Classification of Uncertain Data using Gaussian Process Model

G.V.SURESH
CSE Department, Universal college of engineering technology,
Guntur,A.P,India
vijaysuresh.g@gmail.com

E.V.Reddy
Assoc. Professor, CSE Department
, Universal College of Engineering Technology,
Guntur,A.P,India
evr_universal@yahoo.com

Shabbeer Shaik
Assoc. Professor, MCA Department
Tirmula College of Engineering
Guntur, India
skshabbeer@yahoo.com

Abstract

Data uncertainty is common in real-world applications due to various causes, including imprecise measurement, network latency, out-dated sources and sampling errors. These kinds of uncertainty have to be handled cautiously, or else the mining results could be unreliable or even wrong. We propose that when data mining is performed on uncertain data, data uncertainty has to be considered in order to obtain high quality data mining results. In this paper we study how uncertainty can be incorporated in data mining by using data clustering as a motivating example. We also present a Gaussian process model that can be able to handle data uncertainty in data mining.

.*Keywords*: Gaussian process, uncertain data, Gaussian distribution, Data Mining

.

1. Introduction

Data is often associated with uncertainty because of measurement inaccuracy, sampling discrepancy, outdated data sources, or other errors. This is especially true for applications that require interaction with the physical world, such as location-based services [1] and sensor monitoring [3]. For example, in the scenario of moving objects (such as vehicles or people), it is impossible for the database to track the exact locations of all objects at all-time instants. Therefore, the location of each object is associated with uncertainty between updates [4]. These various sources of uncertainty have to be considered in order to produce accurate query and mining results. We note that with uncertainty, data values are no longer atomic. To apply traditional data mining techniques, uncertain data has to be summarized into atomic values. Taking moving-object applications as an example again, the location of an object can be summarized either by its last recorded location or by an expected location. Unfortunately, discrepancy in the summarized recorded value and the actual values could seriously affect the quality of the mining results. In recent years, there is significant research interest in data uncertainty management. Data uncertainty can be categorized into two types, namely existential uncertainty and value uncertainty. In the first type it is uncertain whether the object or data tuple exists or not. For example, a tuple in a relational database could be associated with a probability value that indicates the confidence of its presence. In value uncertainty, a data item is modelled as a closed region which bounds its possible values, together with a probability density function of its value. This model can be used to quantify the imprecision of location and sensor data in a constantly-evolving environment

1.1. *Uncertain data mining*

There has been a growing interest in uncertain data mining[1], including clustering[2], [3], [4], [5], classification[6], [7], [8], outlier detection [9], frequent pattern mining [10], [11], streams mining[12] and skyline analysis[13] on uncertain data, etc. An important branch of mining uncertain data is to build classification models on uncertain data. While [6], [7] study the classification of uncertain data using the support vector model, [8] performs classification using decision trees. This paper unprecedentedly explores yet another classification model, Gaussian classifiers, and extends them to handle uncertain data. The key problem in Gaussian process method is the class conditional density estimation. Traditionally the class conditional density is estimated based on data points. For uncertain classification problems, however, we should learn the class conditional density from uncertain data objects represented by probability distributions.

2. Research Background

2.1. *Gaussian Process Models*

The conditional distribution $p(y \mid x)$ describes the dependency of an observable y on a corresponding input $x \in \chi$. The class of models described in this section assumes that this relation can be decomposed into a systematic and a random component. Further- more, the systematic dependency is given by a latent function $f : X \to \mathbb{R}$ such that the sampling distribution, i.e. the likelihood, is of the form

$$p(y \mid (f(x), \theta) \tag{2.1}$$

which describes the random aspects of the data-generating process. In general, we will use to denote additional parameters of the likelihood besides *f*. Note that the conditional distribution of the observable *y* depends on *x* via the value of the latent function *f* at *x* only. The aim of inference is to identify the systematic component *f* from empirical observations and prior beliefs. The data comes in the form of pair-wise observations $D = \{(y_i, x_i)\} \mid i = 1,..m\}$ where *m* is the number of samples. Let $X = [x_1, ..., x_m]^T$ and $y = [y_1, ..., y_m]^T$ collect the inputs and responses respectively. In general we assume $x \in \mathbb{R}^n$ unless stated otherwise. The name Gaussian process model refers to using a Gaussian process (GP) as a prior on *f*. The Gaussian process prior is non-parametric in the sense that instead of assuming a particular parametric form of *f*(x, θ) and making inference about, ϕ the approach is to put a prior on function values directly. Each input position $x \in \chi$ has an associated random variable *f(x).* A Gaussian process prior on f technically means that a priori the joint distribution of a collection of function values f=$[f(x_1), ..., f(x_{m'})^T]$ associated with any collection of m' inputs $[x_1. \ . \ . \ , x_{m'}]^T$ is multivariate normal (Gaussian)

$$p(f \mid X, \psi) = N(f \mid m, K) \tag{2.2}$$

with mean m and covariance matrix K. A Gaussian process is specified by a mean function *m(x)* and a covariance function $k(x, x', \psi)$ such that $K_{ij} = k(x_i, xj, \psi)$ and $m = [m(x_1), ...m(x_{m'})]^T$ ..By choosing a particular form of covariance function we may introduce hyper-parameters ψ to the Gaussian process prior. Depending on the actual form of the covariance function $k(x, x', \psi)$ the hyper-parameters ψ can control various aspects of the Gaussian process. The sampling distribution (2.1) of *y* depends on f only through *f(x).* As an effect the likelihood of *f* given *D* factorizes

$$p(y \mid f, \theta) = \prod_{i=1}^{m} p(y_i \mid f(x_i), \theta) = p \mid (y \mid f, \theta) \tag{2.3}$$

and depends on *f* only through its value at the observed inputs f . According to the model, conditioning the likelihood on f is equivalent to conditioning on the full function f. This is of central importance since it allows us to make inference over finite dimensional quantities instead of handling the whole function *f*. The posterior distribution of the function values f is computed according to Bayes' rule

$$p(\mathrm{f} \mid D, \theta, \psi) = \frac{p(y \mid \mathrm{f}, \theta) p(\mathrm{f} \mid X, \psi)}{p(D \mid \theta, \psi)} = \frac{N(\mathrm{f} \mid m, K)}{p(D \mid \theta, \psi)} \prod_{i=1}^{m} p(y_i \mid f_i, \theta) \tag{2.4}$$

Where $f_i = f(x_i)$.The posterior distribution of **f** can be used to compute the posterior predictive distribution of ***f*(x*)** for any input x* where in the following the asterisk is used to mark test examples. If several test cases are given, **X*** collects the test inputs and f* denotes the corresponding vector of latent function values. The predictive distribution of **f*** is obtained by integration over the posterior uncertainty

$$p(\mathrm{f}_* \mid D, X_*, \theta, \psi) = \int p(\mathrm{f}_* \mid \mathrm{f}, \mathrm{X}, \mathrm{X}_*, \psi) p(\mathrm{f} \mid D, \theta, \psi) df \tag{2.5}$$

where the first term of the right hand side describes the dependency of f_* on f induced by the GP prior. The joint prior distribution of f and f_* due to the GP prior is multivariate normal

$$p(\mathrm{f}_* \mid \mathrm{f}, \mathrm{X}, \mathrm{X}_*, \psi) = N\left(\begin{bmatrix} \mathrm{f} \\ \mathrm{f}_* \end{bmatrix} \middle| \begin{bmatrix} \mathrm{m} \\ \mathrm{m}_* \end{bmatrix}, \begin{bmatrix} \mathrm{K} & \mathrm{K}_* \\ \mathrm{K}_*^{\mathrm{T}} & \mathrm{K}_{**} \end{bmatrix}\right) \tag{2.6}$$

Where the covariance matrix is partitioned such that K_{**} is the prior covariance matrix of the f_* and K_* contains the covariances between f and f_*. The conditional distribution of $\mathrm{f}_* \mid$f can be obtained from the joint distribution (2.6) using relation to give

$$p(f_*|f,X,X_*,\psi)=N(f_*|m_*+K_*^T K^{-1}(f\text{-}m),K_{**}\text{-}K_*^T K^{-1}K_*) \quad (2.7)$$

which is again multivariate normal. The simplest possible model assumes that the function can be observed directly y = f (x) so that the posterior on f becomes p (f | D) = δ (f −y), describing that no posterior uncertainty about f remains. From this posterior the predictive distribution of f* can be obtained according to eq. (2.5) which corresponds to simply replacing f by y in eq. (2.7) Figure 1 shows the posterior Gaussian process which is obtained by conditioning the prior on the five observations depicted as points. The predictive uncertainty is zero at the locations where the function value has been observed. Between the observations the uncertainty about the function value grows and the sampled functions represent valid hypothesis about f under the posterior process.

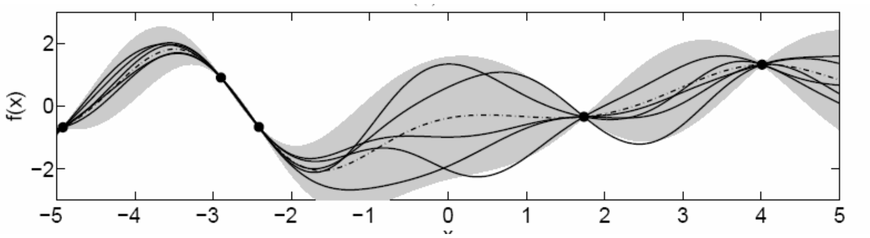

Fig. 1 show the posterior Gaussian process which is obtained by conditioning the prior on the five observations depicted as points

3. Overview of the Problem

We assume the following statistical model

$$t = f(x) + \varepsilon_t \quad (3.1)$$

Where x is a D-dimensional input and ε_t the output, additive, Gaussian uncertain data such that $\varepsilon_t \sim N(0, v_t)$, where v_t is the unknown data variance. Such a model implies that

$$E[t \,|\, \mathrm{x}] = f(x) \quad (3.2)$$

Now, let $\mathrm{x} = \mathrm{u} + \varepsilon_x$ or $\mathrm{x} \sim N(\mathrm{u}, v_x I)$ where I is the $D\ X\ D$ identity matrix and v_x is the input data variance. In this case, the expectation of t given the characteristics of x is obtained by integrating over the input distribution

$$E[t \,|\, \mathrm{u}, v_x] = \int f(x) p(x) dx \quad (3.3)$$

This integral cannot be solved analytically without approximations for many forms of f(x)

3.1. *Analytical approximation using the Delta Method*

The function f of the random argument x can always be approximated by a second order Taylor expansion around the mean u of x:

$$f(\mathrm{x}) = f(\mathrm{u}) + (\mathrm{x}-\mathrm{u})^T f'(\mathrm{u}) + \frac{1}{2}(\mathrm{x}-\mathrm{u})^T f''(\mathrm{u})(\mathrm{x}-\mathrm{u}) + O(\|\mathrm{x}-\mathrm{u}\|^3) \quad (3.4)$$

where $f'(\mathrm{u}) = \frac{\partial f(\mathrm{x})}{\partial \mathrm{x}}$, and $f''(\mathrm{u}) = \frac{\partial^2 f(\mathrm{x})}{\partial \mathrm{x} \partial \mathrm{x}^{\mathrm{T}}}$, evaluated at x=u .Within this approximation, we can now solve the integral (3.3). We have

$$E[t \,|\, u, v_x] \simeq \int \left[f(\mathrm{u}) + (\mathrm{x}-\mathrm{u})^T f'(\mathrm{u}) + \frac{1}{2}(\mathrm{x}-\mathrm{u})^T f''(\mathrm{u})(\mathrm{x}-\mathrm{u}) \right] p(\mathrm{x}) d\mathrm{x}$$

$$\simeq f(\mathrm{u}) + \frac{1}{2}\mathrm{Tr}[f''(\mathrm{u}) v_x I] = f(u) + \frac{v_x}{2}\mathrm{Tr}[f''(\mathrm{u})] \quad (3.5)$$

Where Tr denotes the trace. Thus, the new generative model for our data

$$\begin{cases} t = g(\mathrm{u}, vx) + \varepsilon_t \\ g(\mathrm{u}, v_x) = f(\mathrm{u}) + \frac{v_x}{2}\mathrm{Tr}[f''(\mathrm{u}) \end{cases} \quad (3.6)$$

4. RELATED WORKS

4.1. *Defining a new Gaussian Process*

In the case of uncertain or random inputs, the new input/output relationship is given by (3.6), where the former function *f*, in the noise-free case, has been replaced by $g(\mathrm{u},v_x) = f(u) + \frac{v_x}{2}\mathrm{Tr}[f''(\mathrm{u})]$. If we put a Gaussian prior on *f* (u), we can derive the corresponding prior on its second derivative and then define the prior on the space of admissible functions $g(\mathrm{u},v_x)$ which is viewed as the sum of the two correlated random functions, *f* (u) an $\frac{v_x}{2}\mathrm{Tr}[f''(\mathrm{u})$. We use results from the theory of random functions [3.5]. Let us recall that if $X(r)$ and $Y(r)$ are two random functions of the same argument *r,* with expected values $m_x(r)$ and $m_y(r)$ and covariance functions $C_x(r,r')$ and $C_y(r,r')$ respectively, then the mean and covariance function of $Z(r) = X(r) + Y(r)$ are given by

$$mz(r) = mx(r) + my(r) \tag{4.1}$$

$$C_z(r,r') = C_x(r,r') + C_y(r,r') + C_{xy}(r,r') + C_{yx}(r,r') \tag{4.2}$$

in the case $X(r)$ and $Y(r)$ are correlated $C_{xy}(r,r')$ and $C_{yx}(r,r')$ are the cross-covariance functions. We can now apply this to our function $g(.)$. Let us first derive the mean and covariance function of $g(\mathrm{u},v_x)$ in the one-dimensional case and then extend these expressions to *D* dimensions. Given that $f(u)$ has zero-mean and covariance function $C(u_i,u_j)$ as given by $C(\mathrm{u}_i,\mathrm{u}_j) = v\exp(-\frac{1}{2}\sum_{d=1} w_d(u_i^d - u_j^d)^2$, its second derivative, $f''(u)$ has zero-mean and covariance function $\partial^4 C(u_i,u_j)/\partial u_i^2 \partial u_j^2$. It is then straightforward that $\frac{v_x}{2} f''(u)$ has zero-mean and covariance function $\frac{v_x^2}{4}\partial^4 C(u_i,u_j)/\partial u_i^2 \partial u_j^2$ also, the cross-covariance functions between $f(\mathrm{u})$ and $\frac{v_x}{2} f''(u)$ is given by $\frac{v_x}{2}\partial^2 C(u_i,u_j)/\partial u_i^2$. Therefore, using the fact we have $\frac{\partial^2 C(u_i,u_j)}{\partial^2 u_i^2} = \frac{\partial^2 C(u_i,u_j)}{\partial^2 u_j^2}$, in one dimension, $g(u,v_x) = f(u) + \frac{v_x}{2} f''(u)$ has zero-mean and covariance function

$$\mathrm{cov}[g(u_i,v_x), g(u_j,v_x)] = C(ui,uj) + \frac{v_x^2}{4}\frac{\partial^4 C(u_i,u_j)}{\partial u_i^2 \partial u_j^2} + v_x \frac{\partial^2 C(u_i,u_j)}{\partial^2 u_i^2} \tag{4.3}$$

In the case of *D*-dimensional inputs, we have

$$\mathrm{cov}[g(\mathrm{u}_i,v_x), g(\mathrm{u}_j,v_x)] = C(\mathrm{u}i,\mathrm{u}j) + \frac{v_x^2}{4}\mathrm{Tr}\left[\frac{\partial^2}{\partial \mathrm{u}_i \partial \mathrm{u}_i^T}\left[\frac{\partial^2 C(\mathrm{u}_i,\mathrm{u}_j)}{\partial \mathrm{u}_i \partial \mathrm{u}_j^T}\right]\right] + v_x \mathrm{Tr}\left[\frac{\partial^2 C(\mathrm{u}_i,\mathrm{u}_j)}{\partial \mathrm{u}_i \partial \mathrm{u}_i^T}\right] \tag{4.4}$$

where $\frac{\partial^2}{\partial \mathrm{u}_i \partial \mathrm{u}_i^T}\left[\frac{\partial^2 C(\mathrm{u}i,\mathrm{u}j)}{\partial \mathrm{u}_j \partial \mathrm{u}_j^T}\right]$ is a *D X D* each entry of which being a *D X D* the block (r,s) contains $\frac{\partial^2}{\partial \mathrm{u}_i^r \partial \mathrm{u}_i^s}\left[\frac{\partial^2 C(\mathrm{u}i,\mathrm{u}j)}{\partial \mathrm{u}_j \partial \mathrm{u}_j^T}\right]$ So we see that the first term of the *corrected* covariance function corresponds to the noise-free case plus two correction terms weighted by the input noise variance, which might be either learnt or assumed to be known *a priori*.

4.2. *Inference and prediction*

Within this approximation, the likelihood of the data $\{t_1.....,t_N\}$ is readily obtained. We have

$$\mathrm{t}|\mathbf{U} \sim N(0,\mathbf{Q}) \text{ With } Q_{ij} = \sum\nolimits_{ij}' + vt\delta_{ij}$$

Where t is the *N X 1* vector of observed targets $\mathbf{U}$ .The *N X D* matrix of input means, $\sum_{ij}^{'\,\cdot}$ is given by (4.4) and $\delta_{ij} = 1$ when $i = j, 0$ otherwise. The parameters $\Theta = [w_1...., w_D, v, v_x, v_t]$ can then be learnt either in a Maximum Likelihood framework or in a Bayesian way, by assigning priors and computing their posterior distribution.When using the *usual* GP, the predictive distribution of a model output corresponding to a new input $\mathbf{u}_*$, $p(f(\mathbf{u}_*) \mid \Theta, \{\text{u,t}\}, u_*)$ is Gaussian with mean and variance respectively given by

$$\begin{cases} \mu = \mathbf{k}^T Q^{-1} \text{t} \\ \sigma 2 = k - \mathbf{k}^T Q^{-1} \mathbf{k} \end{cases} \tag{4.5}$$

Where $\mathbf{k}$ is the vector of covariances between the test and the training inputs and ~the covariance between the test input and itself. We have $Q_{ij} = \sum_{ij} + v_t \delta_{ij}$ and

$$\sum_{ij} = C(\mathbf{u}_i, \mathbf{u}_j), k = C(\mathbf{u}_*, \mathbf{u}_*) \tag{4.6}$$

for $i, j = 1, ..., N$ and with $C(.,.)$.

With our new model, the prediction at a new (one-dimensional) noise-free input leads to a predictive mean and variance, again computed using (4.6) but with $Q_{ij} = \sum_{ij} + v_t \delta_{ij}$ with $\sum_{ij}^{'}$ computed as (4.3), and

$$\begin{aligned} k_i &= C(u_*, u_i) + \frac{v_x}{2} \frac{\partial^2 C(u_*, u_i)}{\partial u_i^2} \\ k &= C(u_*, u_*) \end{aligned} \tag{4.7}$$

thus taking account of the randomness in the training inputs. With this new model, the prediction at a random input is straightforward, simply by using the *corrected* covariance function to compute the covariances involving the test input. Assuming $x_* \sim N(u_*, v_x)$ we have

$$\begin{aligned} k_i &= C(u_i, u_*) + \frac{v_x^2}{4} \frac{\partial^4 C(u_i, u_*)}{\partial u_i^2 \partial u_*^2} + v_x \frac{\partial^2 C(u_i, u_*)}{\partial u_i^2} \\ k &= C(u_*, u_*) + \frac{v_x^2}{4} \frac{\partial^4 C(u_*, u_*)}{\partial u_*^2 \partial u_*^2} + v_x \frac{\partial^2 C(u_*, u_*)}{\partial u_*^2} \end{aligned} \tag{4.8}$$

5. Experiments

5.1. *Results*

We have implemented the this approach using Matlab 6.5, on 3 real data sets taken from the UCI Machine Learning Repository i.e. Glass data, Iris, Wine data sets. We compare the classification performance of this model on this UCI datasets. Statistics of the datasets are listed in Table.1

Table 1. Statistics Of The Datasets Are Listed As Follows

	Glass	**Iris**	**Wine**
Number of Data	214	150	178
Number of features	10	4	13
Number of Classes	6	3	3

We specify a Gaussian process model as follows: a constant mean function, with initial parameter set to 0, a squared exponential with covariance function This covariance function has one characteristic length-scale parameter for each dimension of the input space, and a signal magnitude parameter, for a total of 3 parameters . We train the hyper-parameters using to minimize the negative log marginal likelihood. We allow for 40 function evaluations, and specify that inference should be done with the Expectation Propagation (EP) inference method and pass the usual parameters

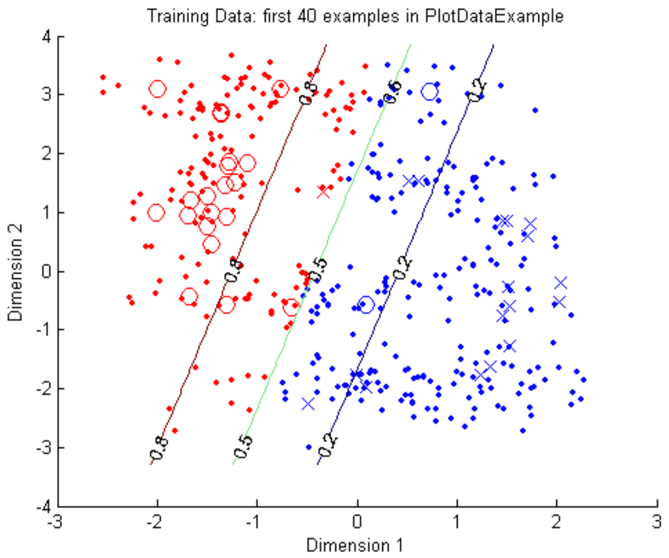

Fig. 2: Data plot for the given Data sets

We also Plot the prediction function with the Original data for Glass data

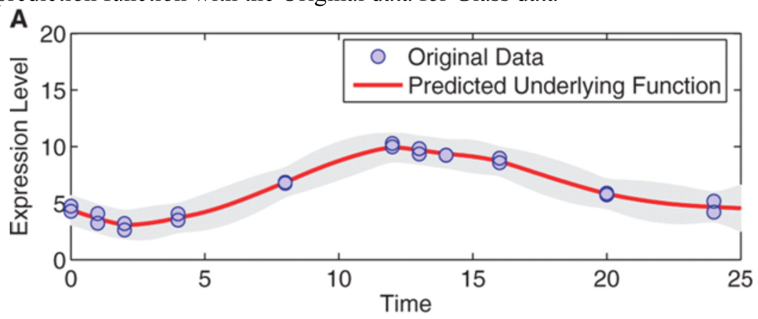

Figure 3: Plot for original data and the Predicted Data

We also plot Log-log plot of the log likelihood of the data against the length scales. The log likelihood is shown as a solid line. The log likelihood is made up of a data fit term (the quadratic form) shown by a dashed line and a complexity term (the log determinant) shown by a dotted line. The data fit is larger for short length scales; the complexity is larger for long length scales. The combination leads to a maximum around the true length scale value of 1

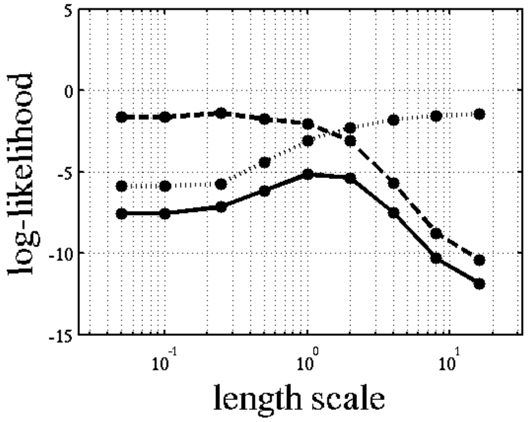

Fig 4: Plot for Length Scale and Log-Likelihood

Therefore, the classification and prediction process is more sophisticated and comprehensive and has the potential to achieve higher accuracy

6. Conclusion

In this paper, we propose a Gaussian Process model for classifying and predicting uncertain data. The new process is based on an approximation of the random function around the input mean. This process also highlights the correlation between all the parameters, indicating the nature of the likelihood function, and the potential problems for maximum likelihood optimization. We employ the probability distribution which represent the uncertain data attribute, and redesign the Gaussian Process so that they can directly work on uncertain data distributions. We plan to explore more classification approaches for various uncertainty models and find more efficient training algorithms in the future

References

[1] Aggarwal, C.C.: A Survey of Uncertain Data Algorithms and Applications. IEEE Transactions on Knowledge and Data Engineering 21(5) (2009)

[2] Cormode, G., McGregor, A.: Approximation algorithms for clustering uncertain data. In: Principle of Data base System, PODS (2008)

[3] Aggarwal, C.C., Yu, P.: A framework for clustering uncertain data streams. In: IEEE International Conference on Data Engineering, ICDE (2008).

[4] Singh, S., Mayfield, C., Prabhakar, S., Shah, R.,Hambrusch, S. :Indexing categorical data with uncertainty. In: IEEE International Conference on Data Engineering (ICDE), pp. 616–625 (2007)

[5] Kriegel, H., Pfeifle, M.: Density-based clustering of uncertain data. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 672–677 (2005). .

[6] Bi, J., Zhang, T.: Support Vector Machines with Input Data Uncertainty. In: Proceeding of Advances in Neural Information Processing Systems (2004)

[7] Aggrawal, R., Imielinski, T., Swami, A.N.: Database mining: A performance perspective.IEEE Transactions on Knowledge& Data Engineering (1993)

[8] Aggarwal, C.C.: Managing and Mining Uncertain Data. Springer, Heidelberg (2009)

[9] C. C. Aggarwal and P. S. Yu, "Outlier detection with uncertain data," in *SDM*. SIAM, 2008, pp. 483–493

[10] Hamdan, H. and Govaert, G. "Mixture Model Clustering of Uncertain Data," *IEEE International Conference on Fuzzy Systems* (2005) 879-884

[11] Girard, A. & Rasmussen, C. E. & Murray-Smith, R. (2002) Gaussian Process Priors With Uncertain Inputs: Multiple-Step Ahead Prediction. Technical Report, TR-2002-119, Dept. of Computing Science, University of Glasgow.

[12] Quinonero Candela, J. & Girard, A. (2002) Prediction at an Uncertain Input for Gaussian Processes and Relevance Vector Machines - Application to Multiple-Step Ahead Time-Series Forecasting. Technical Report, IMM, Danish Technical University.

[13] Rasmussen, C. E. (1996) Evaluation of Gaussian Processes and other Methods for Non-Linear Regression PhD thesis, Dept. of Computer Science, University of Toronto.

[14] Williams, C. K. I. & Rasmussen, C. E. (1996) Gaussian Processes for Regression Advances in Neural Information Processing Systems 8 MIT Press

[15] Williams, C. K. I. (2002) Gaussian Processes To appear in The handbook of Brain Theory and Neural Networks, Second edition MIT Press.

[16] Corinna Cortes and Vladimir N. Vapnik. Support-Vector Networks. Machine Learning, 20(3):273–297, 1995.

[17] Vladimir N. Vapnik. Statistical Learning Theory. John Wiley & Sons, NY, 1998

[18] Joaquin Qui ˜ nonero-Candela and Agathe Girard, "Prediction at an uncertain input for Gaussian processes and relevance vector machines - application to multiple-step ahead time-series forecasting," Tech. Rep., IMM, DTU, 2002, http://isp.imm.dtu.dk/staff/jqc/prop_uncert.ps.gz.