I+D Revista de Investigaciones
ISSN 22561676
Volumen X Número X Año XX mes-mes 2014 p.p Artículo de Investigación

Challenges of multivariable and multiclass classification problems

Edwin Alberto Silva Cruz[1], Carlos Humberto Esparza Franco[2]

[1]Universidad, seccional ciudad, País.

Artículo recibido en mes XX de año; artículo aceptado en mes XX de año

Citación del artículo: Primer apellido,J. & Primer apellido,M. (año). Efectos de la ejercitación gestual mediante sensores faciales en la expresión pro social de la ira. *I+D Revista de Investigaciones*, 1(2), pp-pp.

Abstract

Multivariate and multiclass classification problems are highly relevant in recent machine learning developments, because there are numerous real life applications that rely on high number of parameters and classes. For example, genetics, artificial vision, physics, language synthesis, text and speech analysis and other applications. Due to this, there has been exhaustive work on the development of machine learning methodologies able to deal with this kind of problem, such as boosting of weak classifiers, neural networks and ensemble of classifiers. However, there are a number of unsolved difficulties, including

[1] Título de pregrado, Universidad. Título más alto posgrado, Universidad XXXX. Docente- investigador del grupo GPS. Universitaria de Investigación y Desarrollo (Colombia): Dirección XXXXX, PBX:XXXX . Correo electrónico institucional: xxx

[2] Título de pregrado, Universidad. Título más alto posgrado, Universidad XXXX. Docente- investigador del grupo: XXX. Universidad XXX de la ciudad de XXX (Colombia): Dirección XXXXX, PBX:XXXX . Correo electrónico institucional: xxx.

issues with high dimensionality, overfitting and optimization problems. In this paper some of the challenges for multiclass high dimensionality classification problems are shown, as well as conventional methodologies used in these problems. A case study with facial expression recognition is presented, in which some of the inconveniences of traditional classification methodologies are used, mainly due to high dimensionality.

Introduction

A multiclass and multivariate classification problem refers to the discrimination between more than two different classes using a high number of features. Multivariate classification is well known in machine learning, especially because many real life applications require the use of a high number of features. This necessity boosted the development of machine learning, including the creation of novel methodologies with high potential. During the 70s and 80s, this development produced high expectations for machine learning systems to solve increasingly complex problems, including sophisticated AI (Artificial Intelligence). Neural networks were created and developed based on the perceptron, which is a mathematical model simplification of a natural neuron. ANN (Artificial neural networks) could be trained to learn an objective function even using simplified activation functions for each neuron. Due to this and the improvement of computation systems with better performance and memory, the relatively common belief that ANN would become the AI solution and sooner or later they would produce learning and adaptation capabilities on par with the human brain was not surprising.

With these perspectives, the use of a high number of features became common. As long as the ANN could be trained, it was believed that a high number of features, even if some of them could be redundant or simple noise, would improve the classification performance. In addition, perceptrons were highly regarded for the solution of countless problems, including the affirmation by Rosenblatt, the inventor of perceptrons: “(perceptrons) may be able to learn, make decisions and translate languages” (Dormehl, 2016).

On the other hand, there were problems on the horizon. The main one is there are simple functions that cannot be trained using a single perceptron. The simplest example is the XOR function. A continuous classification boundary cannot be traced to separate the two classes in a XOR function. Naturally, the easiest approach is to use multiple layers of

perceptrons, so that non-linearities and more complex classification boundaries can be built. However, the learning stage could not be achieved using the learning methodologies available at the time, so new methodologies were developed, namely backpropagation. With backpropagation the blame of the classification error is shared between the different layers (as opposed to the previous methodology in which error was blamed exclusively to the last layer), so training would ideally converge towards the target function.

This kept machine learning on track, but the high optimism proved to be an issue in the 90s, when expert systems became troublesome due to costs, learning problems and mathematical optimization issues.

The underlying process is commonly called "AI Winter". During the AI winter the interest and funding in machine learning have been dramatically reduced. An AI winter is commonly followed by a recovery period, generally because the benefits of machine learning outweigh the difficulties or because there has been a scientific breakthrough on the field that allows further development beyond a previous dead end.

Recently, expectations are increasing, due to important developments on the field. The weakness of methodologies such as neural networks and SVM when dealing with high-dimensionality low-sample size data (HDLSS) have been addressed by using clever techniques, such as reducing the data dimensionality with feature selection so that the SVMs and NNs will not have to handle complex and scarce data in high dimensionality. Another approach is to research novel machine learning methodologies more suited to deal with complicated data. One of them is the arrival of Deep Learning, whose potential has been already suggested with classification systems with hierarchical capabilities, categorization and labeling superior to the previous state of the art methodologies.

Multiclass classification system

A classification system includes a supervised learning methodology. The objective is to learn a function using a labeled training set, consisting of a number of features and a class label in its simplest form. The classification problem is the optimization of a target function so that the input $\mathbf{X}_i$ is transformed to an output Y_i, where $X_i \in \mathcal{X}$, $Y_i \in \mathcal{Y}$ and $\mathcal{Y} =$

$\{1,2,\dots,K\}$. When $K > 2$, where K is the number of classes, the problem is multiclass[3] (Rifkin, 2008). In order to achieve the classification, a classification function is required so that g: $\mathcal{X} \to \mathcal{Y}$.

Multiclass problems require particular methodologies because many classification techniques are binaries. The simplest one is called one vs. all (Rifkin, 2004). In this strategy, a classifier is trained for each class, so there are K classifiers f_k (Bishop, 2006). Once the set of classifiers f_k is trained, the classification decision is normally determined by the highest confidence score given by equation (1) :

$$y = \arg\max_{k \in 1\dots K} f_k \; x \tag{1}$$

This methodology is simple, but it has several problems, namely the confidence per classifier can have very different values and the training set is unbalanced considering generally the negative training subset is larger than the positive training subset, which could produce classifier whose output is typically negative.

One variation to the one vs. one classification architecture is to create multiple classifiers to separate between different classes per sample. This methodology is called one vs. one and the procedure is to train multiple $f_{k,l}$ one vs. one classifiers where k is the first class and l is the second class and the task of the classifier is to differentiate between these two classes. In total there are $\sum_{k=0}^{K-1} k$ or $K(k-1)/2$ classifiers. The classification is performed with a voting scheme, so that the class that gets the highest number of votes is selected, as shown in equation (2).

$$y = \arg\max_{k \in 1\dots K} \left(\sum_{l} f_{k,l} \; x \quad \right) \tag{2}$$

Whereas this methodology is better than one vs. all, it has some problems such as the high number of classifiers required, especially if there is a high number of classes, the possible tie between different classes and the likelihood of a rogue class getting a high number of votes vs. recessive classes. Finally, one more recent technique is based on the use of output codes. Each class is assigned an output code, generally a binary code whose length is

[3] There are also multilabel problems, for instance a category classification of texts in which a text can belong to different categories, but these problems are not included in the scope of this paper.

defined by the number of classes K. Theoretically, the minimum code length is $\log_2 K\,/K$ in order to univocally represent each class, but in reality this value is not a good choice because of the small Euclidian (or most other metrics) distance between the codes.

These techniques are pretty straightforward, so much that several authors invented them independently. In real life problems, the choice between them is mostly a matter of computational power, because the performance is similar. More sophisticated ideas have been developed, such as the single machine approach created in (Weston & Watkins, 1999). However, this methodology does not perform better than one vs. all or one vs. one but in very specific cases, where the data are not easy to separate but there are data subsets where the penalization for the correct class is lower than for the wrong classes (i.e. less negative values).

Multivariate classification

There are multiple classification problems that use a high number of features/classifiers. These class of problems are typically called multivariate, high dimensional or multivariable classification (Fan & Fan, 2008). Most conventional classification methodologies are not fitted to deal with a high number of parameters whose class separation capabilities are low (also known as weak features). For example, Artificial Neural Networks (ANN) are not usually a viable solution for problems with high number of weak features due to instability problems (Cunningham, Carney, & Jacob, 2000) . Conventional neural network training usually produce considerably different models if the training data is slightly different (Breiman, 1996). When the number of features is high, this issue is exacerbated, so it is difficult to achieve proper convergence in the training stage.

One useful technique to deal with multivariate problems is by treating each feature as a single metaclassifier and then combining the individual outputs, for example using a winner takes all scheme (Biehl & Ghosh, 2005). This is reasonable because of several reasons: i. Instability problems are avoided, ii. It is easier to train one classifier per feature than a complex classifier with multiple inputs (one per feature) and iii. High dimensionality issues are prevented. However, in more recent times novel techniques using weak features have been developed, for example AdaBoost (Viola & Jones, 2001). In AdaBoost the weak

features are combined using a weighted sum and the weights are tweaked in order to favor the misclassified instances. Each weak classifier does not need to have good classification power; as long as it is higher than $\frac{1}{N}$ where N is the number of classes and the features are not dependent, multiple weak features can be combined in order to produce strong classification. Neural networks and SVM, instead, suffer from high dimensionality due to the Hughes Effect (Hughes, 1968). The Hughes Effect is the phenomenon by which the increase of the dimensionality of a problem requires a higher number of samples in order to properly describe the manifolds of the classes, and whereas in general the number of samples is limited, this means high dimensionality problems are usually poorly characterized.

Fisher Linear Discriminant: Fisher Linear Discriminant (FLD) is not suited well for high dimensionality data because the data may be aligned in high dimension so that the projection into a lower dimensionality space produces perfect separability between the classes but the projection is performed in a direction whose angle with the Bayes direction is high. This can be also viewed as the rotation of the high-dimensionality space so that the data from each class appears to be aligned. In that scenario, data may be projected into the rotation direction with perfect training data subset separability but low generalization capabilities. In general, FLD requires the estimation of the covariance matrix and that estimation can be very inaccurate if the available data size is too small compared with the requirements for high dimension.

Support Vector Machines: The main advantages of SVM in machine learning are: 1) Its convergence towards the Bayes rule when the sample size increases (Lin, 1999), which means given a high enough number of samples the classification rule produced by the SVM converges towards the optimal Bayes rule. 2) The implementation is easy. 3) SVM is robust to model specification (Allen, 1997), so feature selection is less complicated. There are, however, several issues related with SVM for high dimension data, especially in the common case there are, high-dimension, low-sample size data (HDLSS data). SVM machine works by creating separation hyperplanes using support vectors, which usually are the data closest to the separation boundary, so the decision function is given by $f(\mathbf{x}) = (\mathbf{w}, \mathbf{x}) + b$, where w is the direction of the separation hyperplane and b is the intersection.

In high dimensionality, these support vectors are usually a considerable fraction of the whole data. This means a high proportion of the data lies between two parallel hyperplanes to the separation boundary, and this phenomenon is called data-piling (Quiao, Zhow, & Huang, 2008) (Marron, 2015). Data-piling means the separation hyperplane direction may typically have a high angle with the Bayes rule direction, despite it should ideally have the same angle. Related with this, new samples can be severely misclassified if they belong to regions far from the training data, due to the real optimal separation being far away from the SVM hyperplane caused by the high angle between the two hyperplanes. Finally, not only the SVM direction *w* may be very different to the Bayes rule direction, but it can also variate in high degree depending on the training data, which means different training subsets can produce very different separation hyperplanes.

Distance Weighted Discrimination: Distance Weighted Discrimination (DWD) is a more recent approach whose objective is to prevent the data-piling phenomenon found in SVM with HDLSS data (Marron, 2015). Like in SVM, DWD is the solution of an optimization problem. However, in this case a different set of distance functions between the data and the boundary hyperplane is used. The simpler way is to optimize the sum of the inverse distances, so points that lie closer to the boundary are given higher significance.

DWD has shown to produce similar or better results to SVM in high dimensionality problems with data scarcity, and actually better results than SVM in specific cases, such as when there are outliners in the dataset that push the support vectors for SVM.

Case studies

One example of multiclass high dimensionality classification is facial expression recognition (FER). In FER, as in many applications that require the analysis of images or video, the available information is usually large. For example, a 256×256 monochromatic 8 bits per pixel image has a total of 65.536 bytes. A normal classification system would have troubles with dealing with each pixel info as an input, and the total number of possible images in this subspace is 1×10^{1233}, which means, from an information point of view, the

dimensionality of the image data is excessive for every practical multiclass classification problem. In addition, there are several restrictions with regards to the number of features used as a function of the size of the available data. In (Jain & Waller, 1978) it was shown that the optimal number of features depends on a formula stated in equation (3)

$$p_{opt} \approx N - \frac{\xi^{-2N} + 2\ N - 2\ ln\xi - 1}{4\,ln\,\xi[\ N - 2\ \ ln\,\xi - 1]} \tag{3}$$

where ξ is a higher than zero constant and N is the number of samples per class. Whereas this equation was obtained for a 2-classes problem, it shows the optimal number of features approximately linearly increases with the number of samples given a database of enough size. This already produces a limitation on the number of features required for the classification problem, and this is already using several ideal assumptions, such as Gaussian distribution of the data, similar number of samples per class, only two classes and similar covariance matrix Σ for the two classes. There are other considerations, though. The optimal number of features is also heavily dependent on the classification methodology and the actual statistical distribution of the data, as shown in (Hua, Xiong, Lowey, Suh, & Dougherty, 2005), so the real optimal number of features variates from problem to problem and with different methodologies.

For the FER problem we used the CK+ database (Lucey & Kanade, 2010) and LBP codification (Ojala, Pietikäinen, & Harwood, 1994). The CK+ database contains video sequences with validated facial expressions according with the Ekman classification of human facial expression (Ekman, 1993). This kind of classification problem is very appropriate for the subject, because it includes 7 classes (6 facial expressions plus 1 neutral stance) and a high number of LBP features. For the LBP codification used in these tests, a facial region of size 256×256 was extracted and it was divided in 8×8 local regions, as proposed in (Ahonen, Abdenour, & Pietikäinen, 2006). The codes within each region are collected using a histogram, so for each region there is a histogram whose length depends on the possible number of codes (2^N where N is the number of neighbors in the LBP calculation) and the total LBP code is the concatenation of all the histograms.

The LBP code is defined in equation (4).

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{4}$$

where g_p are the neighbor pixels from the evaluated pixel g_c, P is the number of neighbors and R is the radius around the central pixel. The function $s(\cdot)$ is defined in equation (5)

$$s\ x\ = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Using $N = 8$ and no mapping, the length of each histogram is 256 and there are a total of 64 histograms in a LBP code. This means the dimensionality of the problem is high, given the elevated number of features used to describe the facial expressions. The protocol to test the problem with data in high dimensionality is as follows:

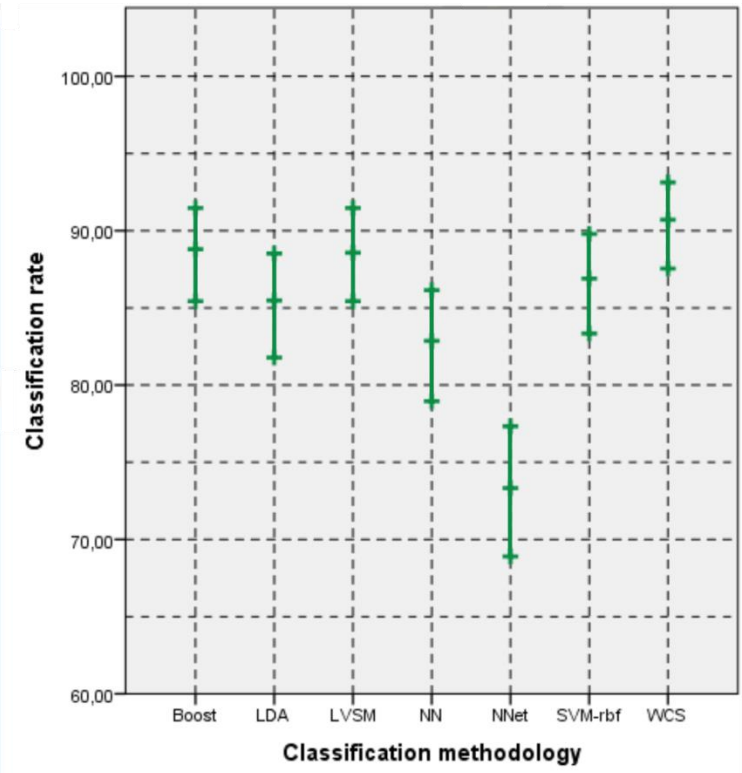
i. The data is divided into training and validation subsets using 10 folds. In order to prevent methodologic errors of negative training, the division was not performed sample-wise but person-wise (i.e. there are not two samples from the same individual in both training and validation subsets, regardless if they are from different classes).
ii. The training subsets were used to train different classification systems as follows: simple nearest neighbors, linear discriminant analysis, linear SVM, SVM-rbf and neural networks. The classification methodology for the binary classifiers (SVM and linear discriminant analysis) is the fusion of 1 vs. 1 classifiers and in the case of neural networks the output layer has 7 neurons, one per class.
iii. The trained classifiers were used to validate their corresponding validation subsets.

For SVM-rbf and neural networks the parameters were not considerably modified during the training protocol, so to prevent fine tweaking in order to produce artificially elevated results. However, some degree of tuning was performed to guarantee the classification hyperboundaries obtained with either of those methods were adequately smooth instead of

very specialized boundaries which are heavily prone to overfitting. In order to achieve that, the SVM parameters for each classifier are the same, and they were obtained using a search grid with variable values for ε and C.

LBP-codes are not very efficient from an information point of view. First, because there are a high number of redundant codes due to the nature of the codification. Even if mapping is performed, so to prevent redundancy from rotationally equal codes or symmetrical codes, there is still a second issue from information point of view. The histogram of LBP codes is not homogeneous. Using real images we found out there is a very high representation of some LBP codes in the histogram, whereas there are other codes with very scarce representation. This means the entropy level is quite suboptimal for codification. As a consequence of this, LBP codes produce a very high dimensionality space to represent a problem with a very limited number of samples; hence the correct approach is not straightforward.

Figure 1. FER classification rates using different multiclass and multivariate methodologies



In

Figure 1 the results of FER classification rates are shown, using AdaBoost (Boost), Linear Discriminant Analysis (LDA), linear SVM (LSVM), Nearest Neighbours (NN), Neural networks (NNet), radial basis functions SVM (SVM-rbf) and Weighted Chi-squared distances (WCS). These results have some interesting implications. Whereas SVM-rbf are widely recognized as one of the most effective techniques in multiclass classification, the obtained result was not superior than using linear SVM or weighted chi-squared distances, and just slightly better than discriminant analysis. This was after already fine tweaking the SVM parameters so to obtain the set of parameters that produced the best possible performance with the validation subsets. As a matter of fact, very small variations to the SVM parameters produced useless SVMs, either whose output was always the same or SVMs whose output always matched the class from the training set with the highest number of samples, which is a typical phenomenon found in SVM training. It is important to notice the confidence intervals mean it cannot be explicitly stated WCS produced better results than SVM-rbf, but at the very least they are not worse, which is remarkable given the considerable simplicity of chi-squared methodology.

The performance of neural networks was the worst by far. This happens because it was not possible to find a set of parameters that provided adequate generalization rates. If the architecture of the neural network was complex, the result was a very overfitted neural network whose output was very close to the target with the training data, but without generalization capabilities. If the architecture was too simple, it was unable to properly capture the nonlinear nuances that describe the facial expressions with LBP-based codification, so the classification rates were low.

A second example was generated with artificial high dimensionality data. The data was generated at dimensionality 80 and 5 classes, and the classes were produced by clustering, with 10 parameters consistent of Gaussian noise, 35 parameters produced with Gaussian dispersion and 35 parameters with non-normal distributions (log-normal, Weibull and exponential). The reason to use non-normal distributions is that the optimal separation

hyperplane for normal Gaussian distribution clusters is a normal equidistant plane from the centroid of each cluster, so classification methodologies such as nearest neighbor are the best regardless their simplicity. For each test, 20.000 samples were simulated, with asymmetric representation for the classes. 2.000 samples belong to class I, 4.000 samples belong to classes II to IV and 6.000 samples belong to class V. The reason to include this asymmetry was to challenge the optimization of target functions for some classification methodologies that have problems with non-homogeneous distribution of data per class.

Table 1. Result of a 5-classes problem using different classification methodologies

Boosted	**Discriminant analysis**	**Linear SVM**	**Nearest neighbor**	**Neural network**	**SVM-rbf**	**Weighted chi-squared**
87.12	86.63	90.73	88.95	65.18	84.32	87.36

These results show a similar trend. Neural networks were unable to adequately characterize the 5-classes problem due to the high number of features, so the classification rates were low. It is possible an extremely finely tuned neural network with specific number of layers and neurons per layers is able to produce both adequate generalization without incurring in underfitting, but that requires user intervention and a slight change of the conditions, for example the removal or addition of parameters, means the architecture will be probably useless. This lack of stability is crucial on processes such as feature selection, because most feature selection algorithms require the addition or exclusion of parameters or set of parameters each iteration, with training and validation protocols per iteration (Pudil, Ferri, Novovicova, & Kittler, 1994). If the machine learning method is too sensible to the number of parameters, the feature selection algorithm will have issues. The results obtained with the other techniques were very similar, but linear SVM had an edge with 90.73 ±0.41% classification rate.

These results are not surprising given the high dimensionality of the data. Good classification rates were obtained because of the independency between the parameters, so each parameter works as a weak classifier, hence the good capabilities of methods that rely

on the individual contribution of each parameter (weighted chi-squared and boosted) compared with methods that use the full ensemble of parameters (LDA, SVM or neural networks).

Conclusions

This work showed different machine learning techniques commonly used in the problem of multi-class classification problems. Whereas these techniques are conventionally applied in problems with a moderate number of parameters, things become more complicated with a high number of features. High dimensionality spaces are complicated due to a number of issues. First, the volume of a hyperspace grows exponentially with the number of parameters, while the distances grow at a lower rate. Given the classification accuracy depends on the separability between classes, and such separability relies on the distance between data from different classes, adding parameters can actually complicate the problem unless either the included parameters actually provide important classification capabilities or if the classification method is based on the ensemble of 1-parameter classifiers (i.e. Bayesian trees or boosted).

Three different case studies were produced. The first one is based on facial expression recognition data from the CK+ database. Data from images and video are very important because in many cases there are a high number of extracted parameters, such as in the codification used for this study (LBP-based). The second and third studies were performed using simulated data of high dimensionality and Gaussian, log-normal, Weibull and exponential distributions. The additional distributions besides Gaussian were used because exclusively Gaussian distributions are a trivial classification problem in which the simplest classification methodology using nearest neighbor produces at the same time the optimal classification boundary. The difference between the second and the third studies is the separability between the classes. For the first study the classes were separated in average a distance equivalent to one standard deviation of intraclass data. For the second study that distance was halved to one half standard deviation of intraclass data, so the classification problem is considerably harder.

The results from the three studies show a similar trend: the scarcity of data is an important issue for multi-class classification problems. This problem was exacerbated when the classification difficulty was increased, namely when the interclass distances were reduced, although the same challenge increase would have produced by either increasing the number of parameters or by reducing the number of samples as well. As a matter of fact, neural networks and non-linear SVMs had increasing troubles with escalating data scarcity, while at the same time simpler classification methodologies, especially nearest neighbors, weighted chi-squared and linear SVMs, consistently produced better results with increasing data scarcity.

The produced observations are a suggestion that high dimensionality multi-class classification problems are still an open field, especially because conventional classification methodologies used in more traditional low dimensionality spaces have important issues with high dimension hyperspaces. Simple methodologies produced better results, but it is reasonable to expect novel approaches could be better suited to properly characterize complex structures in high dimension and, as a result, produce closer to optimal separation hyperboundaries that increase the classification rates in this kind of problems.

This research will continue with the inclusion of more modern techniques, particularly focused on deep learning, which has proven to be an effective methodology in diverse kind of high dimension problems, such as video, image, speech and text classification problems.

References

Ahonen, T., Abdenour, H., & Pietikäinen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Learning*, *28*(12), 2037–2041.

Allen, M. (1997). Model specification in regression analsis. In *Understanding regression analysis* (pp. 166–170). Springer.

Biehl, M., & Ghosh, A. (2005). The dynamics of learning vector quantization. *European Symposium on Artificial*, (April), 27–29. Retrieved from

http://www.dice.ucl.ac.be/Proceedings/esann/esannpdf/es2005-82.pdf

Bishop, C. M. (2006). Pattern recognition. *Machine Learning*, *128*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Cunningham, P., Carney, J., & Jacob, S. (2000). Stability problems with artificial neural networks and the ensemble solution. *Artificial Intelligence in Medicine*, *20*(3), 217–225. http://doi.org/10.1016/S0933-3657(00)00065-8

Dormehl, L. (2016). *Thinking Machines: The inside story of Artificial Intelligence and our race to build the future*.

Ekman, P. (1993). Facial expression and emotion. *American Psychologist*, *48*(4), 384.

Fan, J., & Fan, Y. (2008). High dimensional classification using features annealed independence rules. *Annals of Statistics*, *36*(6).

Hua, J., Xiong, Z., Lowey, J., Suh, E., & Dougherty, E. R. (2005). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, *21*(8), 1509–1515. http://doi.org/10.1093/bioinformatics/bti171

Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Trans. on Information Theory*, *14*(1), 55–63.

Jain, A. K., & Waller, W. G. (1978). On the optimal number of features in the classification of multivariate Gaussian data. *Pattern Recognition*, *10*(5), 365–374.

Lin, Y. (1999). *Support vector machines and the Bayes rule in classification*.

Lucey, P., & Kanade, T. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops* (pp. 94–101).

Marron, J. (2015). Distance-weighted discrimination. *Wiley Interdisciplinary Reviews: Computational Statistics*, *7*(2), 109–114.

Ojala, T., Pietikäinen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. *Pattern Recognition*, *1*(582-585).

Pudil, P., Ferri, F., Novovicova, J., & Kittler, J. (1994). Floating search methods for feature

selection with nonmonotonic criterion functions. In *Proceedings of the Twelveth International Conference on Pattern Recognition, IAPR*.

Quiao, Z., Zhow, L., & Huang, J. (2008). Effective linear discriminant analysis for high dimensional, low sample size data. *Proceeding of He World Congress on Engineering*, *2*, 2–4.

Rifkin, R. (2004). In Defense of One-Vs-All Classification, *5*, 101–141.

Rifkin, R. (2008). Multiclass classification. *Lecture Slides, Statistical Learning Theory and Applications, February*.

Viola, P., & Jones, M. (2001). Fast and robust classification using asymmetric adaboost and a detector cascade. *Advances in Neural Information Processing System*, *14*.

Weston, J., & Watkins, C. (1999). Support Vector Machines for Multi-Class Pattern Recognition. *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN-99)*, (April), 219–224.