Reducing Spatial Data Complexity for Classification Models

Dymitr Ruta* and Bogdan Gabrys†

*British Telecom Group CTO, Adastral Park, Orion MLB 1, PP12, Martlesham Heath IP5 3RE, UK
†Bournemouth University, School of Design, Engineering & Computing, Poole House, Talbot Campus, Fern Barrow Poole, Dorset, BH12 5BB, UK

Abstract. Intelligent data analytics gradually becomes a day-to-day reality of today's businesses. However, despite rapidly increasing storage and computational power current state-of-the-art predictive models still can not handle massive and noisy corporate data warehouses. What is more adaptive and real-time operational environment requires multiple models to be frequently retrained which further hinders their use. Various data reduction techniques ranging from data sampling up to density retention models attempt to address this challenge by capturing a summarised data structure, yet they either do not account for labelled data or degrade the classification performance of the model trained on the condensed dataset. Our response is a proposition of a new general framework for reducing the complexity of labelled data by means of controlled spatial redistribution of class densities in the input space. On the example of Parzen Labelled Data Compressor (PLDC) we demonstrate a simulatory data condensation process directly inspired by the electrostatic field interaction where the data are moved and merged following the attracting and repelling interactions with the other labelled data. The process is controlled by the class density function built on the original data that acts as a class-sensitive potential field ensuring preservation of the original class density distributions, yet allowing data to rearrange and merge joining together their soft class partitions. As a result we achieved a model that reduces the labelled datasets much further than any competitive approaches yet with the maximum retention of the original class densities and hence the classification performance. PLDC leaves the reduced dataset with the soft accumulative class weights allowing for efficient online updates and as shown in a series of experiments if coupled with Parzen Density Classifier (PDC) significantly outperforms competitive data condensation methods in terms of classification performance at the comparable compression levels.

Keywords: Data reduction, data condensation, Parzen density estimation, electrostatic field
PACS: 07.05.Kf, 07.05.Mh, 07.05.Rm

INTRODUCTION

Rapid increase of cheap storage space and computational power led to a massive expansion of terabyte data-warehouses and boosted the demand for accessing and processing these massive information sources. Data analytics and mining continuously gain in importance for virtually any business today [1]. On the one hand increasing number of automated intelligent data-driven processes require information processed in real-time, on the other hand complex analytical and predictive models are being deployed for a variety of on-demand services. All these technologies attempt to use the most advanced data mining and machine learning models, yet the disparity between the requirements and model capabilities is worryingly growing due to inability to process vast amounts of data in real time. Classification methods are at the very heart of pattern recognition and machine learning yet the most advanced models like neural networks, support vector machines or density based classifiers [2] are computationally very expensive with respect to the number of samples to be processed. Such classifiers struggle with the data sizes of the order of tens of thousands yet face now the data sources with millions or even billions or records. To deal with this problem the data needs to be reduced to the manageable size. The challenge is to achieve it with a minimum information loss and no harm on the subsequent classification performance.

The simplest and possibly the most common data reduction methods are random sampling techniques [3]. They are suitable for large datasets with simple structure but fail on smaller noisy data particularly with large density or class imbalances as they tend to ignore less populated data subspaces. It is believed that the proper condensation method should make use of all of the original data points [2]. The state-of-the-art data condensation methods in its majority try to select a subset of data that maximally retains the original data density or other characteristics like quantisation error etc [4]. A number of methods in this group work around the principle of removing samples from the less dense regions in favour of strengthening the evidence in the denser regions such that the deviation from the original data density is as minimal as possible [4], [6]. Other methods use multi-resolution spatial analysis to split data into partitions or clusters [7] and use centres of these clusters as new condensed dataset.

CP963, Vol. 1, *Computational Methods in Science and Engineering, Theory and Computation: Old Problems and New Challenges,* edited by G. Maroulis and T. Simos

Classification-based condensation methods are relatively new and serve directly the purpose of retaining or improving classification performance given the reduced training set. Typical examples of such classification focussed condensation are the reduced nearest neighbour rule, iterative condensation algorithm [5] or locally variable condensation models based on neural networks [8]. In all these efforts none of the methods try to actively change the data from its original position. It is reasonable to assume that releasing data from their original positions could further improve classification performance or at least allow for further condensation given similar classification performance. In [6] Girolami and He obtained improvement of the Parzen density fit of the reduced set by finding the optimal data weights on the course of constrained optimisation process. A natural extension of such model would be to include data vectors themselves into the variables to be optimised yet this would undoubtedly trap the process into large number of local optima building up on the excess of the degrees of freedom for this very-high dimensional search space. In a response to this challenge a new condensation model is here proposed which applies electrostatic-like data field to condense and transform labelled dataset in order to retain or boost the performance of a classifier trained on such dataset. A set of 2 model variations is presented and tested in a form of dynamic iterative simulations carried out on standard datasets used for classification benchmarking.

The remainder of the paper is organised as follows. The next section discusses kernel-based data density estimation as a prerequisite to further analysis. The following section introduces main assumptions of the electrostatic interaction applied to labelled data. Then details of the dynamic condensation process are discussed in the following section along with the model update and tuning techniques. The next section shows the results of some comparative experiments demonstrating condensation and classification performance of the proposed model and follows with brief conclusions.

KERNEL-BASED CLASS DENSITY ESTIMATION

Kernel methods have been proven useful in a number of applications [9]. The rationale behind using kernels for density estimation is to ensure that every data point is actively contributing in the formation of the final density landscape. There is an extensive literature on kernel-based density estimation methods applicable to both labelled and unlabelled data [10]. In both cases Parzen Window Density Estimator (PWDE) with Gaussian kernels is widely used. However, in a business environment with massive temporally evolving datasets kernel based density estimation methods struggle on a number of fronts. First of all they rely on costly-to-obtain distance measures between all the pairs of data which scale rather badly with the dimensionality of the data. Moreover, it is not immediately possible to store a summarised density model and each time a density estimate is required it needs recalculation of contributions from all other data points in the input space. If on top of that the data itself changes for example along the time then the whole distance matrix needs to be continuously rebuilt which would most certainly be intractable for larger dataset. We propose a novel solution to all of these problems in the context of classification data where additionally it is important to make provisions for maintaining class separability and ensuring maximum possible classifier performance. By an analogy to the charged particles in the physical world one can consider the data as charged particles each being the source of a central field affecting other samples in the input space [11], [13], [14]. All the characteristics of such field are the results of the definition of a charge potential or kernel function and can be absolutely arbitrarily chosen depending on various priorities and requirements. We have shown in our previous work that one can construct a classifier or a clustering model using a dynamic data fields methodology [11]. In this work the focuss is on building the most compact representation of the labelled dataset that maximally retains the original class density landscape and thereby allowing for fast application of well performing classifiers. For that purpose we use Gaussian kernels for class density estimation and apply electrostatic data condensation in a specifically designed controlled environment.

Let $X^{[n\times m]}$ stand for the matrix of n m-dimensional data vectors $\mathbf{x_i}$ where $i = 1,..,n$ and let $\omega_i \in \{\Omega_1,..,\Omega_L\}$ mark the class label of the i^{th} data point. Following a Parzen Window approach [2] an estimate of the data density at point $\mathbf{x_j}$ can be obtained by calculating:

$$g_l(\mathbf{x_j}) = f_{jl} = \frac{1}{n}\sum_{i,\omega_i=\Omega_l}\frac{1}{V_n}K\left(\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right) \tag{1}$$

where V_n is a window volume, K is a specific kernel function and h is a smoothing parameter. Since density f shares the characteristics and properties of the kernel function it is often selected such that it has mathematically tractable properties, such as continuity or differentiability. For that reason a Gaussian kernel is an ideal candidate and as we show later it also acts as a perfect definition of class potentials to be used later in the condensation model. Given a Gaussian kernel the density function for each class takes the form:

$$g_{jl} = \frac{1}{2\pi^{m/2} h^m n} \sum_{i,\omega_i=\Omega_l} e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} \tag{2}$$

In the general case, instead of crisp labels, each data point $\mathbf{x_i}$ could be considered to belong to all the classes with various degrees of membership represented by some soft measures like fuzzy membership or probability measure. For each such data point one can assign a corresponding class membership vector $\mathbf{c_i} = [c_{i1}, .., c_{iL}]$ that form a class membership matrix $C^{[n\times L]} = [\mathbf{c_1}, .., \mathbf{c_n}]'$. In that case the class density function becomes:

$$g_{jl} = \frac{1}{2\pi^{m/2} h^m n} \sum_{i=1}^{n} c_{il} e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} \tag{3}$$

Note that if for each data point class memberships add up to a unit i.e.: $\forall_{i=1,..,n} \sum_{l=1}^{L} c_{il} = 1$ then the soft density estimation as given by (3) can be always applied even if only crisp class labels are provided.

Such class density estimates can be easily converted into a soft classifier. Let $p_{jl} = p(\Omega_l|\mathbf{x_j}, X, C)$ stand for an approximation of the posterior probability that point $\mathbf{x_j}$ belongs to class Ω_l. Applying a simple transformation and normalisation to sum up to a unit the class posterior probability p_{jl} can be obtained as follows:

$$p_{jl} = \frac{1/(1+e^{-g_{jl}})}{\sum_{k=1}^{L} 1/(1+e^{-g_{jk}})} \tag{4}$$

The optimal smoothing parameter h can be found through a simple leave-one-out maximum likelihood optimisation that minimises the classification error [2].

ELECTROSTATIC INTERACTION OF LABELLED DATA

So far we have shown how to obtain a class density at any point in space for the multi-class data. Now we show how to define an interaction among these data points that would try to reduce the spatial data complexity and at the same time retain the original class densities as much as possible, while additionally separate different classes from each other. In a search for such interaction we were inspired by the well known electrostatic field which causes the carriers of an electric charge to attract or repel depending on the sign of charge with the magnitude proportional to the squared inverse of a distance and the product of the charge absolute values. Formally the electrostatic field vector $\mathbf{E_j}$ at any particular point in space $\mathbf{x_j}$ emerges as a gradient of the potential V_j that was generated from the individual charge carriers' contributions i.e.:

$$\overrightarrow{E_j} = \mathbf{E_j} = -\overrightarrow{\nabla} V_j = -\left[\frac{\partial V_j}{\partial x_{j1}}, ..., \frac{\partial V_j}{\partial x_{jm}}\right] = \frac{\partial}{\partial \mathbf{x_j}} \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{n} \frac{q_i}{|\mathbf{x_j}-\mathbf{x_i}|} = -\frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{n} q_i \frac{\mathbf{x_j}-\mathbf{x_i}}{|\mathbf{x_j}-\mathbf{x_i}|^3} \tag{5}$$

where q_i are the charges of individual carriers and $1/4\pi\varepsilon_0$ is a field constant which we can drop for our analysis. The actual force acting on the carrier located in point $\mathbf{x_j}$ would depend on the field vector $\mathbf{E_j}$ as well on the size and the sign of the charge placed in the point $\mathbf{x_j}$ such that:

$$\mathbf{F_j} = q_j \mathbf{E_j} \tag{6}$$

Adopting the electrostatic model to the labelled data points one can consider class labels as sources of certain potential proportional to the soft class membership value $c_i l$ for each class. To encourage reduction of spatial data complexity the proposed interaction needs to result in attracting data from the same class while repelling different classes from each other. In order to achieve such interaction we embody Parzen density estimate into class-sensitive potential. In the calculation of the potential we assume that at the probe measuring point $\mathbf{x_j}$ all the classes are equipartitioned i.e. $\forall_{l=1,..,L} c_{jl} = 1/L$. The class sensitive potential becomes:

$$V_j = -\frac{1}{2\pi^{m/2} h^m n} \sum_{i=1}^{n} \sum_{l=1}^{L} \left[\underbrace{\frac{1}{L} c_{il}}_{attraction} \underbrace{-\frac{1}{L}(1-c_{il})}_{repultion}\right] e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} = -\frac{1}{2\pi^{m/2} h^m n} \sum_{i=1}^{n} \left[\frac{2-L}{L}\right] e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} \tag{7}$$

Note that electrostatic interaction between two points with soft class partitions effectively breaks down into an interaction between two sets of L points each with the weights equal to the class partitions. Given (7) and the assumption of equipartitioned probe data point the following holds:

- Given just 1 class factor $(2-l)/L$ becomes a unit and the whole field remains attracting only
- For 2 classes the whole potential vanishes to nil.
- For 3 and more classes factor $(2-l)/L$ becomes negative hence there will be always an excess of repelling interaction.

An attractive property of such representation of the potential is that the field vector $\mathbf{E_j}$ at any point of the input space can be immediately obtained by a simple calculation of the gradient from potential:

$$\overrightarrow{E_j} = \mathbf{E_j} = -\overrightarrow{\nabla} V_j = -\left(\frac{\partial V_j}{\partial x_{j1}}, ..., \frac{\partial V_j}{\partial x_{jm}}\right) = -\frac{\partial V_j}{\partial \mathbf{x_j}} = \frac{L-2}{\pi^{m/2} h^{m+2} L} \sum_{i=1}^{n} (\mathbf{x_j} - \mathbf{x_i}) e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} \tag{8}$$

The simple formula for the field potential given by (7) effectively stands for the energy of an interaction with labelled data $\mathbf{x_i}$ per unit of class-equipartitioned charge positioned in point $\mathbf{x_j}$. In the general case, however, uniform distribution of class partitions is a very unlikely simplification and the energy of interaction would have to be derived from sums of all individual pairwise interactions among class partitions over all the labelled data points in the input space i.e.:

$$U_j = -\frac{1}{\pi^{m/2} h^m n} \sum_{i=1}^{n} \sum_{l=1}^{L} \left[c_{jl} c_{il} - c_{jl}(1 - c_{il})\right] e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} = -\frac{1}{\pi^{m/2} h^m n} \sum_{i=1}^{n} \left[2\mathbf{c_j} \times \mathbf{c_i^T} - 1\right] e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} \tag{9}$$

where "$\times$" denotes standard matrix multiplication operator. Generalising even more one could assume that data points could have different charge weight. Thus, instead of assuming that $q_i = \sum_{l=1}^{L} c_{il} = 1$ there could be arbitrary different charge values q_i scattered in the input space and in that case the potential energy of interaction between points $\mathbf{x_i}$ with charges q_i and class partitions $\mathbf{c_i}$ and the point $\mathbf{x_j}$ with charge q_j and class partition vector $\mathbf{c_j}$ becomes:

$$U_j = -\frac{q_j}{\pi^{m/2} h^m n} \sum_{i=1}^{n} q_i \left[2\mathbf{c_j} \times \mathbf{c_i^T} - 1\right] e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} \tag{10}$$

Similarly, if the electrostatic field vector is applied to an arbitrary labelled data point $\mathbf{x_j}$ with a charge q_j and class partitions c_j the simple form of field vector as expressed in (8) becomes a force vector $\mathbf{F_j}$ which inline with (9) and (10) can be expressed by:

$$\mathbf{F_j} = -\frac{q_j}{\pi^{m/2} h^{m+2} n} \sum_{i=1}^{n} q_i \left[2\mathbf{c_j} \times \mathbf{c_i^T} - 1\right] (\mathbf{x_j} - \mathbf{x_i}) e^{-\frac{1}{2}\left|\frac{\mathbf{x_j}-\mathbf{x_i}}{h}\right|^2} \tag{11}$$

Note that the only quantity that decides about the direction of force is the term $2\mathbf{c_j} \times \mathbf{c_i^T} - 1$ describing distribution of class partitions while the charges themselves decide about the magnitude of force.

DYNAMIC CONDENSATION PROCESS

Given the forces acting upon all the labelled data points in the input space we can now release the data from their original locations and let them move along the data forces towards more stable lower energy states. In order to preserve the original class densities we propose two methods of constraining the electrostatic field: one through a direct interaction of the mobile condensing data with the fixed uncondensed original data and the second through a dynamic reevaluation of the mobile data class partitions $\mathbf{c_i}$ according to the original Parzen density classifier. Both methods allow to dynamically control and change the interaction among the moving data depending on the proximity to the other mobile data cross-referenced with the original data locations. On the course of such guided simulation the data particles are allowed to merge and join their class charges if the distance between them falls below an arbitrary threshold value d. The condensation process continues until the total energy stops falling i.e. the system reached the equilibrium state.

Before we formally describe the condensation process using the above mentioned two different options let us simplify the notation using matrix formulation. Let "$\circ$" denote element-wise or Hadamard matrix multiplication, and let $\mathbf{1}^{[n,m]}$ be an n by m matrix with all unit elements. Let us consider a generalised case in which an interaction is between the data stored in a matrix $X^{[n_X,m]}$ with the corresponding soft class charges $Q_X^{[n_X,L]}$ and the data $Y^{[n_Y,m]}$ with the class charges $Q_Y^{[n_Y,L]}$. Note that the class charge matrix is different from class partitions as it could accumulate class charges and unlike for class partitions their sum for individual points can exceed the unit. We also associate the mass of individual data point with the sum of charges such that the vector of data points masses will be denoted by $\mathbf{Q} = Q \times \mathbf{1}^{[L,1]}$.

Let A and B stand for the auxiliary matrices expressed as follows:

$$A^{[L,L^2]} = \begin{pmatrix} \overbrace{1\ldots1}^{L} & 0\ldots & \ldots & \ldots 0 \\ 0\ldots0 & 1\ldots1 & 0\ldots & \ldots 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0\ldots & \ldots & \ldots 0 & \underbrace{1\ldots1}_{L} \end{pmatrix} \quad B^{[L^2,L]} = \begin{pmatrix} 1 & 0 & .. & 0 \\ 0 & 1 & .. & 0 \\ 0 & .. & .. & .. \\ 0 & .. & .. & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & .. & .. & .. \\ 0 & .. & .. & 1 \\ .. & .. & .. & .. \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & .. & .. & .. \\ 0 & .. & .. & 1 \end{pmatrix} \tag{12}$$

What is interesting about these matrices is that given the class charges matrices Q_X and Q_Y the term $[Q_X \times A] \times [B \times Q_Y^T]$ generates a matrix of all the possible products combinations across all the points and classes between datasets X and Y. That way all the balanced repelling and attracting interactions among class contributions can be captured in a single matrix $H_{XY}^{[n_X,n_Y]}$ that given the (11) and (12) can be calculated by:

$$H_{XY}^{[n_X \times n_Y]} = [Q_X \times A] \times [B \times Q_Y^T] - 2Q_X \times Q_Y^T \tag{13}$$

For computational complexity the critical operation is a calculation of distances between all the pairs of data points. Using matrix formulation of the problem and the appropriate mathematical software, this task can be accomplished rapidly even for thousands of training sources. Given our dataset X we want to obtain the matrix of distances $D^{[n_X,n_Y]} : \{d_{i,j}\}$ where $i,j = 1,..,n$. The distance matrix can be then calculated instantly by:

$$D_{XY} = \sqrt{((X \circ X) \times \mathbf{1}^{[m,n_Y]} \times -2 \times Y^T + \mathbf{1}^{[n_X,m]} \times Y^T \circ Y^T)} \tag{14}$$

Let D_{XY}^k stand for the matrix of all individual differences $x_{ik} - y_{jk}$ along k^{th} dimension given by:

$$D_{XY}^k = \mathbf{X_k} \times \mathbf{1}^{[1,n_Y]} - \mathbf{1}^{[n_X,1]} \times \mathbf{Y_k}^T \tag{15}$$

where $\mathbf{X_k}$, $\mathbf{Y_k}$ are the vectors of all the values for the k^{th} dimension.

Denoting further the fixed terms in (10) by scalars $a = (2\pi^{m/2}h^m)^{-1}$ and $b = -(2h^2)^{-1}$ we can now obtain a vector with potential energies of all the data in X caused by the interaction with the data in Y in a single matrix operation:

$$\mathbf{U_X} = -\frac{a}{n_X} H_{XY} \circ e^{bD_{XY}} \times \mathbf{1}^{[n_Y,1]} \tag{16}$$

Similarly the force matrix $F_X = [\mathbf{F_X^1}, .., \mathbf{F_X^m}]$ can be obtained in a single matrix operation for each dimension i.e:

$$\mathbf{F_X^k} = -\frac{a}{2bn} H_{XY} \circ D_{XY}^k \circ e^{bD} \times \mathbf{1}^{[n_Y,1]} \tag{17}$$

The examples of the potential energy and force vectors generated by the labelled data points of the famous Iris dataset are presented in Figures 1 and 2.

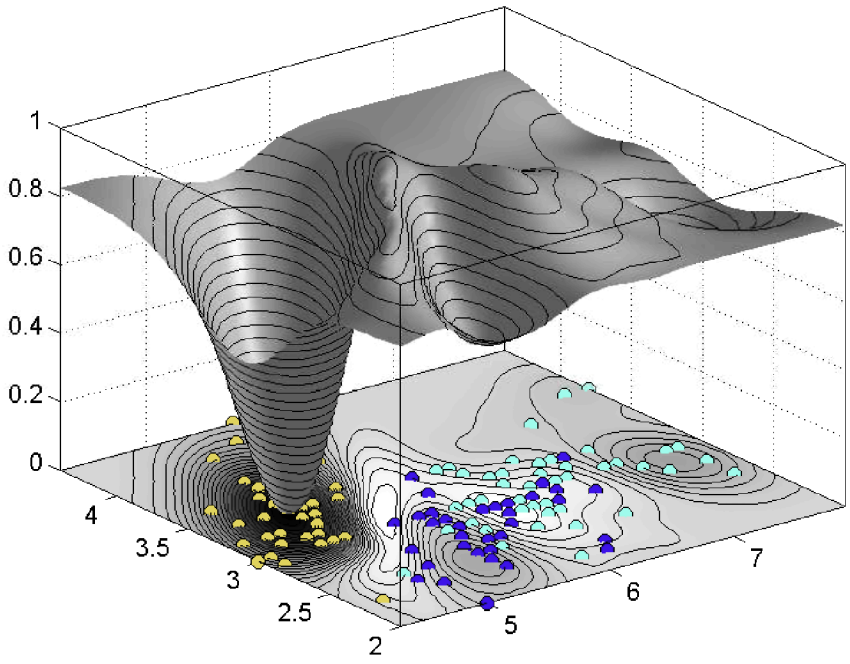

FIGURE 1. Potential energy of the electrostatic interactions among the labelled points in Iris data set

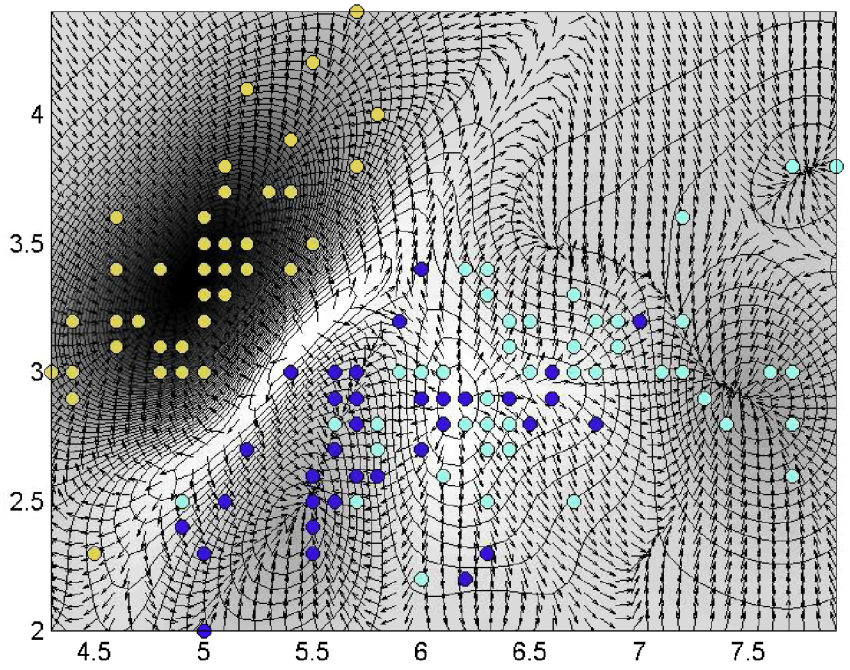

FIGURE 2. Force vectors pointing at directions of the maximum fall of energy of the interactions among the classes in Iris dataset

The forces act upon all the data $\mathbf{X}$ in the input space and cause them to move and merge. Assuming the force vector does not change within proximity of points the distance by which the data move in a small time period Δt is proportional to the force and the inverse of the charge and can be approximated by $F\Delta t^2/Q$. In order to guard against overshooting the time period applied to each step is set such that each data point moves by a distance of at most d i.e.:

$$\Delta t_d = \max_{i=1}^{n} \sqrt{\frac{dq_i}{|\mathbf{f_i}|}} = \max \sqrt{dQ \times \mathbf{1}^{[L,1]} \circ \frac{1}{\sqrt{(F \circ F) \times \mathbf{1}^{[m,1]}}}} \tag{18}$$

Given Δt_d, the matrix of shifts in all dimensions at each step can be simply obtained by:

$$\Delta X = F\Delta t_d^2 \circ \frac{1}{Q_X \times \mathbf{1}^{[L,m]}} \tag{19}$$

After the shifts all the points are tested against mergers i.e. if any pair of points is closer than d from each other then they are merged into a single point with the summed class charges and the merge location of the new point at:

$$\mathbf{x_{merge}}(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_j}\frac{q_j}{q_i + q_j} + \mathbf{x_i}\frac{q_i}{q_i + q_j} \tag{20}$$

Then the whole process repeats and the data continue to move until the equilibrium is found in which the total energy does not fall any more.

Summarising, the condensation process can be formalised in the following steps:

1. Build Parzen density classifier PDE on the the original labelled data (Y, Q_Y) as shown in (3) and (4)
2. Calculate distance matrix D^2 between all the pairs of points between X and Y using (14)
3. Prepare the charge matrix Q_X by taking the class labels or applying soft PDE estimate according to (4)
4. Using D_{XY}, Q_X and Q_Y calculate the class interaction matrix H_{XY} using (13)
5. Using D_{XY} and H_{XY} calculate the force matrix F_X according to (17) and/or energy vector $\mathbf{U_X}$ using (16)
6. Using Q_X and F_X calculate the optimal time period Δt_d as shown in (18)
7. Using F and Δt_d calculate the matrix of shifts ΔX according to (19)and move the points to the new locations
8. Apply the merge test to all the pairs in X and merge the points that were tested positive by calculating the new weighted location according to (20) and summing the corresponding soft class charges.
9. If total energy of interaction measured at X did not decrease then STOP else go to step 2.

Now in the first option of the condensation process one could consider X to be a mobile data-particles that can move and merge as an interaction of the fixed dataset Y that stays unchanged. That way the data in X are continuously guided by the original uncompressed datasets Y. In the other option of the algorithm the datasets X and Y are considered to be the same and the moving points interact with each other rather than with their original "ghost" copies. For both options the guidance of the original class densities can be further strengthened by continuous reevaluation of the charge matrix Q_X such that it is in proportion with the original Parzen density estimates i.e. at step resetting the charge to the original Parzen density enforced values:

$$Q_X = Q_X \times \mathbf{1}^{[L,L]} \circ P \tag{21}$$

where $P^{[n_X,L]}$ is a Parzen density based evaluation of the class posterior probabilities obtained according to (4)

MODEL TUNING AND ONLINE UPDATE

The advantage of the presented data reduction technique is that at any moment as well as at the equilibrium it contains all the necessary state parameters and can be easily updated by the incoming data or even appended by the whole whole new dataset. The only requirement is that the new data have to be consistent in terms of dimensionality and the number of classes with the original dataset. Before we get into details on condensation model update let us summarise the model parameters and their meaning for controlling condensation process. So far there are only two model parameters:

- The Gaussian width h controls the balance between local and global strength of the data density field and is optimised with respect to the generalisation performance of the Parzen density based classifier built on this data

- Data merger distance d is a threshold distance below which a pair of point is merged and is set to be about $1/500$ of the maximum distance between 2 points found in the input space.

In cases of labelled data with well beyond 2 classes there could be a risk of imbalance between attracting and repelling interaction caused by the excessive repulsion from many combinations of differently labelled data. In order to remove this imbalance we can introduce a generic adjustment to the class interaction matrix H which can be parametrised using regularisation parameter s to the general form:

$$H_{XY}^{[n_X,n_Y]}(s) = H_s = [Q_X, A] \times [B \times Q_Y^T] - 2sQ_X \times Q_Y^T \tag{22}$$

Parameter s is defacto a factor expressing how many times the attracting interaction is to be stronger/weaker than the repelling interaction. To free the user from setting this parameter arbitrarily we can introduce a generic rule that sets s such that total energy of the data interaction cancels out to nil. To achieve this we have to solve the following matrix equation:

$$\sum U_X = \mathbf{1}^{[1,n_X]} \times (H_s \circ e^{bD_{XY}}) \times \mathbf{1}^{[n_X,1]} = 0 \tag{23}$$

with respect to the regularisation parameter s. In the model we used bisection method to find numerical estimation of the parameter s. That way we always achieve a balanced field such that the excess of inter-class repulsion is always compensated by the boosted intra-class attraction and vice versa in cases of small number of well separated classes.

After each iteration as well as in the equilibrium state the state of the condensed dataset is fully defined by the matrix of current locations X and the summarised charges matrix Q. Following multiple mergers individual class charges q_{il} of certain points $\mathbf{x_i}$ can be in excess of tens or hundreds. Such points typically sit in the local maxima of class-conditional Parzen density estimates and are resistant to move due to interaction with low charge particles. The class charges serve here two purposes: on the one hand they represent the amount of class represented in data-point units and secondly in the absence of kinetic effects they are the sole sources of particles' inertial momentum that in that case is a tendency to maintain the current position in the input space. These properties allow for easy updates of the condensed data by the new data point or even whole new datasets without the need to rerun the condensation. The update process is achieved by simply releasing the new data in their original locations in the input space and letting them move down the potential until the updated system reaches the new equilibrium state. The same applies to an update by multiple data points or larger new datasets.

The PLDC model can be updated in two different modes: accumulative and adaptive. In the accumulative mode the sum of class charges keeps increasing by the new labelled data. In the adaptive mode, the sum of class charges is kept constant by renormalisation following the new additions, i.e. when the new dataset X_{new} with the class charges Q_{new} is added to the existing partially or fully condensed dataset X_{old} with class charges Q_{old} then the new overall dataset becomes:

$$\begin{cases} X^{[n_{old}+n_{new},m]} = [X_{old}; X_{new}] \\ Q^{[n_{old}+n_{new},L]} = \dfrac{\mathbf{1}^{[1,n_{old}]} \times Q_{old} \times \mathbf{1}^{[L,1]}}{\mathbf{1}^{[1,n_{old}]} \times Q_{old} \times \mathbf{1}^{[L,1]} + \mathbf{1}^{[1,n_{new}]} \times Q_{new} \times \mathbf{1}^{[L,1]}} [Q_{old}; Q_{new}] \end{cases} \tag{24}$$

where operation $[A;B]$ means merging the two matrices A and B together along the first dimension.

EXPERIMENTS

The presented electrostatic condensation model has been evaluated in terms of classification performance obtained once trained on the reduced datasets and compared with the performance obtained for training on the original data. A number of well-known datasets from the UCI Repository [1] have been selected for evaluation. A brief summary of these datasets is presented in Table 1.

All the datasets were compressed using both options of the algorithm and their final states containing compressed data locations and the class charge matrices stored and used further to build Parzen density classifiers. Figures 3 and 4 show the compressed states for Iris and Land Satellite Images datasets overlayed over the original data plots. The sizes of the compressed data points are proportional to the sum of class charges $\sum_j q_{ij}$ accumulated during the mergers with other data points.

[1] University of California Repository of Machine Learning Databases and Domain Theories: ftp.ics.uci.edu/pub/machine-learning-databases

TABLE 1. A list of datasets used in the experiments

Dataset	Size	Features	Classes
Iris	150	4	3
Ionosphere	351	34	2
Diabetes	768	8	2
Satimage	6435	36	6

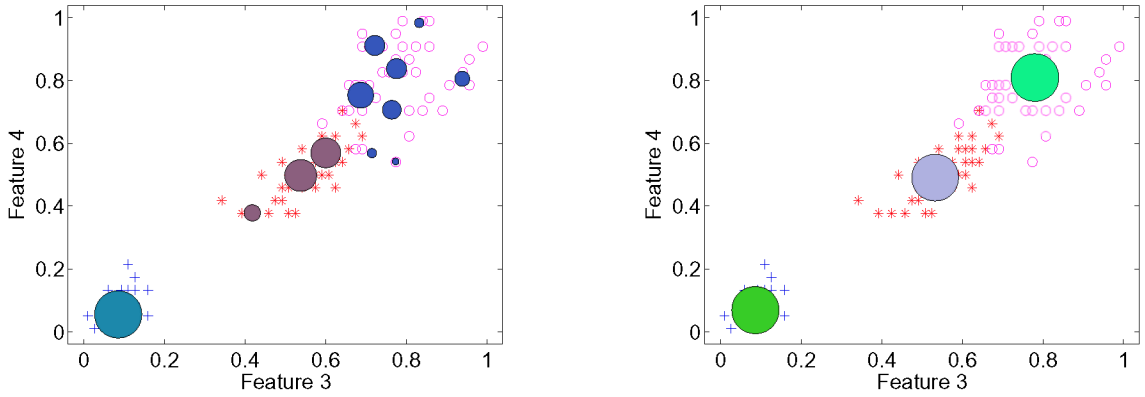

FIGURE 3. Iris dataset compressed using electrostatic interaction (left) with the fixed density guidance (right) with the dynamic class relabelling

The trained models were then applied to classify the corresponding original uncompressed datasets and obtain misclassification errors that were compared to the standard 10-fold cross-validation error obtained without any compression. These errors together with the compression rates obtained for both options of the PLDC model are presented in Table 2.

Please note that dynamic PLDC achieves higher compression levels yet results in worse classification performance than static PLDC which is in agreement with the intuitive trade-off between complexity of the data structure and the accuracy of the model built on this data.

Now using the same criterion of the compression rate and misclassification error we have compared PLDC method with other data reduction techniques. Three other methods were picked for comparison, two rather simple ones based

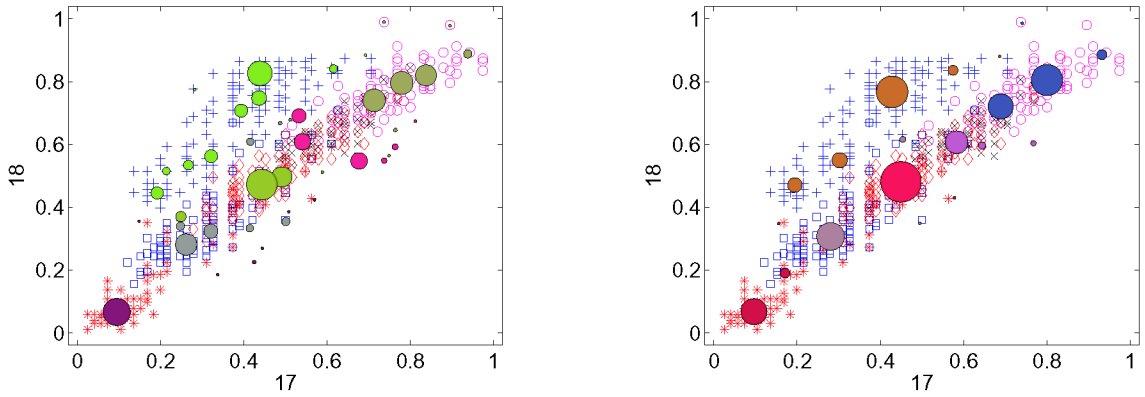

FIGURE 4. Land Satellite dataset compressed using electrostatic interaction (left) with the fixed density guidance (right) with the dynamic class relabelling

TABLE 2. Classification errors and compression levels obtained from application of PLDC to different datasets from UCI Repository

Dataset	10-fold cross-validation error [%]	Fixed PLDC error [%]	Reduction level [%]	Dynamic PLDC error [%]	Reduction level [%]
Iris	4.00	4.3	92.0	5.3	98.0
Ionosphere	13.1	13.3	91.6	14.8	98.1
Diabetes	25.1	25.8	95.3	29.4	98.7
Satimage	13.2	13.3	95.4	14.1	98.9

TABLE 3. Comparison of various data reductions techniques with PLDC for the Land Satellite dataset.

Method	Error [%]	Reduction level [%]
10-fold c-v	13.2	0
Fixed PLDC	13.3	95.4
RSDE	15.7	82.3
Random Sampling	18.1	95.0
K-means clustering	16.8	95.0

on k-means clustering and random sub-sampling and a more advanced condensation method called Reduced Set Density Estimator (RSDE) introduced in [6]. Due to complexity of the problem and a lack of presentation space the comparison has been carried out only for Land Satellite dataset and the results are shown in Table 3.

For the k-means and random sampling methods the level of compression has been fixed to 95% such that it is comparable to the fixed PLDC compression levels. The results clearly indicate that at the same compression rates PLDC allows to obtain higher classification performance close to the original performance of the classifier trained on the complete uncompressed dataset.

CONCLUSIONS

This work introduces a new methodology of labelled data reduction that is directly optimised to a particular classification model. The rationale behind such coupled data reduction technique is to exploit specific modelling engine of a particular classifier in order to strengthen the importance of key data points while weaken or remove points that play a minor role in shaping the class distributions and ultimately class boundaries. The data reduction takes place as a result of dynamic data condensation caused by an electrostatic-type interaction that causes the data to move and merge in a search for stable local energy minima. The interaction among the data has been designed in a class-sensitive yet balanced manner such that the intra-class attraction and inter-class repulsion are normalised to result in a zero total interaction energy. The presented data reduction methodology is generic and applicable to any classifier yet is particularly effective for classifiers capable of delivering soft class probabilities as has been shown in detail for the Parzen density classifier. This particular instance of the condensation model called PLDC has been shown to reduce all the tested datasets down to few percent of their original sizes yet virtually retaining the spatial class distribution in-tact such that the classifier trained on the reduced dataset performs as well as if trained on the uncompressed data. We have shown two ways of constraining the reduction process by the original class densities. In the first method called fixed PLDC the mobile data were interacting with the fixed original data which resulted in a less condensed data though better matching the original class distributions and better subsequent classification performance. The other model version called dynamic PLDC used continuous redistribution of class charges according to the original class distributions and resulted in very high condensation levels often exceeding 98%. Such high condensation have been achieved for the price of only slightly worse classification performance. Future advancements will include attempts to incorporate categorical data and provide automated field tuning mechanisms along with an attempt to make the presented algorithms more scalable and numerically stable.

REFERENCES

1. M. Morgan, *Unearthing the customer: data mining is no longer the preserve of mathematical statisticians. Marketeers can also make a real, practical use of it (Revenue-Generating Networks)*. Telecommunications International, May 2003.
2. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, John Wiley & Sons, New York, 2001.
3. S. M. Weiss, and C. A. Kulikowski, *Computer systems that learn*, Morgan Kaufmann Publishers, San Mateo, 1991.
4. P. Mitra, T. R. Murthy, and S. K. Pal, *Density-based multiscale data condensation*, IEEE Transactions on Pattern Analysis and Machine Intelligence 24(6), 2002, pp 734–747.
5. D. R. Wilson, and T. R. Martinez, *Reduction techniques for instance-basd learning algorithms*, Machine Learning 38(3), 2000, pp 257–286.
6. M. Girolami and C. He, *Probability density estimation from optimally condensed data samples*, IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10), 2003, pp 1253–1264.
7. Y. Leung, J.-S. Zhang, and Z.-B. Xu, *Clustering by scale-space filtering*, IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 2000, pp 1396–1410.
8. M. Plutowski, and H. White, *Selecting concise training sets from clean data*, IEEE Trans. on Neural Networks 4(2), 1993, pp 305–318.
9. J. Shawe-Taylor, and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge University Press, Cambridge, 2004.
10. V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer Verlag, 2006.
11. D. Ruta and B. Gabrys, *A Framework for Machine Learning based on Dynamic Physical Fields*, to appear in the Special Issue of Natural Computing Journal on Nature-inspired Learning and Adaptive Systems, 2007.
12. S. Hochreiter, M. C. Mozer, *An Electric Approach to Independent Component Analysis*, Proceeedings of the 2^{nd} International Workshop on Independent Component Analysis and Signal Separation, Helsinki, 2000, pp 45–50.
13. J. Principe, I. Fisher, D. Xu: Information Theoretic Learning. In S. Haykin (Ed.): Unsupervised Adaptive Filtering. New York NY (2000).
14. K. Torkkola: Nonlinear feature transforms using maximum mutual information. Proc. of IJCNN'2001, Washington DC, USA (2001).