Machine Learning for Biological Trajectory Classification Applications

By Ivo F. Sbalzarini†, Julie Theriot ‡ AND Petros Koumoutsakos ¶

Machine learning techniques including clustering algorithms, support vector machines and hidden Markov models are applied to the task of classifying trajectories of moving keratocyte cells. The different algorithms are compared among each other as well as to expert and non-expert test persons using concepts from signal detection theory. The algorithms performed very well as compared to humans, suggesting a robust tool for trajectory classification in biological applications.

1. Motivation and Objectives

Empirical sciences create new knowledge by inductive learning from experimental observations (data). Biology, or life science in general, is a prime example for a field that is facing a rapidly growing amount of data from continuously more sophisticated and efficient experimental assays. The general lack of predictive models makes quantitative evaluation and learning from the data one of the core processes in the creation of new knowledge. Trajectories of moving cells, viruses or whole organisms are a particularly interesting example as they represent dynamic processes. The application of machine learning techniques for automatic classification of data mainly serves 3 goals: First, one wishes to learn more about the biological or biochemical processes behind the observed phenomenon by identifying the parameters in the observation that are significantly influenced by a certain change in experimental conditions (*causality detection*). Second, the information contents of a given data set with respect to a certain property of interest may be estimated (*capacity estimation*) and third, automatic identification and classification of vast amounts of experimental data (*data mining*) could facilitate the process of interpretation. The paper starts by formally stating the problem of classification and introducing the notation. Then, different machine learning techniques are summarized starting from clustering methods in the d-dimensional real space $\mathbb{R}^d$ and proceeding to risk-optimal separation in $\mathbb{R}^d$ and dynamic signal source models in $\mathbb{R}^d \times \mathbb{T}$. Finally, the results of two automatic classification experiments of keratocyte cell trajectories are presented and compared to the performance of human test subjects on the same task.

2. The classification problem

Classification is one of the fundamental problems in machine learning theory. Suppose we are given n classes of objects. When we are faced with a new, previously unseen object, we have to assign it to one of the classes. The problem can be formalized as follows: we are given m empirical data points

† Institute of Computational Science, ETH Zürich, 8092 Zürich, Switzerland
‡ Department of Biochemistry, Stanford University, Stanford, U.S.A.
¶ Institute of Computational Science, ETH Zürich and CTR/NASA Ames

$$(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathcal{Y} \tag{2.1}$$

where $\mathcal{X}$ is a non-empty set from which the *observations* (sometimes called *patterns*) are taken and in the present context $\mathcal{Y} = \{1, \ldots, n\}$. The $y_i \in \mathcal{Y}$ are are called *labels* and contain information about which class a particular pattern belongs to. Classification means *generalization* to unseen data points (x, y), i.e. we want to predict the $y \in \mathcal{Y}$ given some new observation $x \in \mathcal{X}$. Formally, this amounts to the *estimation* of a function $f : \mathcal{X} \mapsto \mathcal{Y}$ using the input-output *training data* (2.1), generated independent and identically distributed (i.i.d.) according to an unknown probability distribution $P(x, y)$, such that f will optimally classify unseen patterns $x \in \mathcal{X}$. The criterion of optimality is to minimize the *expected risk*

$$R[f] = \int_{\mathcal{X} \times \mathcal{Y}} l\left(f(x), y\right) \, dP(x, y) \tag{2.2}$$

where l denotes a suitably chosen *cost function*. A common choice is the *0/1-loss*, for which $l(f(x), y)$ is 0 if (x, y) is a correct classification and 1 otherwise. Unfortunately, the expected risk cannot be minimized directly, since the underlying probability distribution $P(x, y)$ is unknown. Therefore, machine learning algorithms try to *approximate* $R[f]$ based on the available information from the training data. The most common approximation is the *empirical risk*

$$R_{emp}[f] = \frac{1}{m} \sum_{i=1}^{m} l\left(f(x_i), y_i\right) \tag{2.3}$$

Different classifiers use different approximations to (2.2) as well as different methods to minimize those approximations.

3. Machine learning methods used

3.1. *k–nearest neighbors (KNN)*

One of the simplest classifiers if $\mathcal{X} = \mathbb{R}^d$ is the k–nearest neighbor (KNN) algorithm. A previously unseen pattern x is simply assigned to the same class $y \in \mathcal{Y}$ to which the majority of its k (to be chosen) nearest neighbors belongs. The algorithm can be seen as a very simple form of a self-organizing map (Kohonen (2001)) with fixed connections.

3.2. *Gaussian mixtures with expectation maximization (GMM)*

Gaussian mixture models (GMM) are more sophisticated clustering algorithms in $\mathcal{X} = \mathbb{R}^d$. They make use of Gaussian probability distributions on $\mathbb{R}^d$ and try to approximate the unknown distribution $P(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ by a mixture of n Gaussians $\mathcal{N}_i(x, y, \mu_i, \Sigma_i)$ with means $\mu_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ and covariance matrices $\Sigma_i \in \mathbb{R}^{d \times d}$, $i = 1, \ldots, n$. The parameters μ_i and Σ_i are chosen so as to maximize the *log-likelihood* that the given training data has actually been drawn i.i.d. from the probability distribution $P(x, y) = \sum_{i=1}^{n} \mathcal{N}_i(x, y, \mu_i, \Sigma_i)$. The algorithm can be written as follows:

Step 1: Choose a set of initial means $\mu_1, \ldots, \mu_n$ using the *k–means* clustering algorithm (Hartigan & Wong (1979)); all covariances are initialized to identity: $\Sigma_i = I_d$.

Step 2: Assign the m training samples to the n clusters Γ_i using the *minimum Mahalanobis distance rule*: Sample x belongs to cluster Γ_i if the corresponding log-likelihood measure becomes minimum, i.e. $i = \arg\min_i \left[\log\left(\det\left(\Sigma_i\right)\right) + \left(x-\mu_i\right)^\top \left(\Sigma_i\right)^{-1}\left(x-\mu_i\right)\right]$.

Step 3: Compute new means $\mu_i \leftarrow \sum_{x\in\Gamma_i} x/\#\{\Gamma_i\}$ and new covariance estimates $\Sigma_i \leftarrow \sum_{x\in\Gamma_i}\left(x-\mu_i\right)\left(x-\mu_i\right)^\top/\#\{\Gamma_i\}$ where $\#\{\Gamma_i\}$ denotes the number of vectors x assigned to cluster Γ_i in step 2.

Step 4: If the changes in the means and covariances are smaller than a certain tolerance, stop, otherwise go to step 2.

3.3. *Support Vector Machines (SVM)*

Support vector machines (SVM) are kernel-based classifiers (Müller *et al.* (2001)) for binary classification in $\mathcal{X} = \mathbb{R}^d$. Past applications included time series prediction (Mukherjee *et al.* (1997)), gene expression analysis (Brown *et al.* (2000)) as well as DNA and protein analysis (Zien *et al.* (2000)). SVM make use of the following theorem of *statistical learning theory* by Vapnik (1998) that gives an upper bound for the expected risk:

Theorem 1: Let h denote the Vapnik-Chervonenkis (VC) dimension of the function class $\mathcal{F}$ and let $R_{emp}[f]$ be the empirical risk for the 0/1-loss of a given classifier function $f \in \mathcal{F}$. It holds with probability of at least $1-\delta$, that:

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{h\left(\log\frac{2m}{h}+1\right) - \log\left(\frac{\delta}{4}\right)}{m}} \tag{3.1}$$

for all $\delta > 0$, for $f \in \mathcal{F}$ and $m > h$.

The VC dimension h of a function class $\mathcal{F}$ measures how man points $x \in \mathcal{X}$ can be separated in all possible ways using only functions of the class $\mathcal{F}$. Kernel methods use a mapping $\Phi(x)$ of the training data x into a higher-dimensional *feature space* $\mathcal{H}$ in which it can be separated by a hyper-plane $f(\mathbf{x}) = (\mathbf{w}\cdot\Phi(\mathbf{x})) + b$. In $\mathcal{H}$, the optimal separating hyper-plane is determined such that the points $\Phi(x)$ closest to it (called the *support vectors*) have maximum distance from it, i.e. such that the "safety margin" is maximized. This is done by solving the quadratic optimization problem $(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||_2^2$ subject to the condition that $\mathbf{w}\cdot\Phi(\mathbf{x}) + b$ is a separating hyper-plane. Solving the dual optimization problem, the Lagrange multipliers α_i, $i = 1,\ldots,s$ are obtained, where s is the number of support vectors. The classifier function f in $\mathcal{H}$ is then given by:

$$f(x) = \frac{3}{2} + \frac{1}{2}\cdot sign\left(\sum_{i=1}^{s} y_i\alpha_i\left(\Phi(x)\cdot\Phi(x_i)\right) + b\right)$$

Since this only depends on the scalar product of the data in feature space, the mapping Φ does not need to be explicitly known. Instead, a *kernel function* $k(x, x_i)$ is introduced such that $k(x, x_i) = \Phi(x)\cdot\Phi(x_i)$. The support vector classifier $f : \mathcal{X} \mapsto \{1, 2\}$ to be evaluated for any new observation thus is:

$$f(x) = \frac{3}{2} + \frac{1}{2}\cdot sign\left(\sum_{i=1}^{s} y_i\alpha_i k(x, x_i) + b\right) \tag{3.2}$$

Notice that the sum only runs over all support vectors. Since generally $s \ll m$, this allows efficient classification of a new observation by comparing it to a small relevant subset of the training data.

3.4. *Hidden Markov Models (HMM)*

Hidden Markov models (HMM) are stochastic signal source models, i.e. they do not require observations $x \in \mathbb{R}^d$ but can treat discrete dynamic time series $x = \{O_1, \ldots, O_T\} \in \mathcal{X}$, $O_i \in \mathbb{R}$. In the past, their most successful application was in speech recognition (Rabiner (1989)). An HMM attempts to model the source producing the signal x as a dynamic system which can be described at any time t as being in one of r distinct discrete states, $Q_1, \ldots, Q_r$ that are hidden, i.e. cannot be observed. At regularly spaced discrete times $t_i = i\delta t$, $i = 1, \ldots, T$, the system changes its internal state, possibly back to the same state. The process is assumed to be Markovian, i.e. its probabilistic description is completely determined by the present and the predecessor state. Let q_i denote the actual state of the system at time t_i. The Markov property thus states that $P[q_i = Q_j | q_{i-1} = Q_k, q_{i-2} = Q_l, \ldots] = P[q_i = Q_j | q_{i-1} = Q_k]$ where $P[E|F]$ denotes the probability of an event E given that F occurred. The state transitions are described by probabilities $a_{jk} = P[q_i = Q_k | q_{i-1} = Q_j]$ forming the elements of the state transition matrix A and obeying the constraints $a_{jk} \geq 0 \ \forall \ j,k$ and $\sum_{k=1}^{r} a_{jk} = 1$. At each time point t_i the system produces an observable output O_i, drawn from the output probability distribution $b_{Q_i}(O)$ associated with state Q_i; $B = \{b_{Q_j}\}_{j=1}^{r}$. The model is completed with the initial state probabilities $\pi = \{\pi_j = P[q_1 = Q_j]\}_{j=1}^{r}$ and the complete HMM is denoted by $\lambda = (A, B, \pi)$.

Given the form of HMM described above, there are three basic problems of interest that must be solved (Rabiner (1989)):

(1) Given an observation $x = \{O_1, \ldots, O_T\}$ and a model $\lambda = (A, B, \pi)$, compute the probability $P[x|\lambda]$ that the observation x has been produced by a source described by λ.

(2) Given an output sequence $x = \{O_1, \ldots, O_T\}$ and a model $\lambda = (A, B, \pi)$, determine the most probable internal state sequence $\{q_1, \ldots, q_T\}$ of the model λ that produced x.

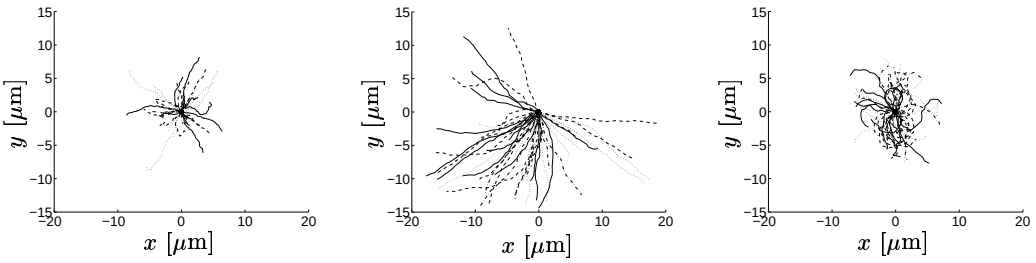
(3) Determine the model parameters $\lambda = (A, B, \pi)$ to maximize $P[x|\lambda]$ for a given observation x.

3.4.1. *Discrete hidden Markov models (dHMM)*

If the set of possible distinct values $\{v_k\}$ of any output O_i is finite, the HMM is called *discrete* (dHMM). The output probability distribution of any state Q_j is thus discrete: $b_{Q_j} = \{b_{Q_j}(k) = P[O_i = v_k | q_i = Q_j]\}$ for $k = 1, \ldots, M$.
Direct solution of problem (1) would involve a sum over all possible state sequences: $P[x|\lambda] = \sum_{\forall \{q_1, \ldots, q_T\}} P[x|\{q_1, \ldots, q_T\}, \lambda] P[\{q_1, \ldots, q_T\}|\lambda]$. The computational cost of this evaluation is $\mathcal{O}(2Tr^T)$, which is about 10^{50} for an average dHMM and thus clearly unfeasible. The *forward backward algorithm* as stated by Baum & Egon (1967), Baum & Sell (1968) solves this problem efficiently in $\mathcal{O}(r^2 T)$. The solution of problem (2) is given by the *Viterbi algorithm* (Viterbi (1967), Forney (1973)) and the "training problem" (3) is solved using the iterative *Baum-Welch expectation maximization method* (Dempster *et al.* (1977)).

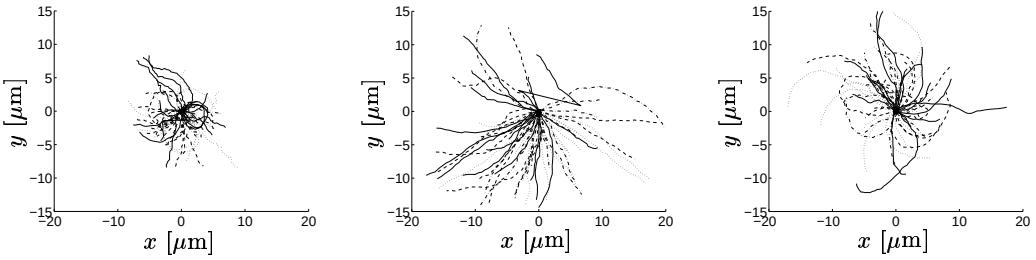
3.4.2. *Continuous hidden Markov models (cHMM)*

If the observations O_i are drawn from a continuum, b_{Q_i} is a continuous probability density function and the HMM is called *continuous* (cHMM). The most general case for which the three above problems have been solved is a finite mixture of M Gaussians $\mathcal{N}_k$, thus $b_{Q_j}(O) = \sum_{k=1}^{M} c_{jk} \mathcal{N}_k(O, \mu_{jk}, \Sigma_{jk})$ (see Liporace (1982), Juang *et al.* (1985)).



10°C observation 20°C observation 30°C observation

FIGURE 1. Temperature data set. 46 trajectories of moving keratocytes were used per class. The 3 classes are defined by the 3 temperatures at which the observations were made. All trajectories are shifted such that they start at the origin of the coordinate system.



10°C acclimation 16°C acclimation 25°C acclimation

FIGURE 2. Acclimation data set. 58 trajectories of moving keratocytes were used per class. The 3 classes are defined by the 3 temperatures at which the fish were acclimated for 3 weeks prior to the experiment. All trajectories are shifted such that they start at the origin of the coordinate system.

4. Application to trajectory classification

4.1. *The data*

All machine learning classifiers described in the previous section were applied to the task of classifying trajectories of living cells. The cells were keratocytes taken from the scales of the fish *Gillichthys mirabilis* (commonly called longjawed mudsucker). Isolated cells cultured on glass coverslips were observed using an inverted phase-contrast microscope connected to a video camera. The 2D trajectories of the moving cells were then extracted from the movies using the semi-automatic tracking software Metamorph (Universal Imaging, Inc.) yielding position readings at equidistant sampling intervals of $\delta t = 15\,\mathrm{s}$. The observations x in the present case were position/time data sets, thus $\mathcal{X} = \mathbb{R}^2 \times \mathbb{T}$ where $\mathbb{T}$ denotes the discrete ordered time space. Two different experiments were performed: For the *temperature data set*, fish were acclimated at 16°C, i.e. they were kept in water of this temperature for at least 3 weeks† prior to cell isolation. The movement of the isolated cells was then recorded at 10°C, 20°C and 30°C using a temperature-controlled microscope stage. 167 trajectories (46 at 10°C, 63 at 20°C and 58 at 30°C) from 60 different cells were collected. To make all classes the same size, 46 trajectories were used from each making a total of $N = 138$. Figure 1 shows them for the 3 temperature classes. For the *acclimation data set* all cells were observed at 20°C but they were taken from three different fish populations acclimated at 10°C, 16°C and 25°C, respectively. From

† After this time the adaptive changes in liver lipid content are complete.

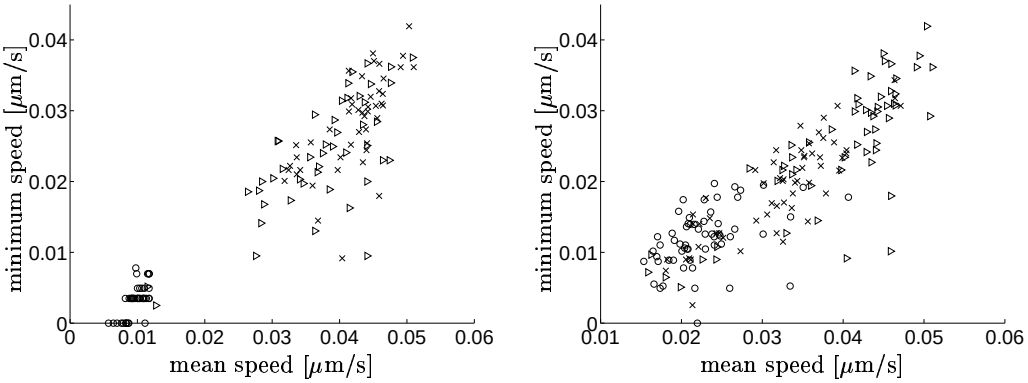


FIGURE 3. Encoded data sets. Both the temperature data set (left) and the acclimation data set (right) were encoded using the average and the minimum of the speed along a trajectory. Data points from the 10°C temperature and 10°C acclimation classes are denoted by circles (∘), those from the 20°C temperature and 16°C acclimation classes by triangles (▷) and those from the 30°C temperature and 25°C acclimation classes by crosses (×).

the recorded 184 trajectories (58 for 10°C, 63 for 16°C, 63 for 25°C) of 60 different cells a total of $N = 174$ (58 in each class) was used as shown in figure 2. Both data sets have $n = 3$ classes.

4.2. *Data preprocessing and encoding*

Since the trajectories $x \in \mathbb{R}^2 \times \mathbb{T}$ are not vectors in $\mathbb{R}^d$, encoding is necessary for all machine learning algorithms considered. HMM are capable of handling dynamic data in $\mathbb{R} \times \mathbb{T}$ and thus need the least encoding. Since the reaction rates of many biochemical processes that contribute to cell movement depend on temperature, the latter is suspected to influence the speed of the movement. The encoding for the cHMM was thus chosen to be the momentary speed of the movement along the trajectory. For dHMM the speed was discretized into 4 equidistant bins. One HMM was trained for each of the 3 classes. After evaluating the probability $P[x|\lambda_i]$ of a new observation x against the models λ_i for all classes $i = 1, 2, 3$, x is assigned to the class which has the highest probability. For all other algorithms, a quasi-static representation in $\mathbb{R}^d$ has to be found. The following properties were calculated for all trajectories: average speed, standard deviation of speed, mean angle of direction change between 2 subsequent measurement points, standard deviation of those angles, distance between first and last point of trajectory compared to its total path length, decay of autocorrelation functions of speed and direction change angle, minimum and maximum occurring speed and angle change. Histograms of the distribution of these properties among the different classes of trajectories gave evidence about which are the most discriminating properties. For the following considerations, the mean and the minimum of the speed of a trajectory over time were taken as encoding properties. The trajectories were thus represented as vectors in $\mathbb{R}^2$. Figure 3 shows the encoded data sets for both the temperature and the acclimation cases. It can be seen that the clusters mostly overlap, making the data non-separable in this encoding space.

4.3. *Classification and evaluation of the results*

The different classification algorithms were trained on a subset of $m = N/2$ data points from each set and then tested on the remainder of the data. For the KNN we set $k = 5$

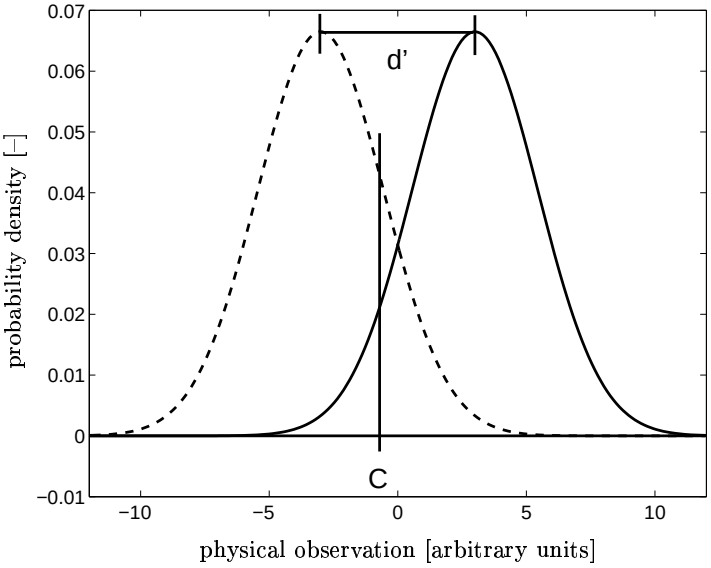


FIGURE 4. Schematic of the theory of signal detection. Observations that belong to a class i of interest occur at a certain probability (———) and observations that do not belong to that class occur at a different probability (– – – –). The classifier chooses a threshold C and will assign all future observations above C to the class of interest. The discrimination capability of the classifier is given by the normalized distance measure d'.

and for the SVM a Gaussian kernel with standard deviation $\sigma = 0.05$ was used. To reduce the random influence of which particular data points are taken for training and which for testing, the whole procedure was repeated 4 times for different partitioning of the data into training and test sets. Let $\mathcal{D} = \{(x_j, y_j),\ j = 1, \ldots, N\}$ be the complete data set of all N recorded trajectories x_j with corresponding class labels y_j, a random $\mathcal{T} \subset \mathcal{D}$ with $\#\{\mathcal{T}\} = m$ the *training set* and $\mathcal{E} \subset \mathcal{D}$ with $\#\{\mathcal{E}\} = N - m$ and $\mathcal{E} \cap \mathcal{T} = \emptyset$ the *test set*. An algorithm, trained on $\mathcal{T}$, classifies the trajectories $x_j \in \mathcal{E}$ without knowing the correct y_j. The outcome of this classification is $\widetilde{y}_j$. The *hit rate* for class i is then defined as $h_i = \#\{x_j \in \mathcal{E} : \widetilde{y}_j = y_j = i\}/\#\{x_j \in \mathcal{E} : y_j = i\}$ where $\#\{\mathcal{A}\}$ denotes the number of elements in a set $\mathcal{A}$. The *false alarm rate* (sometimes also called "false positives") for class i is given by $f_i = \#\{x_j \in \mathcal{E} : \widetilde{y}_j = i \wedge y_j \neq i\}/\#\{x_j \in \mathcal{E} : y_j \neq i\}$. The complementary quantities $m_i = 1 - h_i$ and $r_i = 1 - f_i$ are termed *miss rate* and *correct rejection rate*, respectively. In each classification experiment, both the hit rate and the false alarm rate were recorded for each temperature class since they compose the minimal sufficient set of properties.

Using the *theory of signal detection* (Green & Sweets (1966)), which was originally developed in psychophysics and is widely used in today's experimental psychology, two characteristic parameters were calculated from h_i and f_i. Figure 4 depicts the basic idea: the occurrence of observations that belong to class i and such that do not belong to class i is assumed to be governed by 2 different Gaussian probability distributions. During training, the classifier determines a threshold C above which it will assign all future observations to class i. If, after transformation to standard normal distributions, $C = 0$, the classifier is said to be "neutral", for $C < 0$ it is called "progressive" and for $C > 0$ "conservative". The discrimination capability of the classifier is given by the separation distance d' of the two normalized (by their standard deviation) distributions. $d' = 0$

class	hit [%]	f.a. [%]	d'	C
10°C	100.0	2.2	∞	–
20°C	54.4	24.5	0.8	0.29
30°C	46.7	22.9	0.7	0.41

TABLE 1. KNN on temperature data

class	hit [%]	f.a. [%]	d'	C
10°C	100.0	2.2	∞	–
20°C	54.3	15.7	1.1	0.46
30°C	68.5	20.7	1.3	0.15

TABLE 2. GMM on temperature data

class	hit [%]	f.a. [%]	d'	C
10°C	100.0	2.2	∞	–
20°C	51.1	27.2	0.6	0.29
30°C	41.3	24.4	0.5	0.46

TABLE 3. SVM on temperature data

class	hit [%]	f.a. [%]	d'	C
10°C	100.0	3.3	∞	–
20°C	77.2	28.3	1.3	-0.09
30°C	37.0	11.4	0.9	0.77

TABLE 4. dHMM on temperature data

class	hit [%]	f.a. [%]	d'	C
10°C	100.0	2.2	∞	–
20°C	76.1	30.5	1.2	-0.10
30°C	34.8	11.9	0.8	0.79

TABLE 5. cHMM on temperature data

corresponds to "pure random guessing" where hits and false alarms grow at equal rates and $d' = \infty$ characterizes a "perfect classifier". Since the hit rate is given by the area under the solid curve above C and the false alarm rate is the area under the dashed curve above C, both C and d' can be calculated from h_i and f_i and the latter two completely describe the situation. Classifiers were compared based on d' since algorithms that are capable of better separating the two probability distributions will have a lower expected risk R.

5. Results

5.1. *Temperature data set*

The temperature data set as introduced in section 4.1 was classified using all the algorithms presented in section 3 and the results were evaluated according to section 4.3. Tables 1 to 5 state the average percentage of hits and false alarms (over all different partitioning of the data into training and test sets) as well as the normalized discrimination capabilities d' and thresholds C of the classifiers for each temperature class.

Figure 5 graphically displays the hit and false alarm rates of the classifiers for the 3 temperature classes. The averages over all data partitionings are depicted by solid bars,

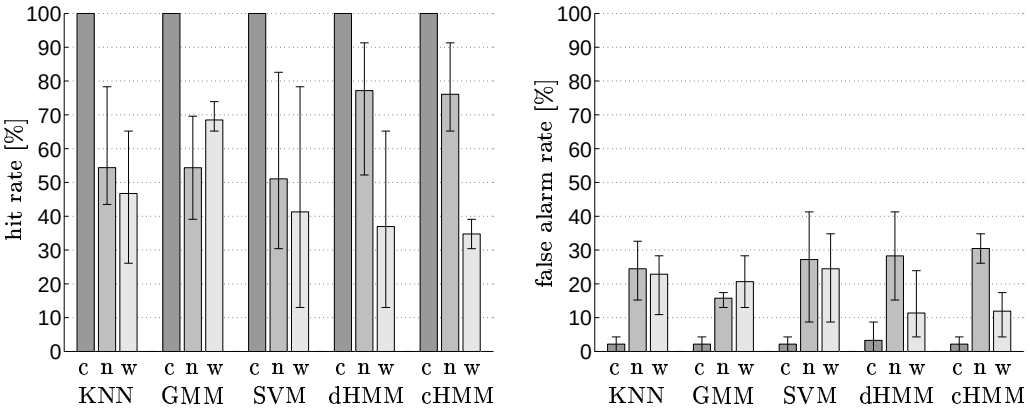


FIGURE 5. Hit and false alarm rates for all classifiers. The percentage of hits (left) and false alarms (right) on the temperature data set is shown for each classifier in each of the 3 temperature classes: 10°C ("c"), 20°C ("n") and 30°C ("w"). The error bars range from the smallest observed rate to the largest one (min–max bars).

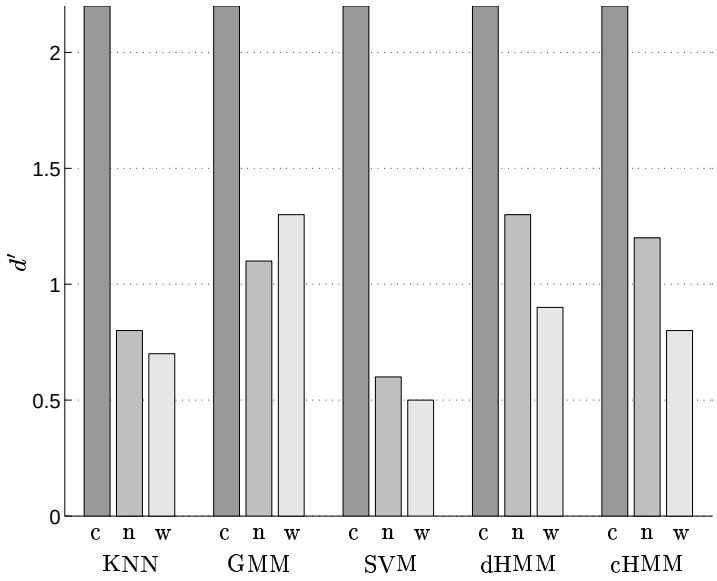


FIGURE 6. d' values for all classifiers. The value of the discrimination sensitivity on the temperature data set is shown for each classifier in each of the 3 temperature classes: 10°C ("c"), 20°C ("n") and 30°C ("w").

the error bars indicate the minima and maxima in the measurements. The d' values of the different classification methods are compared in figure 6.

5.2. *Acclimation data set*

The same classification experiments were also performed using the acclimation data set as introduced in section 4.1. The results are summarized in tables 6 to 10 using the same format as in the previous section. Figure 7 shows the average hit and false alarm rates of the classifiers for the 3 temperature classes along with the min–max bars. Again the classifiers are compared against each other in figure 8 based on their d'.

class	hit [%]	f.a. [%]	d'	C
10°C	77.6	23.3	1.5	-0.02
16°C	59.5	15.5	1.3	0.39
25°C	41.4	22.0	0.6	0.50

TABLE 6. KNN on acclimation data

class	hit [%]	f.a. [%]	d'	C
10°C	88.0	21.1	2.0	-0.19
16°C	58.6	4.3	1.9	0.75
25°C	61.2	20.68	1.1	0.27

TABLE 7. GMM on acclimation data

class	hit [%]	f.a. [%]	d'	C
10°C	86.2	20.3	1.9	-0.13
16°C	62.9	9.9	1.6	0.48
25°C	54.3	18.1	1.0	0.40

TABLE 8. SVM on acclimation data

class	hit [%]	f.a. [%]	d'	C
10°C	84.5	20.3	1.8	-0.09
16°C	71.6	22.4	1.3	0.09
25°C	35.3	11.6	0.8	0.79

TABLE 9. dHMM on acclimation data

class	hit [%]	f.a. [%]	d'	C
10°C	75.0	19.4	1.5	0.09
16°C	56.0	6.9	1.6	0.67
25°C	61.9	27.2	0.9	0.15

TABLE 10. cHMM on acclimation data

class	hit [%]	f.a. [%]	d'	C
10°C	88.5	24.8	1.9	-0.26
16°C	47.3	16.5	0.9	0.52
25°C	33.8	23.8	0.3	0.57

TABLE 11. Humans on acclimation data

In addition to machine learning algorithms, the acclimation data set was also classified by humans. After training on a set of 30 trajectories and their labels, the test subjects were presented one unknown trajectory at a time. Individual position measurement points were symbolized by circles along the trajectory. Since they are equidistant in time, this includes information about the speed. All trajectories were shifted such as to start at $(0, 0)$ and they were rotated by a random angle prior to presentation. Each person classified 174 trajectories appearing in random order. The average result over 5 test subjects is given in table 11. The best performing person who declared after the experiment to have looked at speed information only reached $d' = 2.0$ for the 10°C class, $d' = 1.6$ for the 16°C class and $d' = 0.7$ for the 25°C class. The best person of all reached $d' = 2.1$, $d' = 1.9$ and $d' = 1.0$, respectively by taking into account both speed and shape (curvature) information. The lowest result of the test group was $d' = 1.9$, $d' = 0.1$, $d' = -0.6$.

6. Conclusions and future work

Considering the results of section 5, the following conclusions can be made: (1) All methods perform equally well in the case of separable clusters (10°C temperature class). (2) On the acclimation data set, GMM perform best, closely followed by SVM. This is evidence that the data is actually normally distributed. (3) All classifiers have relatively low values of C, thus being more or less neutral. (4) HMM are the least robust method due

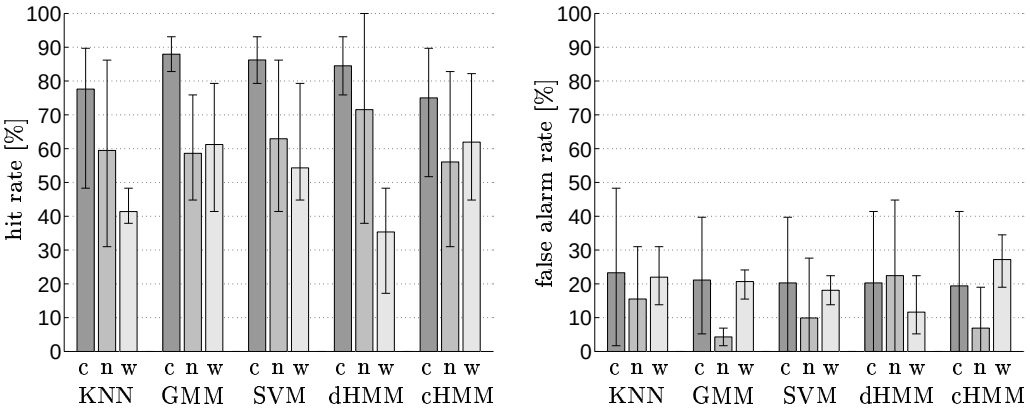


FIGURE 7. Hit and false alarm rates for all classifiers. The percentage of hits (left) and false alarms (right) on the acclimation data set is shown for each classifier in each of the 3 temperature classes: 10°C ("c"), 16°C ("n") and 25°C ("w"). The error bars range from the smallest observed rate to the largest one (min–max bars).

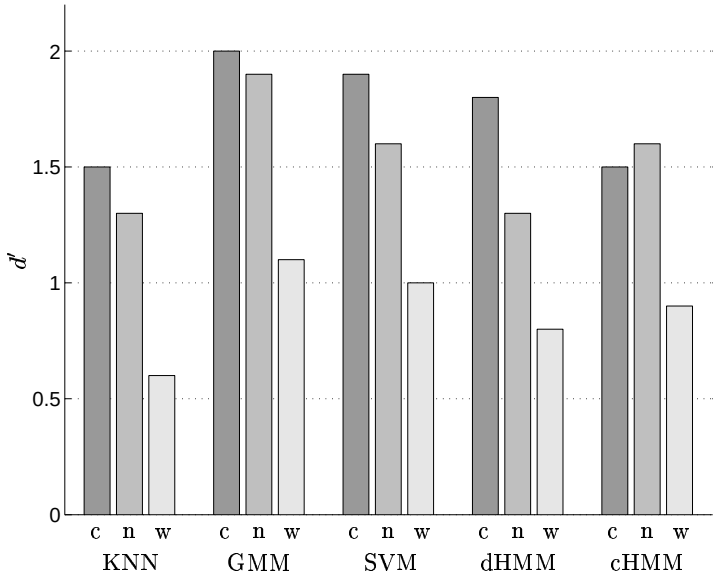


FIGURE 8. d' values for all classifiers. The value of the discrimination sensitivity on the acclimation data set is shown for each classifier in each of the 3 temperature classes: 10°C ("c"), 16°C ("n") and 25°C ("w").

to their dynamic character. The dHMM and the cHMM have comparable performance. (5) Humans on average perform less well than the algorithms. This could be due to bias based on prior information or expectations, fatigue effects or inaccuracy. (6) The best test person performs about equally well as the best machine learning algorithm, indicating that the latter was able to extract and use all the information contained in the data set. In summary, it has been demonstrated that automatic classification of biological trajectories is possible with near-maximum accuracy and that machine learning techniques can be a useful tool in estimating the information content and the relevant parameters in a data set. Future work will be concerned with implementing a general purpose framework code

for classification of dynamic data sets in multiple dimensions. A modular approach will allow different classification algorithms to be used and a preprocessor is envisaged that automatically detects those properties that best cluster (separate) the data at hand. Future applications will include the analysis of *Listeria* and *Shigella* movement inside host cells as well as virus movement and automatic virus detection and identification systems.

REFERENCES

Baum, L. E. & Egon, J. A. 1967 An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. *Bull. Amer. Meteorol. Soc.* **73**, 360–363.

Baum, L. E. & Sell, G. R. 1968 Growth functions for transformations on manifolds. *Pac. J. Math.* **27** (2), 211–227.

Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Furey, T. S., Ares, M. & Haussler, D. 2000 Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl. Acad. Sci. USA* **97** (1), 262–267.

Dempster, A. P., Laird, N. M. & Rubin, D. B. 1977 Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.* **39** (1), 1–38.

Forney, G. D. 1973 The Viterbi algorithm. *Proc. IEEE* **61**, 268–278.

Green, D. M. & Sweets, J. A. 1966 *Signal detection theory and psychophysics*. New York: Krieger.

Hartigan, J. & Wong, M. 1979 A k-means clustering algorithm. *Appl. Stat.* **28**, 100–108.

Juang, B. H., Levinson, S. E. & Sondhi, M. M. 1985 Maximum likelihood estimation for multivariate mixture observations of Markov chains. *IEEE Trans. Informat. Theory* **IT–32** (2), 307–309.

Kohonen, T. 2001 *Self-Organizing Maps*, 3rd edn. Springer-Verlag.

Liporace, L. A. 1982 Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Trans. Informat. Theory* **IT–28** (5), 729–734.

Mukherjee, S., Osuna, E. & Girosi, F. 1997 Nonlinear prediction of chaotic time series using a support vector machine. In *Neural Networks for Signal Processing VII – Proceedings of the 1997 IEEE Workshop* (ed. J. Principe, L. Gile, N. Morgan & E. Wilson), pp. 511–520. IEEE.

Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K. & Schölkopf, B. 2001 An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks* **12** (2), 181–202.

Rabiner, L. R. 1989 A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77** (2), 257–286.

Vapnik, V. N. 1998 *Statistical Learning Theory*. New York: John Wiley & Sons.

Viterbi, A. J. 1967 Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Informat. Theory* **IT–13**, 260–269.

Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T. & Müller, K.-R. 2000 Engineering support vector machine kernels that recognize translation initiation sites in DNA. *Bioinformatics* **16**, 799–807.