Interpolatable Classifiers for Patient Adaptation

Earlier work at Siemens on optimized patient adaptation for ECG arrhythmia detection:

This preliminary work on optimizing patient adaptivity was done at Siemens Corporate Research, after starting a joint project with Siemens Medical Electronics (SME) to look at the possibility of optimizing the arrhythmia detection performance of the ICU / CCU bedside monitor. One of the primary issues affecting classifier accuracy for this particular monitor (and almost certainly all others) is the ability to track drift in normal beat morphology over relatively long periods of monitoring. This is the original motivation for work along two lines: decreasing inter-patient variability in performance, and increasing the ability to adapt to new patients both during a learning phase and then continuously during monitoring.

In that preliminary work, two concepts were combined – the first of these being the use of a single localized linear discriminant [1] as the function that performs normal vs. ventricular beat discrimination. The second concept is a method of training that combines an aspect of patient independence (weight tying across patients) with a patient dependent origin translation in feature space. This combination yields a linear discriminant which combines training data over all patients to optimize feature weighting but allows patient dependent adaptation, both during a learning phase, and then continuously in an unsupervised fashion, by centering (during the learning phase) and then tracking the patient's normal beat cluster using the localization function. This simple method utilizes a single hyperplane normal vector, which is “cloned” across all patients in the database during training, while allowing the origins of each patient-specific localized discriminant to "float" adaptively in the feature space independently for each patient. For each “new” patient (not in the training data), the final step was the use of a small amount of “learn mode” data from that patient, as was part of the Siemens bedside monitor setup protocol for each new patient to be monitored. This new patient adaptation data was used to adjust the patient specific origin, localization function widths, and a feature selection step. Despite the extremely simple architecture, this method significantly outperformed both the existing SME classifier and a standard issue MLP (of that era) using ECG data from the MIT-BIH arrhythmia database (unpublished). The original plan to implement unsupervised updates to the localized discriminant function, weighted by classification confidence, was never implemented due to time constraints of the project, which is one of the motivations to continue work on this architecture.

The remainder of this note will discuss a much expanded concept for adaptation to gradual pattern drift in just such an application. The problem is then broken down into two parts:

- Synthesizing an initial classifier using a method of classifier interpolation. The intent is to optimize performance in a way meant to be specific to a new patient, based on multiple classifiers with identical architectures, having been trained on other patients.

- Using a classifier with is “drift aware”, in order to be able to track gradual drift in the patterns of normal variation of the patient state in a way that optimizes detection of pathological conditions (such as an arrhythmia) in a dynamic monitoring situation. This technique might also be used in situations involving taking ongoing, but not necessarily continuous, data to assess the progress of a patient that may be moving towards a disease state. The proposed classifier architecture, “A Network of Layered, Localized Multi-scale Decision Boundary Components for Gradual Pattern Drift Adaptation”, is described in a companion document.

Interpolated classifiers:

Synthesis of an optimized classifier for a new patient, or new operating condition, starts with a set of classifiers, all having the exact same architecture, but trained independently on other patients, or under a range of conditions. The original motivation for this concept was the observed need for classifier synthesis in two exemplary cases:

- Graceful modulation of the internal parameters of a classifier to handle gradually changing operating conditions that change relatively slowly over time.

- Rotation of hyperplanes and adjustment of nonlinearities in synthesis of an optimized classifier for a new patient, through interpolation of the parameters of classifiers trained on similar sets of patients.

The most recent inspiration to continue work on this architecture was from a missile warning application, where a missile detection algorithm needs to be operating at the highest level of accuracy while the aircraft is flying through varying atmospheric conditions. In this example, if you could train a set of architecturally identical classifiers under a range of atmospheric conditions, then you can envision the scenario in which the aircraft obtains measurements describing the current atmospheric conditions, and then updates all parameters of the classifier effectively by interpolation, using a convex combination of the parameters associated with the trained classifiers which are at the nearest neighboring points in a grid of atmospheric conditions. In this case, the parameter updates derived from the nearby classifiers would essentially just be incremental adjustments to hyperplane orientations and positions, localization function shapes and sizes (if used), along with the gain parameters of nonlinearities.

There is also a practical benefit to training a set of classifiers in a way that disentangles training data from different conditions (or patient pathologies) – because the training data is not pooled across conditions or patients. The hypothesis is that training for each network separately takes place using data which has cleaner, less complex decision boundaries. This results in classifiers of lower complexity, which, at least theoretically, have better generalization performance.

In doing searches for "interpolatable classifier" I have only turned up two papers over the past several years. A description of a similar technique is in [2]; I was not aware of this work when I came up with this proposal, which fortunately helps to validate the concept.

It is thought that this technique is also be applicable where classifiers may not be able to be placed on points within a space with relatively linear relationships like that of atmospheric conditions. This would much more likely be the case for patients with varying pathologies, and in fact, variable combinations of pathologies. For that application the proposed method would be to obtain training data from the new patient and then use the ensemble of pre-trained classifiers to label the data from this patient in a “learn mode”. A set of the best performing classifiers would be selected, and the new, patient specific classifier would be synthesized using a convex combination of the parameters of the selected set of neighboring classifiers, possibly weighted by their performance on the “learn mode” dataset. Eventually, for some given application (like ECG monitoring), a large set of trained classifiers would be available in a library, further improving the process as the sampling of this particular feature space by the set of classifiers becomes more dense.

The earliest idea for an interpolated classifier originates with the work I did on ECG arrhythmia detection at Siemens, with the goal of higher accuracy classification of abnormal beats in the ICU. The thought at that time was that a classifier might be more accurate if, instead of being trained on data that simply pooled across multiple patients, it could be trained in such a way that each patient's data was treated separately, resulting in a common network architecture which can be run with patient-

specific parameters to optimize accuracy. This comes from a very simple observation. Taking the most basic two-class example (and with only two clusters of data, one for each class), it's much easier to find maximum margin linear discriminants, along with wider margin widths, for each patient independently, as opposed to training on the pooled patient data. It's easy to visualize that, for each patient the margins are nice and wide, but the orientation of the margins vary with each patient. Even assuming that the data from each patient is normalized before pooling, pooling all of this data together would result in the overall margin shrinking, if not disappearing outright. Add in the effect of the distributions of the data from each class changing a bit with each patient, then the origin of each set of data can shift as well, resulting in further degredation of the margins. A network could readily be able to "solve" this problem using a more complex decision boundary, but it will likely result in reduced width margins, and theoretically lower generalization performance. This might also be viewed as a way to increase the usable ratio of training data complexity to network complexity -- if a simpler, interpolatable network can achieve the same error rate as a more complex, non-interpolatable classifier, then it should also have the potential for better generalization performance.

Continuing with that simple example, if you imagine then, a single linear discriminant being shifted and rotated into a position (in an effective feature space) which is optimized for each new patient, as opposed to a much more complex network, implementing a complex decision boundary with reduced margins (because it has been trained on a pool of data from all patients). From this perspective it is easy to imagine either a higher false alarm rate or lower arrhythmia detection rate for the more complex classifier trained on pooled patient data, as opposed to the simpler network synthesized for each new patient. It is also easier to imagine a way to effectively adapt the simpler, patient specific network over time (adapting to relatively slow physiologically based changes which are not meaningful as far as arrhythmia detection), in comparison to a method to adaptively update a much more complex network trained on pooled patient data.

So there are potential benefits to be had through independent training by patients , if a common network can be utilized:

- A simpler architecture, theoretically resulting in better generalization performance
- Larger margins, which should also result in better generalization performance

Proposed application of the localized linear discriminant network:

Envisioning this approach was a natural result of the work I did in development of a classifier architecture which is a network of localized linear discriminants (the “LLDN”*, [2]). In that network, each hidden unit is (roughly speaking) a hybrid of a linear discriminant, two Gaussian radial basis functions, and a sigmoid. These hidden units are the product of localization of the training of each discriminant and also localize (in feature space) their impact on the network output. It is a simple step to visualize a set of these localized discriminants being shifted and rotated in feature space in a process of adaptation, which is how the original concept arose. It is also easy to see that the use of localized hidden units reduces the level of interference between hidden units that may be the result of interpolation. There is a companion note to this one on the extension of this network architecture to a multi-scale implementation intended to optimize performance with gradual pattern drift. The multi-scale architecture may have the additional benefit of allowing gradual optimization of the initial, interpolated network parameters.

Potential application of deep networks:

It is possible that the deep convolutional network architectures might also be perfectly appropriate for linear interpolation of network parameters to accomplish the same goal. One of the reasons I

developed the localized linear discriminant network was to avoid the problem of hyperplane interference. In the standard MLP of the first wave of neural networks, complex decision boundaries have to be formed from the joint orientation and positioning of the hyperplanes associated with the hidden layer(s). Since hyperplanes have infinite extent in feature space, they may have impacts on more than one portion of the overall classification problem, and the proper co-ordination of interacting hyperplanes of infinite extent is one of the problems in training the standard MLP, from my perspective. One very attractive property of the convolutional network architectures is the fact that the convolutional units are effectively filters which operate in a sub-space of the overall feature space, and literally do not interact with most of the other convolutional hidden units – which, in my relatively naive belief, may be one of the reasons that they have had much greater success in deep architectures (impossible to train in the earlier days of MLP's).

It may be that some of the architectural / deep learning specific training features of the current deep networks like RELU's or other hidden unit output nonlinearities, max-pooling, dropout, etc., would prevent effective use of parameter interpolation, but it's not immediately obvious (while being only a casual observer of deep networks at this point). Returning to the example of atmospheric conditions, as long as all parameters are interpolated between pre-trained atmospheric grid points, the interpolated parameters such as the convolution kernel weights, offsets, and gains should not change the operation of the network in any fundamental way, and simply allow the network to be optimized for it's current atmospheric operating conditions. Although the presence of non-linearities probably makes the ideal interpolation non-linear, I'm proposing that any sort of interpolation will yield a network which performs better at the particular conditions it is operating under than a standard network trained on data pooled over all operating conditions.

In order to develop the set of trained classifiers, returning to the example of atmospheric conditions, a "primal" network would be trained using a set of median atmospheric conditions, and then incrementally trained with nearest neighbor atmospheric conditions to establish the set of grid points in between which the network parameters would be interpolated. The set of trained network parameters (i.e. weights, offsets / thresholds, and gains) would be incrementally acquired as the "primal" network, trained at the median atmospheric point, and then incrementally trained as the nearest neighbor points in the grid are traversed, moving outward to fill in the grid with snapshots of the trained networks at each grid point. Incremental training would guarantee that each hidden unit would essentially have the same function in the network, and not suffer the fate of being "re-purposed" each time a network is re-trained. Note that incremental training is also efficient in that the initial network weights are not randomized, but start from the final set of trained weights of the nearest neighbor network.

For patient adaptation, as previously mentioned, there is certainly no “grid of pathologies” that can be utilized, due to the complex relationships between pathologies, in comparison to simple axes like temperature, humidity, angle of light, etc. In this application, the process of starting the training with a “median classifier” is more complicated, but potentially doable. After training on all available patient data, the deep network with the set of parameters closest to the means of each parameter might be chosen, but this may not be feasible nor critical. Since the architecture of, say, a CNN is fixed, then the next patient chosen to update the CNN incrementally would simply be the patient out of all remaining patients which, when used to update the CNN, results in a minimal change in the CNN parameters. That process would be repeated, although multiple branches from each patient would be allowed. The goal is simply a set of CNN's which have minimal deviations from each other as you move from patient to patient. If the localized linear discriminant network architecture is used in this application, the networks arising from each patient will be distinct. This scenario would likely call for the “seed” network to be an aggregate of the individual patient networks. This might be done, for example, in such a way as to cluster the localized linear discriminants and use the centers of gravity of each cluster of the discriminants pooled over the patient dataset. In that way the architecture which can handle the most difficult patient (in terms of the required complexity of the decision boundary for that patient) will be in place, allowing adaptation to patients with less complex decision boundaries. In

this case, hidden units which are essentially not used for a given patient may be zeroed out for generalization performance, and simply do not contribute to the network interpolation process.

[1]: "Multi-View Object Detection by Classifier Interpolation",
Xiaobai Liu , Haifeng Gong, Shuicheng Yan, Hai Jin
http://grid.hust.edu.cn/xbliu/papers/ICASSP_camera.pdf

[2]: “A Network of Localized Linear Discriminants”, Martin S. Glassman, NIPS '91
https://papers.nips.cc/paper/5

*Nobody has ever used this term but the author, so I'm desperately trying to advertise it, most likely to absolutely no effect.