Networks of Multi-scale Localized Linear Discriminants for Concept Drift Adaptation and Synthesis of Classifiers for Optimized Adaptation to New Patients: Example Application to Continuous Glucose Monitoring

These concepts evolved from research on patient adaptive classification for arrhythmia detection. In the Learning Department at Siemens Corporate Research (Princeton, NJ), I developed an approach to concept (pattern) drift to better track normal beat QRS complex morphology while achieving more accurate detection of arrhythmias in the Siemens ICU / CCU bedside monitor. This extremely simple algorithm, utilizing a single localized linear discriminant, incorporated weight tying across patients for discriminant training, coupled with learning patient specific origins and localization functions in feature space with patient-specific feature selection on a very small set of “learn mode” examples. That algorithm yielded a reduction of 4.66x in error rate using a select (including some of the most difficult) set of 20 patients from the MIH / BIH arrhythmia database, compared to the standard Siemens Medical Electronics algorithm. I've been interested in this problem ever since.

Recently I have come to realize that the problem of adaptation to concept drift is a much more general problem, and, for example, may apply to applications such as autonomous driving under varying atmospheric conditions. (I will use that more widely used term, although “pattern drift”, or “dataset shift”, also terms in use in the literature, give a better picture for the intended applications). It is under such conditions of drift that the integrity of the between-class boundary (margin) regions must be maintained for optimum performance of a classifier. I am proposing two synergistic architectures do address this issue in a complementary way. Both of these proposals represent extensions of a network architecture and training algorithm for classification that I presented at NIPS:

https://papers.nips.cc/paper/525-a-network-of-localized-linear-discriminants.pdf

After an overview of these two methods that utilize this basic network architecture / training algorithm, and a brief description of how they work together in a synergistic way, I have included a discussion with some details for potential application to continuous glucose monitoring. I have also written up more extensive expositions of these two concepts, and I can send those along if there is any interest in additional detail.

Multi-scale networks for classification and synthesis of new classifiers by interpolation as companion systems:

The first of these proposals extends this architecture to a multi-scale, multi-layer network that decomposes decision boundaries into localized, multi-scale components, and utilizes the multi-scale nature of the resultant set of localized linear discriminants (LLD's) to theoretically optimize the process of concept drift adaptation. (LLD's can be thought of very roughly as hybrids of linear discriminants, sigmoids, and radial basis functions). The companion proposal involves the simultaneous, partially coupled training of a set of localized linear discriminant networks (LLDN's), each trained on disjoint sets of data. In application to glucose monitoring, for example, the data would be isolated into disjoint datasets of diabetes subtypes, and possibly further subdivided statistically using some form of clustering. Training multiple classifiers in parallel using disjoint datasets is a technique that is capable of distilling commonalities across the datasets, resulting in robust discriminants, while at the same time resulting in classifiers tuned for subtype / cluster specific classification. In the proposed method of application, a new classifier would be synthesized by interpolation of parameters between classifiers which are nearest neighbors in “disease space” to the new patient. A synthesized, multi-scale LLDN, would then represent a theoretically optimized classifier for each new patient, followed by adaptation to slow drifts in patient states. Causes of slow drift in glucose monitoring could be due, for example, to changes in weight, diet, metabolism, sleep duration and quality, activity / exercise levels, etc., that almost are certain to occur for every patient simply due to, among other things, getting feedback from the glucose monitor itself.

(cont'd on next page)...

Networks based on multi-scale decomposition of decision boundaries:

The primary themes of the multi-scale concept (pattern) drift adaptation architecture are:

- All drifting feature vector data should be incorporated in the updates of a classifier in the attempt to maintain the integrity of large margin decision boundaries
- Decision boundaries in a feature space can be decomposed into a set of multi-scale components
- Coarse scale pattern drift can be used to:
  - Update the coarse scale components of decision boundaries
  - Update the finer scale components in concert with their neighboring coarse scale components
- Concept drift adaptation occurs at differing rates over scale, inversely proportional to the scale of components of the decision boundary
  - Coarser scale components are effectively updated with higher learning rates as they cover relatively larger regions of feature space
- Adaptation occurs in a hierarchical fashion from coarse to fine
  - Coarser-scale adaptations shift and rotate finer scale components to maintain the large margin decision boundaries when the receptive fields of finer scale components do not capture new feature vectors
  - Finer scale components are adapted, independently of coarser scale adaptations, with learning rates relative to their contribution to classification confidence
  - Finer scale components become the coarser scale components for decision boundary components at the next level of detail of approximation in feature space
- Learning rates are scaled based on the confidence of individual classifications in the online adaptation setting (in practice, low confidence classifications will not be used for network adaptation)

Note that this approach *does not refer to the use of multi-scale features*, although such a front end would likely be beneficial and would in no way interfere with the decomposition of decision boundaries into multi-scale components. This decomposition, while inspired by the way that signals or images can be decomposed using wavelets, instead uses localized linear discriminants (LLD's) as locally data adaptive kernels, which are biased during network construction first towards coarser scales, followed by the addition of successively finer scale LLD's, driven by the goal of error reduction at the network output. Iteratively adding finer scale LLD's to focus on error is quite similar to way the Adaboost algorithm iteratively focuses on error in the training data. A multi-scale approach to decision boundary decomposition also offers the potential for lower overall network complexity, with the accrued theoretical benefits to generalization performance. But more specifically, it is possible to optimize parameters of each LLD component through cross-validation on data localized to each distinct region of feature space. Since, in any non-trivial problem, the distributions of data from each class will vary across feature space as they represent clusters of data from distinct sub-problems of the overall classification problem, optimization of kernel parameters should also be localized in feature space for best generalization performance. Note that it is also possible to perform feature selection for each local discriminant with no loss of generality, since this merely sets some of the discriminant weights to zero, simply removing the contribution of those features to the output of that particular LLD node.

Adaptation to concept drift using multi-scale networks:

A fundamental hypothesis is that in order to properly track the drift in the margin region, both the coarser-scale drift of clusters of data must be taken into account in concert with drift at the finer scales (represented by the support vectors lying nearest the margins). Since all data from a given class, or cluster, originates from the same source, optimized drift detection will utilize all data from the source – although in a differential fashion based on scale and relationship to the local margins. Adaptation will occur based on confidence of the classifier to each new input so that a consistent sequence of lower confidence classifiers can adjust network parameters to the same degree as a single, higher confidence classification, but all adjustments will be made conservatively to account for an expected error rate in the field. Note also that adjustments primarily affect only the local portions of the decision boundaries that are closest to each newly acquired feature vector, so that ripple effects throughout the network are minimized -- which is one of the main benefits of networks of localized kernel functions. While this algorithm was inspired by classification problems such as arrhythmia detection, where patterns from one class, like normal QRS complexes, can drift slowly with circadian rhythms and other physiological parameters affecting the electrical properties of the myocardium, it is potentially useful for adaptation in other domains where the nature or distribution of background noise changes slowly, another factor driving the requirement for updated decision boundaries for optimal classifier performance in the field.

Clustering patient data using classification error:

Since the goal of a classifier used in a continuous glucose monitor would be that of generating warnings (alarms) that a patient was likely to become either hypo- or hyper-glycemic, and that these alarms would be based on the (compensated) glucose sensor signal levels within some time window (or windows of multiple lengths) of recent history, it makes sense to cluster patient data for training based on this data alone. And while clustering could easily be done in a traditional fashion using features extracted from the glucose sensor samples, it makes even more sense to cluster this data using a “wrapper” method incorporating training of the classifiers, in a similar fashion to doing feature selection using the wrapper method with a particular type of classifier architecture / training algorithm. Clustering patients would then proceed in an agglomerative way, by iteratively adding patients to a cluster based on the results of the incremental training of each classifier (representing the existing clusters) and choosing the cluster which yields the lowest cross-validation error rate that is below an acceptable threshold – or else starting a new cluster if the cross-validation error rate exceeds the acceptable threshold. In this way patient data is clustered in a way most directly meaningful to this application.

Synthesis of new classifiers by interpolation over a set of pre-trained classifiers:

The second of the two complementary algorithms is based on the concept of new classifiers that are synthesized using parameters interpolated from a set of pre-trained classifiers, for optimized initialization and adaptation to a new patient. Here the basic idea is that classifiers that are trained on smaller datasets, each focused on individual clusters of patients (which may represent distinct pathologies or simply clusters), will generally have lower complexity due to a lower level of interference from other pathologies / clusters which can cause interference in the margin regions of the decision boundaries. Viewing classifier training as a process of approximation of the true decision boundary, the use of training data pooled over all patients is likely to require a greater number of components in order to define the decision boundary in sufficient detail to achieve a target approximation error. The hypothesis is that the required complexity to approximate the decision boundaries (to the same target error) for discrimination within each distinct dataset, trained independently, will be substantially lower. During simultaneous training of a set of classifiers, each using a disjoint set of training data from distinct sources, the updates to each of the localized linear discriminants are partially tied to other localized linear discriminants using a “coupling parameter” based on proximity in feature space of the local clusters of data and margin orientation. This allows training updates to be shared across datasets through a “partial weight tying” method, where the strength of the weight tying is based on the similarity of local clusters of data across the isolated training datasets. By this method, data from patients with distinct pathologies or morphologies do not clutter up the margin regions due to being pooled in a single dataset. Instead, partially shared training of localized discriminant functions takes place in parallel across isolated datasets where the support vectors from each dataset do not directly interfere with each other, yet allow similar but distinct versions of discriminant functions to evolve – partially independently, but also partially jointly, based on the value of the coupling parameter.

This not only allows for classifiers that have theoretically better generalization performance for the given pathology or operating condition they are trained for, it also allows for a patient adaptive classifier that can be synthesized by interpolation of the parameters within a set of the “nearest neighbor” pre-trained classifiers that best match the new patient's data. Since it is generally possible to obtain only a small amount of data from a new patient in the “learn mode” of a system like a bedside monitor, utilizing the results of training on similar patients allows the effective use of much more, and carefully labeled training data, as the pre-trained classifiers represent knowledge about the patient clusters embedded in the structure and parameters of these classifiers. Only the classifiers that are nearest neighbors are selected for use in interpolation to generate the new patient classifier. The same incremental training technique just discussed for clustering would be used to determine the nearest neighbor classifiers. Alternatively, if a sufficient amount of labeled “learn mode” data is not available, unlabeled data could be used for selection of the nearest neighbor classifiers, using the classification confidence measures accumulated for each pre-trained classifier over the unlabeled samples from the new patient. Since the LLD components of each classifier represent a combination of local density estimates and linear discriminants, confidence measures from this type of network are a function of both proximity to clusters of training data, and relationship to the local portions of the decision boundary region (local margins). Unlabeled feature vectors from a new patient will achieve highest confidence measures when they are in close proximity to one of the clusters of training data, as well as sufficiently far (on either side) from the decision boundary margin region. Distributions of unlabeled data from new patients will have aggregated higher confidence measures for pre-trained classifiers for which the data matches these two conditions, and this will be a good indicator, even with the lack of labels, to select the pre-trained classifiers to be included in the set used to synthesize the classifier for the new patient.

Distinction with ensemble methods and the mixture of experts model:

This structure is similar to a mixture of experts model, with the distinction that the supervisory network does not simply gate / weight the outputs of a set of individual experts / networks, but rather synthesizes the parameters for a single (new) network through interpolation. In this way the synthesized network is adjusted for optimized performance with each new patient through rotations and shifts of decision boundaries along with the width parameters of the localization functions, and the gain adjustments for the sigmoid nonlinearities. Therefore the output of the overall classifier is not just the result of a weighted / gated ensemble of individual networks, but is pre-tuned to operate in a more focused way for a new patient. One form of mixture of experts model could be viewed as a weighted k-nearest neighbor algorithm, with more weight being given to the “experts” (classifiers) which have more relevance to the new patient. That relevance could be the distance in a separate feature space, like say a disease parameter space using parameters indicative of diabetes types, but distinct from the features extracted from windowed sampled data from a glucose sensor. The relevance measure could also be a classifier-based confidence measure like the one just discussed using “learn mode” data. But using a weighted combination of the outputs of the most relevant classifiers cannot correct errors generated at earlier layers within each classifier – at best, the overall output of the mixture of experts model will be a weighted average including more and less patient relevant classifiers. This is exactly the same problem with any ensemble method, including bagging and boosting; none of these ensemble techniques have the ability to synthesize a single new classifier that is “tuned” to the specific data of the new patient through the process of interpolation of classifier parameters. That is not to deny the obvious benefits of ensemble methods (from one who has used them extensively), but simply to point out the analogy that taking a weighted average of neighboring data points is not the same as taking the value of an interpolated function at a sample point not in the training data. While interpolation functions not well matched to the problem at hand can clearly generate outputs with higher error than simply using a weighted average, the concept of interpolatable classifiers has hardly yet been explored, and this proposal offers nothing more than a conservative approach – a simple weighted averaging of neighboring classifier parameters in order to synthesize a new classifier.

Partially coupled training of classifiers with distinct structures and the process of interpolation:

It is not necessary for the pre-trained classifiers to have identical structures in order to generate a new network through the process of interpolation. During parallel training of classifiers using disjoint datasets, each LLD will share training updates in a weighted fashion based on the relative strength of the coupling to neighboring LLD's (in feature space) across all classifiers being trained in parallel. Since most non-trivial classification problems are really composed of sub-problems, allowing the LLD components of each classifier to (partially) share weights with LLD components solving the same discrimination sub-problem is an optimized use of training data. While two classifiers trained for two distinct patient types may have rather distinct structures in feature space, they may both share relatively similar clusters of data in feature space that represent a common sub-problem in discrimination – it then makes sense for them to share this training data through partial coupling, during training, of the LLD's which have each been instantiated to handle these similar clusters of training data. The next logical extension of this illustration is to consider an example in which one of two similar clusters, in one of two patient type specific datasets, has been clustered itself as distinct sub-clusters. In this case, the LLD being trained on the data from the single cluster in the first dataset can be coupled to each of the two LLD's being trained on the sub-clusters of similar data in the second dataset. Again, this coupling will be weighted based on the similarity between the coupled LLD's.

These two bases cases then serve to illustrate how classifiers which will develop distinct structures in terms of their localized kernels (LLD's) can still partially share training data in order to make the best possible use of data, which in an application like this, is often difficult and tedious to acquire and label correctly. By definition, the flexible structure of networks which are incrementally constructed by adding local kernels associated with local regions of a dataset in feature space, allows a greater degree of adaptivity to the unique structure of the data in each isolated dataset. Interpolation between networks with different numbers of components can be accomplished by simply allowing interpolation between each component in a given network with the nearest neighbor components of the nearest neighbor networks.

This can be visualized as each LLD component generating clones, one for each neighboring LLD component in a neighboring network, each of which transforms to become more like the neighboring components as the interpolation point at which the new network will be synthesized moves closer to a neighboring network. The transformation of each LLD component consists of a shift in origin and rotation of the normal vector defining the local hyperplane surface, along with the size and shape of the localization kernels (as the parameters of the two Gaussians and one sigmoid are interpolated). Since this, like other RBF networks, is a shallow network, the only other parameters to be synthesized are the weights to the output node(s). In the simplest example, as two “clones” of an LLD shift towards two neighboring LLD's, the weights of the new output node will initially be identical, summing to the original weight of the single LLD to it's output node, but transforming to the unique weights to the same output node as these “clones” transform to be more like the neighboring pair of distinct LLD components. If an LLD does not have any reasonably close LLD's in neighboring networks for interpolation, it's clone will simply be added, uninterpolated, to the new network, replicating it's weight to the output node. Output nodes can approximate both “and” and “or” functions. In the “and” case, the summed contributions of multiple hidden nodes are required for the output threshold to be crossed; the hidden units need to cooperate in agreement. In the “or” case, the output node can respond to any of a number of hidden units which provide confidence in a decision independently. Adding a unique, isolated LLD to the new network being synthesized (while not being interpolated) is akin to adding a new case to the “or” function represented by the output node, and therefore the weight from that isolated LLD to the output node can be simply copied as well.

So, while the complexity of a network synthesized by interpolation between several nearest neighbor networks can be higher than the most complex of the set of nearest neighbor networks, due to the complexity lowering effect of isolating datasets with partially coupled training, the overall complexity of an interpolated network may still be lower than one trained on pooled data. From a practical standpoint, since by definition neighboring networks are likely to be developed on similar patient data, the difference in structure between neighboring networks is likely to be fairly low, simplifying the structures of a new patient network as it is synthesized by interpolation between similar networks. Most importantly, the localized components of the synthesized network are be optimized for a new patient in a way simply not possible by using pooled data to train one single network, or even choosing from a set of pre-trained networks. Combined with the previously described multi-scale approach to concept drift adaptation, this approach yields the potential for smoothly adjusting network parameters as the parameters for a given patient change over their progress towards health and away from disease. In the limit, if a large number of patients representing different pathologies were available, this process would approach the much simpler process of simply selecting the pre-trained classifier which is the best match for the new patient. Approaching that limit, though, increasing the amount of training data should allow the process of interpolation to be more effective as the sub-optimal nature of linear interpolation becomes less of a factor in the synthesis of the “true” classifier for each new patient.

Localized kernel networks and interpolation for sensor signal compensation:

While accuracy in measurement of blood glucose is important for patients and their heathcare practitioners, linearization of sensor signals is not necessary prior to it's use with a classifier for alarm generation. (Calibration, is of course necessary). Regarding this independent problem of measurement linearization, the localized linear discriminant network is, more generally in the family of networks of localized kernel functions which can also be applied for sensor signal compensation. This could be as simple as operating in the single dimension of the raw sensor signal, and generating an output which is a convex combination of local regression functions (probably just linear regressions). It is also possible that the input is multidimensional, including patient parameters which might modify the resultant signal compensation function. The amplitude of the associated localization functions (most likely a Gaussian centered on each segment of the curve associated with a different local, linear regression function) would act as weights for the convex combination, providing a smoothly varying output as the signal moves over it's dynamic range. If a set of these networks for compensation were generated, one per patient cluster, then an identical process of interpolation between a set of the nearest neighbor localized regression networks should be useful in synthesizing more accurate, patient specific networks for blood glucose sensor signal linearization.

Continuously updated classifiers in the glucose monitoring application:

Since glucose levels are being sampled continuously over the course of the day, it is possible to use that data in an ongoing basis to generate labels for online updates to the classifier. These labels, “normal” or “hypoglycemic”, for example, can be applied to each sample, based on the measurement results. While a classifier would be initialized, say, to generate an alarm warning of impending hypoglycemia, and can operate with a reasonably low error rate, a classifier which can be updated on the fly using an online learning approach can theoretically improve it's performance in generating alarms as data from the patient collected in real time is incorporated. Since the LLD, the fundamental component of the network described in the NIPS paper (hyperlink above) is capable of online updates, it is a good candidate for this application. But the design of the multi-scale aspect of training and updating this classifier takes it to another level with respect to adaptation to drifting patterns, and the ability to maintain the integrity of the between class boundary regions that comprise the overall decision boundary. There would seem to be a number of relatively slow sources of drift, which is the ideal scenario for this adaptive classifier design. They might include changes in diet, weight, activity / exercise level, sleep quantity and quality, and the patient's own work on lifestyle modifications which may interact in a complex way with other factors, including the feedback they get from the glucose monitoring system itself.

Each sample, acting as a label, is associated with the feature data which is extracted from data within a time window preceeding the sample (there may effectively be multiple time windows, since there are clearly events at multiple time scales which impact blood sugar levels). Assuming for the moment that these features would be CWT wavelet coefficients, and due to the partial independence of wavelet analysis within each frequency band, coefficients can be collected efficiently. Coefficients associated with wavelets which remain in the analysis window, as it shifts, do not have to be recomputed when the window shifts by the number of samples equal to the support of each wavelet (so sets of coefficients would be retained, covering the length of support of each wavelet). This is in contrast to the use of, say, FFT coefficients, which would have to be re-computed with each new sample.

The time window for feature extraction could potentially be extended over a much longer period than a day, if there proves to be predictive information in the variation in patient parameters over multiple days that would bias the patient towards an event for which an alarm should be generated. Use of an ensemble of classifiers, each trained on and operating within a different time window could be effective here. I've been using ensembles of classifiers since my work in speech recognition, and have just completed a classifier for sensor calibration data at BAE Systems which incorporates an ensemble of diverse classifiers quite effectively.

The classifier updates would be done on either the wearable monitor, or possibly a patient's smartphone (although we should not assume everyone has a smartphone), to take advantage of the significant computing power of these devices. The only data needed to be stored is the sampled data over the course of the time window determined to carry relevant information for prediction of hypo- and hyperglycemic events. The update algorithm would run as a background task, with the classifier being updated whenever each set of new parameters is available from the multi-scale concept drift algorithm. Since the updates are fundamentally based on a simple large margin / soft margin perceptron training algorithm at the core of training of the LLD's, and are naturally updated sample-by-sample, updated parameters could potentially be available after processing each new sample from the sensor, depending on sampling frequency and the resultant complexity of the classifier for a given patient.