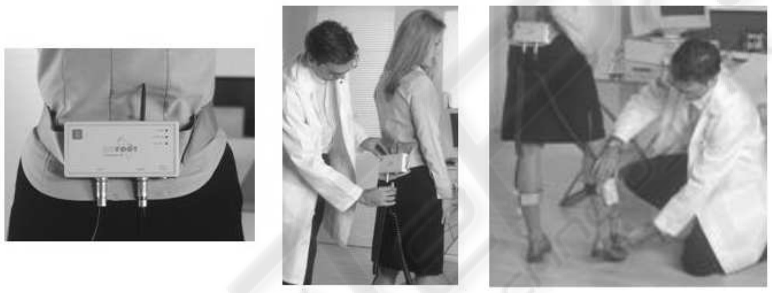

Fig. 2. Biofoot®IBV Use.

In order to obtain a representative sample of the foot problems from a statistically point of view, 60 subject of heterogeneous characteristics carried out the trials with their usual footwear. A specialist in foot diseases classified them in different groups of pathologies (a class label) after an exhaustive evaluation. Working in this way, for each subject we obtain a labelled training vector. The pathologies diagnosed were:

- Rheumatoid arthritis.
- Pain in the heel or in other foot parts different of plant.
- Plantar foot pain.
- Hallus Valgus (bunion).
- Hemiplegia.
- Neuropathic foot.
- Degenerative problems of the Central Nervous System.
- Diabetic foot.
- Consequences of fractures.

Due to the high incidence of *Pain in the heel or in other foot parts different of plant* and *Plantar foot pain*, and the fact that the 85% of the subjects were classified inside of these groups, the study was only focused on these pathologies. Besides, in the database the other pathologies has not a number of subject as great as representative for this work. So, the used database is composed by 49 subjects, that is, 49 training vectors, that are included in the two pathologies mentioned above (17 labelled subjects as "Pain in the heel or in other foot parts different of plant" and 32 as "Plantar foot pain").

4 Experiments

The experiments were carried out with the Biofoot®IBV database. Only two classes were taken into account, as is mentioned above. The classification error of the proposed method (LPD) is compared with the classification error of the well known k-NN rule based on the *Euclidean Distance*. This error is estimated using the well known *Leaving One Out (LOO)* estimation procedure due to the reduced number of available training vectors. In LOO, given N labelled training vectors, two prototypes sets are built. In the first one, called *training set*, we include $N-1$ vectors and the other vector is included (only it) in other set called *test set*. This process is repeated N times, each one including a different vector in the test set and the others $N-1$ in the training set. So, we do N different experiments; each one with different training and test sets. In each experiment, we train a classifier with the training set and test its performance with the test set. The idea is to work as we did not know the real label of the test vector (that is; the real pathology of the subject that corresponds with this vector) and see if the classifier is able to find the class. The final result is the average over the N tests.

Results are shown in table 1. As can be seen, LPD results with reduced training set and local weighted distances, are better than Euclidean Distance with the whole training set results. The best result with LPD is obtained reducing the training set to 3 prototypes per class (14.3 % error rate), this result is a improvement of 20 % over the best results obtained using k-NN with the Euclidean distance and all the training prototypes (18.4 %). The LPD with a reduced set of the vectors is able to improve the classification error in front of the original training set.

Table 1. k-NN error rate using Euclidean distance with different k values (left) and 1-NN LPD error rate for different prototypes per class (right).

k	Error Rate
1	**18.4 %**
2	18.4 %
3	28.6 %
4	24.5 %
5	40.8 %

Prototypes per class	Error Rate
1	16.3 %
2	16.3 %
3	**14.3 %**
4	16.3 %

These results are still worse than those given by an expert, which obtains about 5% error rate. This relative bad result of the automatic methods is due to the low number of samples available to learn adequately the class distributions. Also, the proposed preprocessing step takes the average of all the pressures along the time loosing some important temporal relations that are present in the plant during walking. We hope to improve the error rate in future works with more training samples and the use of other classification methods, as Hidden Markov Models (HMM), which allows us to treat the foot plant pressures as a *sequence* of pressure vectors.

5 Conclusions and Future Work

A system to classify different foot pathologies, from pressure distribution over the foot plant, has been presented. For this purpose, the Nearest Neighbor (NN) classification rule using the Euclidean Distance and the NN classification rule improved by the LPD method have been compared. The LPD obtained the best classification results with a (14.3 % error rate), an 20% of improvement over the NN rule using the original training set and the Euclidean distance. Future work will focus on obtaining more samples and using appropriate classification models (Hidden Markov Models) that allows us to process the pressure of the foot plant as a temporal sequence of pressures.

References

1. Paredes, R., Vidal, E.: Learning prototypes and distances (lpd). a prototype reduction technique based on nearest neighbor error minimization. In: In ICPR 2004. (2004) 442–445
2. Paredes, R., Vidal, E.: Learning prototypes and distances: a prototype reduction technique based on nearest neighbor error minimization. Pattern Recognition, Accepted (2005)
3. Kohavi, R., Langley, P., Y.Yung: The utility of feature weighting in nearest-neighbor algorithms. In: Proc. of the Ninth European Conference of Machine Learning, Prague. Springer-Verlag (1997)
4. Kononenko, I.: Estimating attributes: Analysis and extensions of relief. Technical report, University of Ljubjana, Faculty of Electrical Engineering & Computer science (1993)
5. Wilson, D., Martinez, T.R.: Value difference metrics for continously valued attributes. In: Proc. AAAI'96. (1996) 11–14
6. Howe, N., Cardie, C.: Examining locally varying weights for nearest neighbor algorithms. In: Second International Conference on Case-Based Reasoning. springer (1997) 445–466
7. Paredes, R., Vidal, E.: A nearest neighbor weighted measure in classification problems. In M.I. Torres, A.S., ed.: Proceedings of the VIII Symposium Nacional de Reconocimiento de Formas y Análisis de Imágenes. Volume 1., Bilbao (1999) 437–444
8. Paredes, R., Vidal, E.: A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. Pattern Recognition Letters. **21** (2000) 1027–1036
9. Hastie, T., Tibshirani, R.: Discriminant adaptative nearest neighbor classification and regressioon. Advances in Neural Informaton Processing Systems **8** (1996) 409–415 The MIT Press.
10. Ricci, F., Avesani, P.: Data compression and local metrics for nearest neighbor classification. IEEE Transactions on PAMI **21** (1999) 380–384
11. Short, R., Fukunaga, K.: A new nearest neighbor distance measure. In: Proc. 5th IEEE Conf. Patter Recognition. (1980) 81–86
12. Peng, J., Heisterkamp, D.R., Dai, H.: Adaptative quasiconformal kernel nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004)
13. Paredes, R.: Técnicas para la mejora de la clasificacin por el vecino más cercano. PhD thesis, DSIC-UPV (2003)
14. Paredes, R., Vidal, E.: Weighting prototypes. a new editing approach. In: Proceedings 15th. International Conference on Pattern Recognition. Volume 2., Barcelona (2000) 25–28
15. Paredes, R., Vidal, E., Keysers, D.: An evaluation of the wpe algorithm using tangent distance. In: In ICPR 2002. (2002)