Protein Interaction Detection in Sentences via Gaussian Processes: A preliminary evaluation

Tamara Polajnar*, Simon Rogers and Mark Girolami

Department of Computing Science
University of Glasgow
Glasgow, Scotland, G12 8QQ
E-mail: tamara@dcs.gla.ac.uk
*Corresponding author

Abstract:
Classification methods are vital for efficient access of knowledge hidden in biomedical publications. Support vector machines (SVMs) are modern non-parametric deterministic classifiers that produce state of the art performances in text mining, and across other disciplines, while reducing the need for feature engineering. In this paper we offer a much needed evaluation of the Gaussian Process (GP) classifier, as a non-parametric probabilistic analogue to SVMs, which has been rarely applied to text classification. To this end, we provide an extensive experimental comparison of the performance and properties of these competing classifiers on the challenging problem of protein interaction detection in biomedical publications. Our results show that GPs can match the performance of SVMs without the need for costly margin parameter tuning, whilst offering the advantage of an extendable probabilistic framework for text classification.

Keywords: text mining; Gaussian process; support vector machine; protein interaction; sentence classification

Reference to this paper should be made as follows: Polajnar, T, Rogers, S and Girolami, M. (xxxx) 'An Evaluation of Gaussian Processes for Sentence Classification and Protein Interaction Detection', *Int. J. Data Mining and Bioinformatics*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Tamara Polajnar is a PhD student in the Department of Computing Science at the University of Glasgow. She has a BSc in computer science from University of Northern British Columbia and an MSc in natural language processing from University of Edinburgh. Simon Rogers received a degree in electrical and electronic engineering from the University of Bristol (2001), and the PhD degree in engineering mathematics from the University of Bristol (2004). He now holds a post-doctoral research associate post in the Department of Computing Science at the University of Glasgow. Mark Girolami is a Professor in the Department of Computing Science at the University of Glasgow. He also holds an EPSRC Advanced Research Fellowship which runs from 2007 until 2012.

1 Introduction

Each year vast amounts of biological knowledge are published in journal articles; at the beginning of 2008, for instance, MEDLINE contained over 16.8 million citations, compared to 16.1m in 2007 and 15.4m the year before (NIH, 2008). As the amount of textual data increases, relevant knowledge becomes more difficult to locate resulting in a need for increasingly sophisticated automatic extraction methods. Query-based search tools, which are popular with biomedical researchers due to their broad topic coverage (Hersh, 2005), have a low retrieval precision, which is leading to a more frequent use of text mining (TM) as a research and curation aid to extract key information from text (Donaldson et al., 2003). Many TM applications incorporate classification approaches (Cohen and Hersh, 2005) in order to assign labels to words or documents and pare down large collections into more relevant corpora. Hence, effective use of classification can bridge the gap between the flexibility of search engines and costly personalised TM applications.

Protein-protein interaction (PPI) detection is one of the key tasks in biomedical TM. Proteins are essential cellular components that regulate many functions in the life-cycle of an organism. Their movements and interactions with cellular components and each other are the subject of much biological investigation, the results of which are frequently published in journal articles. In this article we will investigate the detection of sentences that describe PPIs in biomedical abstracts using classification and shallow features. Shallow features refer to surface sentence properties such as the words, their part of speech tags, named entities, etc. this is in contrast to deep structural sentence features such as dependency graphs.

Information in text can be represented in multiple ways, from frequency of words and their co-occurrence to the grammatical structures that are employed. This allows for many different types of features that can be used for text classification, sometimes leading to high-dimensional feature spaces. In one popular model, *bag-of-words* (Lewis, 1998; Manning and Schütze, 1999, p. 237), each unique word in a collection is considered a feature, resulting in a very large, sparse data representation, e.g. a corpus of 250,000 words could yield as many as 10,000 features. Unfortunately, one of the major stumbling blocks to more flexible TM is the fact that the most popular methods for classification of text (such as hidden Markov models or maximum entropy) require expert knowledge for selection of the most effective features (Manning and Schütze, 1999; Berger et al., 1996). Consequently, non-parametric classifiers such as SVMs (Vapnik, 1995; Joachims, 1998) are quickly gaining ground (Song et al., 2005*a*; Wang and Wu, 2006; Sculley and Wachman, 2007; Silva et al., 2007; Theodosiou et al., 2007; Xu and Huang, 2007; Yu et al., 2008). They scale with the number of training documents, learn effectively from data with a large number of features, and allow for more relevant information to be captured by the data. In the past, SVMs have been used in a number of text processing tasks; for example, MEDLINE abstract classification (Donaldson et al., 2003; Yu et al., 2008), named entity recognition (Lee et al., 2003; Lee, Hwang, Kim and Rim, 2004; Takeuchi and Collier, 2005), gene functional annotation (Theodosiou et al., 2007), acronym extraction (Xu and Huang, 2007), hierarchical relation extraction (Wang et al., 2006), or as part of a larger system (Mitchell et al., 2005; Song et al., 2005*b*). SVMs have also been frequently used for PPI sentence detection, where they have been proven to be highly effective (Sugiyama et al., 2003;

Erkan et al., 2007). Bunescu et al. (2005), Airola et al. (2008), and Erkan et al. (2007) use structural features derived from dependency parses of the sentences with graph kernels, while Giuliano et al. (2006), for example, uses kernel combinations of shallow context-based features. In a comparison study between several classifiers, including decision trees and naïve Bayes, Huang et al. (2003) find that SVMs perform the best on their PPI detection data set.

However, SVMs are only one example from a family of non-parametric methods that also includes a Bayesian method, Gaussian processes (GPs). While GPs have properties similar to SVMs (Opper and Winther, 1999; Rasmussen and Williams, 2006, pp. 141–146) they have failed to attract the same kind of attention in the text processing community. They have been applied to a variety of other bioinformatics tasks, such as protein fold prediction (Girolami and Zhong, 2007; Lama and Girolami, 2008) and biomarker discovery in microarray data (Chu et al., 2005). Kote-Jarai et al. (2006) use both GPs and SVMs to detect which cells have a certain DNA mutation. Their results showed that GPs were able to distinguish cells with BRCA1 and BRCA2 mutations with 100% accuracy, whereas the SVM had an accuracy of 90% over 10 leave-one-out experiments. GPs have also been applied to text classification, but only in a few instances. Online Gaussian processes (Chai et al., 2002) and Informative Vector Machines were investigated for multiple classes on the Reuters collection by Stankovic et al. (2005). In addition, GPs and SVMs were compared for preference learning on the OHSUMED corpus (Chu and Ghahramani, 2005*b*) and an extension of GPs for sequential data such as named entities was proposed by Altun et al. (2004).

In this article we offer for the first time an extensive experimental evaluation of GPs for identification of protein-protein interactions in biomedical texts. To examine probabilistic and non-parametric qualities of this algorithm, we will compare GPs with naïve Bayes (NB) and the SVM. In order to compare the training potential of abstracts versus sentences, we use binary data sets compiled from biomedical abstracts and marked up for presence or absence of PPIs. In addition, we demonstrate the effectiveness of a single multiclass GP classifier versus a collection of binary SVMs by considering a further corpus, collected from full-text articles and annotated for different types of interactions. With various types of information stored in biomedical texts, efficient multiclass classification is essential. Finally, we demonstrate that the multiclass classifier can be further extended to perform semi-supervised learning, allowing the costly expert-annotated data to be effectively leveraged with unlabelled text.

For these experiments, we will use shallow bag-of-words features, and compare this approach with a feature set augmented so that protein names are replaced with placeholder strings, to show that this simple method leads to an increase in classification accuracy of sentence data regardless of whether the proteins are annotated manually or automatically. The disadvantage of shallow features is that the granularity of information extraction is as small as the annotation. As we mainly rely on relevance judgements we can only find abstracts or sentences that contain the interaction and not the interactions themselves. In cross-corpus experiments we examine if it is possible to increase the granularity of prediction by training on abstracts, but classifying sentences. The advantage of using minimal annotation to extract features, on the other hand, is that any errors, such as can be seen in the following example, are not propagated through the system. The sentence below is an example from the UTexas corpus where the number of interactions was correctly annotated, but the main interacting protein *IL-8* was incorrectly annotated in a way that is grammatically difficult to process. The effect is that the subject protein of the sentence is no longer interacting with the object proteins.

This work shows that single and double Ala substitutions of His18 and Phe21 in < prot> IL - 8 </prot> reduced up to 77 - fold the binding affinity to <prot> < p1 pair=1 > <p1 pair=2 > <p1 pair=3 > <prot> IL - 8 </prot> </p1> </p1> </p1> receptor subtypes A </prot> (<p2 pair=1 > <prot> CXCR1 </prot> </ p2>) and B (<p2 pair=2 > <prot> CXCR2 </prot> </p2>) and to the <p2 pair=3 > <prot> Duffy antigen </prot> </p2> .

By reducing the complexity of the task we gain in accuracy in locating PPIs in the documents, but only to sentence precision. We do not provide the actual interacting pairs. Our approach improves on the F-score of systems that identify the interacting pairs by approximately 20% F-score. For example, Airola et al. (2008) achieve an F-score of 56.4 (84.8 AUC) on the UTexas data set using an SVM and a graph kernel with dependency-tree features, compared to the GP F-score of 71.0 (79.0 AUC). By relying on minimal annotation for training purposes we present a method that is adequate for research and curation purposes, but would need a further layer of processing for automatic knowledge-base population.

On the whole, the experimental results demonstrate an equivalence between GPs and SVMs regarding raw classification performance. GPs have a significantly higher F-score and AUC on sentence data, but are equivalent to SVMs on abstract and multiclass data. The naïve Bayes classifier often has the highest F-score, but this is an artefact of high recall (defined in Section 4.6). The low precision rate is reflected in the low AUC and classification accuracy. Thus in comparison with the state-of-the art SVM approach GPs can offer a performance improvement. In addition, they operate within a probabilistic framework permitting elegant and advantageous algorithm extensions, such as multiclass and semi-supervised GPs.

The next section describes GPs and SVMs and contrasts these to the NB classifier. Section 3 provides a brief discussion of the similarities and the differences between the two classifiers. Section 4 describes the data and the experiments which were used to compare GPs and SVMs and Section 5 shows the results of these experiments. Finally, in Section 6 we present our conclusions.

2 Description of the Algorithms

Input into all three algorithms is a matrix representation of the data. In sentence classification, using a bag-of-words model, each sentence is represented as a row in the data matrix, $\mathbf{X}$. Considering N documents containing M unique features, the i^{th} document corresponds to the vector $\mathbf{x}_i = [x_{i1}, \ldots, x_{im}]$ where each x_{ij} is a count of how many times word j occurs in the document i. These vectors are then used directly by the NB, while for the GPs and SVMs the *kernel trick* (Aizerman et al., 1964; Boser et al., 1992) is then used to embed the original feature space into an alternative space where data may be linearly separable. That kernel function transforms the NxM input data to a square NxN matrix, called the *kernel*, which represents the similarity or distance between the documents. The principal difference between the approaches is in how the kernel is used; while SVMs use geometric means to discriminate between the positive and negative classes, GPs model the posterior probability distribution of each class.

2.1 Support Vector Machine

The goal of training an SVM is to determine, within the kernel space, a separating hyperplane that divides the two data classes. The chosen decision boundary is that which maximises the margin - the perpendicular distance between the boundary and the closest points on either side. These points are called *support vectors* and are chosen through quadratic programming. When a new point is tested, the SVM determines on which side of the hyperplane it lies. The separating function is $sign(\sum_{i=1}^{n} \lambda_i t_i \mathbf{C}(\mathbf{x}_i \cdot \mathbf{x}_*) + w_0)$, where λ_i are non-negative Lagrange multipliers that are non-zero only for support vectors, $\mathbf{C}(\mathbf{x}_i \cdot \mathbf{x}_*)$ is the inner product of the training vector $\mathbf{x}_i$ and the test vector $\mathbf{x}_*$, t_i is the class of the training vector (-1 for negative, 1 for positive), and w_0 is a weight vector offset (Shawe-Taylor and Cristianini, 2004; Rasmussen and Williams, 2006). If the data is non-linearly separable in the kernel space, or there is suspected noise, such as annotation errors, the margin parameter C is introduced to balance the need to compensate for noise in the data with maintaining clear class boundaries. In practice, the margin parameter is essential for accurate classification and needs to be calibrated for each particular data set. If a kernel with hyperparameters is used, these hyperparameters need to be tuned along with C. The tuning is done via cross validation experiments which range through a set of possible values for each of the parameters. For example, searching over a small grid of 10 possible values for the SVM parameter and 10 for one kernel parameter would require 100 separate cross-validation experiments.

2.2 Gaussian Process

Since it operates within a probabilistic framework, the GP classifier, which is also a kernel-based method, does not employ a geometric boundary and hence does not require a margin parameter. Instead, we use the GP framework to predict the probability of class membership for a test vector $\mathbf{x}_*$. This is achieved via a latent function $m(\mathbf{x})$, which is passed through a step-like likelihood function in order to be constrained to the range $[0, 1]$ to represent class membership. The smoothness of $\mathbf{m} = \{m_i = m(\mathbf{x}_i) | \mathbf{x}_i \in \mathbf{X}\}$ is regulated by a Gaussian process prior placed over the function and further specified by the mean and covariance functions.

In other words, the model is described by the latent function $\mathbf{m}$ such that $p(\mathbf{m}) = \mathcal{N}(\mathbf{m}|0, \mathbf{C})$, where $\mathbf{C}$ analogous to the kernel function in the SVMs and would normally require some parametrisation. The function posterior is $p(\mathbf{m}|\mathbf{X}, \mathbf{T}) \propto p(\mathbf{T}|\mathbf{m})p(\mathbf{m}|\mathbf{X})$. In GP regression this is trivial as both terms are Gaussian; however, in the classification case the non-conjugacy of the GP prior and the likelihood, $p(\mathbf{Y}|\mathbf{m})$ which can be for example probit, makes inference non-trivial.

In order to make predictions for a new vector $\mathbf{x}_*$, we need to compute the predictive distribution $p(t_*|x_*, \mathbf{X}, \mathbf{T}) = \int p(t_*|\mathbf{x}_*, \mathbf{m})p(\mathbf{m}|\mathbf{X}, \mathbf{T})d\mathbf{m}$, which is analytically intractable and must be approximated. The strategy chosen to overcome this will depend on the likelihood function chosen (options include the logistic and probit functions). In this work, we follow Girolami and Rogers (2006) and use the probit likelihood, $p(t_i = 1|m_i) = \Phi(m_i) = \int_{-\infty}^{m_i} \mathcal{N}(z|0, 1)dz$, where the auxiliary variable trick (Albert and Chib, 1993) enables exact Gibbs sampling or efficient variational approximations.

2.3 Naïve Bayes

The naïve Bayes is a generative probabilistic classifier, and as such it does not determine a discriminative boundary like SVMs, but instead it is used to check whether a particular document was generated by a particular distribution. While the GP estimates the probability of class membership given a document, the NB does this by directly applying the Bayes rule and calculating the probability that a document belongs to a particular class.

In contrast to both GPs and SVMs, NB is parametric and thus its complexity grows with the number of features and not with the number of documents. For the data sets of the size we are using here the document or feature number is not sufficiently high to visibly distinguish the algorithms based on these properties. However, using larger data sets would increase the number of features as well as the number of documents, in which case feature selection would have to be used in order for NB to be used effectively. On the other hand, all features could be employed by both SVMs and GPs and the document number would not effect the sparse implementation of the GP, the IVM.

We use a multiclass NB mixture model, where each mixture component corresponds to a class. The classifier is an application of the Bayes rule. It calculates the probability of a class given a document $p(c_j|\mathbf{x}_i) = \frac{p(c_j)p(\mathbf{x}_i|c_j)}{p(\mathbf{x}_i)}$, where c_j is one of K classes. Each document is generated by the mixture of classes $p(\mathbf{x}_i) = \sum_{j=1}^{K} P(c_j)p(\mathbf{x}_i|c_j)$. Calculation of the $p(\mathbf{x}_i|c_j) \propto p(|\mathbf{x}_i|)\prod_{w \in W} p(w|c_j)$ term involves multiplication over all words w in the vocabulary W, a procedure that can sometimes lead to numerical errors when there is a large number of features.

We use a Dirichlet prior with parameter settings that correspond to Laplace smoothing. The implementation is described in detail in Nigam et al. (2006) and Lewis (1998).

3 Comparison of Gaussian Processes and SVMs

GPs and SVMs are both non-parametric kernel-based algorithms whose complexity grows with the number of training documents and not with the number of features. In fact, SVMs can be considered as a special case of GPs (Opper and Winther, 1999; Rasmussen and Williams, 2006, pp. 141–146). Nevertheless, while they can both be applied to the same data transformed by the same kernel (covariance) functions, there are differences stemming from the fact that GPs are probabilistic in nature.

3.1 Comparison of GP and SVM complexity

SVMs have benefited from widely available implementations, for example the C implementation SVM^{light} (Joachims, 2007, 1999), which use an algorithmic procedure where only a subset of training data is used; however, informative vector machines (IVMs) (Lawrence et al., 2003, 2005; Girolami and Rogers, 2006), which are derived from GPs, now offer an analogous probabilistic alternative. Naive implementation of SVM has the computational complexity $O(N^3)$, due to the quadratic programming optimisation. However, with engineering techniques this can be reduced to $O(N^2)$ or even more optimally to $O(ND^2)$ where D is a much smaller set of carefully chosen training vectors (Keerthi et al., 2006). Likewise, the GP has $O(N^3)$ complexity; with techniques such as the IVM this can be reduced to the worst case performance of $O(ND^2)$. On the datasets presented in this paper the difference for combined training and classification user time for GPs and SVMs was imperceptible.

3.2 Benefits of the probabilistic non-parametric approach

The clear advantages of the probabilistic approach to classification have inspired attempts to develop probabilistic extensions of SVMs. For example, Platt (1999) proposed an *ad-hoc* mapping of SVM output into probabilities; however, this is not a true probabilistic solution as it yields probabilities that tend to be close to 0 or 1 (Rasmussen and Williams, 2006, p. 145). On the other hand, the GP output probabilities give an accurate depiction of class membership that can be used to choose the optimal precision-recall trade off for a particular problem or further post-processing for appropriate decision making.

3.3 Extensibility of the algorithms

The Bayesian framework allows for additional mathematical extensions of the basic algorithm, such as multiple classes (Rasmussen and Williams, 2006; Girolami and Rogers, 2006; Seeger and Jordan, 2004), sequential data (Altun et al., 2004), and ordinal classes (Chu and Ghahramani, 2005*a*). One advantage of the particular Gaussian process classifier used in this paper is its ability to handle unlabelled training data (semi-supervised learning (Rogers and Girolami, 2007)). This is especially useful in text classification since there is a wealth of unlabelled documents available. SVMs can also be used for semi-supervised learning (Silva et al., 2007); however, here we combine semi-supervised and multiclass learning in a single classifier, following the semi-supervised extension to the multinomial probit classifier in Rogers and Girolami (2007). Essentially, the null category likelihood of Lawrence and Jordan (2006) is extended to the multi-class setting, by augmenting the problem with an additional *null* class, inside which no data (labelled or unlabelled) can exist. This has the effect of forcing the GP decision boundaries to lie in areas of low data density, thereby enforcing the so-called *cluster assumption* (Lawrence and Jordan, 2006) - the assumption that in the input space, data will be concentrated in class-specific clusters surrounded by areas of low data density.

SVMs are frequently used on many different kinds of binary data, but extension of SVMs for multiple class classification is more difficult than for GPs. There are theoretical extensions. For example Lee, Lin and Wahba (2004) demonstrate the use of multiclass SVM on cancer microarray data; however, the implementation is $O(N^3K^3)$ (Crammer and Singer, 2001), where K is the number of classes. Thus most applications of SVM to multiple class problems use combinations of multiple binary classifiers, as was the case in an application to the multiclass corpus of hierarchical relations in news text by Wang et al. (2006) and comprehensively presented in Hsu and Lin (2002).

Two popular strategies are *one vs. all* and *one vs. one* . When using the former strategy one class is considered positive and the rest are negative resulting in K classifiers, while in the latter approach each class is trained against each of the others resulting in $\frac{K \cdot (K-1)}{2}$ classifiers. For example, (Ding and Dubchak, 2001) use 351 SVM classifiers, per feature space, to predict 27 protein fold classes. For the same problem, Damoulas and Girolami (2008) demonstrate how a single probabilistic multiclass kernel machine tailored to learn from multiple types of features for protein fold recognition can outperform a multiple classifier SVM solution.

This work is an example of how the probabilistic framework enables elegant extensions of the algorithm. Such extensions of the GP include the basic binary classifier, multiclass GP (Girolami and Rogers, 2006, e.g.), and the IVM (Lawrence et al., 2003). In this paper we investigate the suitability of the GP for PPI detection in order to build a solid foundation for further investigation. To this end we apply the binary, multiclass, and semi-supervised GPs to various PPI corpora.

4 Corpora and Experiments

We conducted four different experiments using high-dimensional feature spaces produced by the bag-of-words representation in order to demonstrate the range of capabilities of the GP classifier. In the first three experiments we used the binary corpora, initially with plain features, but subsequently we made use of named entity recognition (NER) software to find protein names. We then used the tags produced by the NER algorithm to anonymise the proteins from the text, producing interaction templates which can be matched to an interaction regardless of the specific proteins involved. Finally, we used the BioText multiclass corpus with the plain features. The data sets, experiments, and feature extraction procedures are described below.

4.1 Data

One of the most researched tasks in biomedical text mining is relation detection (Cohen and Hersh, 2005). For this task, there are several available data sets that are annotated with protein-protein interactions (PPIs), and we compare SVM and GP performance on three such corpora (described in Table 1) containing at least two classes of examples, for instance corpora including sentences with and without PPIs. Two of the corpora, PreBIND (Donaldson et al., 2003) and UTexas (Bunescu et al., 2005), were collected from MEDLINE abstracts, and subsequently annotated for presence or absence of interactions. PreBIND was originally collected to train an SVM to detect abstracts and sentences that describe PPIs and is used as an aid in the curation of the BIND database (BIND, 2007). It comprises of abstracts carefully chosen from results of various MEDLINE queries and with the aid of an SVM trained on a smaller corpus. PreBIND is annotated solely for the presence or absence of an interaction in the abstract. Meanwhile, UTexas is a collection of 230

Corpus	Classes	Documents	Positive	Feature types	Annotation
PreBIND	2	1093	694 (63%)	plain, NER	Presence or absence of an interaction in the entire abstract
UTexas	2	1980	614 (31%)	plain, NER, annotated	Protein names, sentence level interactions
BioText	25	3,020	39 to 416	plain	type of interaction

Table 1 Short description of the corpora. For the multiclass BioText corpus the number of positive elements varies from class to class between 39 to 416.

abstracts, from the Database of Interacting Proteins (DIP) (Xenarios et al., 2001), in which every sentence is individually annotated for proteins and interactions. Finally, the third corpus, BioText, was compiled from full-text articles, referenced in the HIV Human Protein Interaction Database and separated into several types of interactions, including *interacts with, stimulates, inhibits,* and *binds* (Rosario and Hearst, 2005). The selection procedure for the candidate abstracts in each of these corpora was different, resulting in a distribution of positive examples that differ from the natural distribution of PPIs in the whole MEDLINE corpus.

4.2 Experiment 1: Cross-validation with plain features

Firstly, we applied both the GP and SVM classifiers to the PreBIND and UTexas corpora in a 10-fold cross-validation experiment, repeated ten times, with plain bag-of-words features. These features were extracted by scanning the text, changing each word to lower case, excluding the stop words, and finally cropping the remaining ones to maximum of ten letters. Each feature is then mapped to a column $\mathbf{x}_j$ of the data matrix, while each document is represented by a row $\mathbf{x}_i$. Each matrix entry, x_{ij}, was the number of times a word occurred in a document, which in the case of PreBIND is an abstract, but in UTexas is a sentence. For example, from this sentence in the UTexas corpus:

```
We have identified a new TNF - related ligand , designated human <p1  pair=2 >
<prot>  <p1  pair=1 >  GITR </p1>  ligand </prot>  </p1>  ( <p1  pair=3 >  <p2
pair=1 >  <prot>  hGITRL </prot>  </p2>  </p1>  ) , and its human receptor ( <p2
  pair=2 >  <p2  pair=3 >  <prot>  hGITR </prot>  </p2>  </p2>  ) , an ortholog
of the recently discovered murine <prot>  glucocorticoid - induced TNFR - relate
d ( <prot>  mGITR </prot>  ) protein </prot>  [ 4 ] .
```

we can extract the following plain features:

```
identified tnf related ligand designated human gitr ligand hgitrl human receptor
 hgitr ortholog recently discovered murine glucocorti induced tnfr related mgitr protein
```

Stemming and term frequency/inverse document frequency (TF-IDF) (Manning and Schütze, 1999, pp. 541–544) word weighting were examined as alternative representations, but both lead to a decrease in performance. All of the unique words that appeared in a corpus were collected, but features which did not occur in the training data were also removed from the test data. We extracted 8,386 features from the PreBIND corpus, 3,393 words from UTexas, and 1,625 words from BioText.

4.3 Experiment 2: Cross-validation with protein features

Next, we conducted a 10-fold cross-validation experiment, repeated ten times, using protein names as features. As can be seen in the example above, the UTexas corpus was annotated for protein names; at the same time, PreBIND has only minimal annotation (see Table 1). So while we can evaluate the effect of using gold standard protein annotation, we cannot do it consistently for both corpora. For this reason we employed Lingpipe hidden Markov model (HMM) NE tagger to find proteins (Alias-I, 2008). This tagger was trained on the GENIA corpus (Kim et al., 2003) that was marked up for several types of named entities. Out of these, there are a few which are related to proteins including *protein_molecule, protein_family_or_group*, and *protein_complex*. We compared multiple combinations of these annotations to the gold standard UTexas data, and chose *protein_molecule* as the best representation. Further analysis is provided in Section 5.2. Continuing with the example, we can see the product of NER:

```
We have identified a new <ENAMEX TYPE="protein_molecule">TNF - related ligand</ENAMEX> ,
designated <ENAMEX TYPE="protein_molecule">human  GITR ligand</ENAMEX> (
<ENAMEX TYPE="protein_molecule">hGITRL</ENAMEX> ) , and its
<ENAMEX TYPE="protein_family_or_group">human receptor</ENAMEX>
(  <ENAMEX TYPE="protein_family_or_group">hGITR</ENAMEX> ) , an ortholog of the recently
discovered  <ENAMEX TYPE="protein_family_or_group"> murine  glucocorticoid - induced TNFR -
related ( mGITR ) protein </ENAMEX> [ 4 ] .
```

The resulting NEs were translated into features by substituting a place-holder string `PTNGNE`, concatenated with the protein index, for the words comprising the NE. The index is a counter from 1 to the total number of proteins in the document, where each occurrence of a protein in a document was counted as unique unless it was enclosed in parentheses following another protein. In this case the simple algorithm assumed that both tagged entities referred to the same protein, but it did not keep track if the same protein occurred twice in different parts of the document. The final features extracted were:

```
identified  ptngne1 designated ptngne2 ptngne2 human receptor ortholog recently discovered
murine glucocorti induced tnfr related mgitr protein
```

The original protein names were discarded as they provided no increase in cross validation performance on UTexas data. When the same procedure was repeated with the gold standard annotations for UTexas, we found 3,105 features. However, when considering the automatically extracted NEs there were 3,226, indicating that less proteins were found by the HMM tagger than were hand annotated. For PreBIND there were 7,824 features after automatically extracted protein names were substituted for the `PTNGNE` tags.

4.4 Experiment 3: Cross-corpora evaluation

Unlike the previous two, the following is not a cross-validation experiment, instead, it is a cross-corpus evaluation. In order to test how a model based on one data set would perform, for example, as a tool used to select PPI sentences from various MEDLINE queries, we test it out on the other available binary data set. Because PreBIND and UTexas were compiled by annotators through deliberate sampling from the MEDLINE database, the distributions of positive to negative examples differ, making them ideal for such an experiment. In the PreBIND corpus each document is an abstract, while in UTexas each document is a sentence, consequently the documents have different lengths and different numbers of proteins and other interaction-indicating words. We used both types of features to examine the effects of the differing training distributions and to compare the knowledge in plain text versus the knowledge in the template-like NER features.

4.5 Experiment 4: Multiclass and semi-supervised GP extensions

Finally, we used the multiclass BioText corpus to demonstrate two extensions of the basic GP algorithm, the multiclass and the semi-supervised. In the first part of the experiment the data was evaluated in a cross-validation experiment using a single multiclass GP (Girolami and Rogers, 2006) and a one vs. one implementation of the SVM (Cawley, 2000).

BioText was then used in a GP multiclass semi-supervised classifier to investigate effects of learning with partially labelled data. We employ the semi-supervised classifier described in Rogers and Girolami (2007), which allows inputs that belong to none of the classes. The classifier works by assigning high-density areas of unlabelled data points that surround labelled points the corresponding class. The data points are artificially randomly unlabelled and passed to the classifier.

Here half the data was used for training and the other half was reserved for testing. Instead of doing cross-validation, this division was randomly chosen, and repeated 20 times to create different train-test splits. The purpose of the experiment was to measure the accuracy when the GP was trained with the labelled data only, and then compare this to the results of training with both the labelled and unlabelled data. To this end, we tested through a range percentages, by retaining 1, 2, 5, 10, 20, or 50% of the labels in the training set, and using the rest as unlabelled. The results in Figure 2 show the difference between these two measurements.

4.6 Experimental setup and evaluation

We used the cosine kernel $k(\mathbf{x}_i, \mathbf{x}_*) = \frac{\mathbf{x}_i \cdot \mathbf{x}_*}{|\mathbf{x}_i||\mathbf{x}_*|}$ in all of the experiments. We also considered the Gaussian kernel, but found it did not increase the area under the ROC curve for either of the data sets (which was 0.83 for the SVM with both kernels, and 0.67 for the GP with the Gaussian and 0.80 with the cosine kernel). Results were evaluated using the precision, recall, and F measures, which are defined in terms of true positives (tp), false positives (fp), true negatives (tn), and false negatives (fn): $precision = \frac{tp}{tp+fp}$, $recall = \frac{tp}{tp+fn}$, $F = \frac{2 \cdot precision \cdot recall}{precision+recall}$ (Van Rijsbergen, 1979). The area under the receiver operator characteristic (ROC) curve is also employed as a standard measure. The ROC is a plot of the true positive rate vs. the false positive rate, and the larger the area under the curve (AUC) the better the performance of the classifier. When perfect classifier performance is achieved the AUC is 1. To assess the the quality of the rankings provided by the algorithms we use the mean average precision (MAP) measure. MAP is a frequently used information retrieval measure. The measure assesses the precision of the top results by calculating the ranking of the results and averaging across queries. Here we consider a cross-validation run a query. If the output of a classifier is considered

Data	Features	NB	GP	SVM
UT	Plain	†F=**0.6785 ± 0.0080** †A=51.4009 ± 0.9111 †P=0.5140 ± 0.0091 †R=1.0000 ± 0.0000 †AUC=0.2894 ± 0.0076	†F=0.6441 ± 0.0105 †A=77.1309 ± 0.7102 †P=0.6236 ± 0.0096 †R=0.6679 ± 0.0160 †AUC=**0.7365 ± 0.0126**	†F=0.6014 ± 0.0130 †A=74.0353 ± 0.7717 †P=0.5744 ± 0.0118 †R=0.6336 ± 0.0194 †AUC=0.7030 ± 0.0139
UT	annotated	F=0.6915 ± 0.0108 †A=52.9561 ± 1.2742 †P=0.5296 ± 0.0127 †R=1.0000 ± 0.0000 †AUC=0.2617 ± 0.0158	†F=**0.7099 ± 0.0154** †A=81.0926 ± 0.8885 †P=0.6757 ± 0.0175 R=0.7518 ± 0.0210 †AUC=**0.7898 ± 0.0102**	F=0.6872 ± 0.0178 †A=78.7958 ± 1.2361 †P=0.6350 ± 0.0184 R=0.7532 ± 0.0237 †AUC=0.7738± 0.0118
UT	NER pm	†F=**0.7243 ± 0.0141** †A=56.9674 ± 1.7439 †P=0.5697 ± 0.0174 †R=1.0000 ± 0.0000 †AUC=0.2399 ± 0.0057	†F=0.7117 ± 0.0087 †A=81.4798 ± 0.3983 †P=0.6878 ± 0.0133 †R=0.7413 ± 0.0159 †AUC=**0.7886 ± 0.0075**	†F=0.6611 ± 0.0141 †A=78.1370 ± 0.7351 †P=0.6345 ± 0.0129 †R=0.6926 ± 0.0205 †AUC=0.7500 ± 0.0097
UT	NER pm+pfg	†F=**0.6455 ± 0.0153** †A=47.8439 ± 1.6409 †P=0.4784 ± 0.0164 †R=1.0000 ± 0.0000 †AUC=0.3092 ± 0.0082	†F=0.5925 ± 0.0180 †A=74.2450 ± 1.1850 †P=0.5876 ± 0.0259 R=0.6074 ± 0.0232 †AUC=**0.6942 ± 0.0173**	†F=0.5556 ± 0.0075 †A=70.1948 ± 0.6240 †P=0.5196 ± 0.0133 R=0.6052 ± 0.0198 †AUC=0.6655 ± 0.0123

PB	Plain	†F=0.8350 ± 0.0095
		†A=71.7861 ± 1.4432
		†P=0.7179 ± 0.0144
		†R=1.0000 ± 0.0000
		†AUC=0.3590 ± 0.0140
			F=**0.8621** ± **0.0114**
			A=82.6097 ± 1.2976
			P=0.8600 ± 0.0142
			†R=0.8651 ± 0.0121
			AUC=**0.8069** ± **0.0157**
				F=0.8547 ± 0.0091
				A=81.7756 ± 1.1916
				P=0.8656 ± 0.0165
				†R=0.8453 ± 0.0041
				AUC=0.8033 ± 0.0158

PB	NER pm	†F=**0.8141 ± 0.0074**
		†A=68.7152 ± 1.0689
		†P=0.6872 ± 0.0107
		†R=1.0000 ± 0.0000
		†AUC=0.4131 ± 0.0170
			F=0.7187 ± 0.0148
			A=64.2192 ± 1.6666
			P=0.7166 ± 0.0197
			R=0.7251 ± 0.0188
			AUC=0.6128 ± 0.0213
				F=0.7264 ± 0.0115
				A=65.1232 ± 1.0334
				P=0.7205 ± 0.0119
				R=0.7358 ± 0.0187
				AUC=**0.6239** ± **0.0124**

PB	NER pm+pfg	F=0.8461 ± 0.0073
		†A=73.3874 ± 1.0987
		†P=0.7339 ± 0.0110
		†R=1.0000 ± 0.0000
		†AUC=0.3390 ± 0.0161
			F=0.8535 ± 0.0099
			A=81.4715 ± 1.1134
			P=0.8530 ± 0.0131
			R=0.8553 ± 0.0120
			AUC=0.8009 ± 0.0196
				F=**0.8575** ± **0.0130**
				A=82.0506 ± 1.5046
				P=0.8585 ± 0.0125
				R=0.8578 ± 0.0169
				AUC=**0.8163** ± 0.0217

Table 2 Cross-validation results for NB, GPs, and SVMs. The results are presented as F-score (F), accuracy (A), precision (P), recall (R), and area under the ROC (AUC), and include the standard error. The † symbol indicates that the paired t-test significance analysis shows that the difference between the indicated value and the corresponding values from the other two algorithms is significant (P-value < 0.05). In the feature column, NER *pm* indicates that we used entities labelled *protein_molecule* as features, while *pm+pfg* indicates we also used entities labelled with *protein_family_or_group*.

a ranked list, then the for the top results the average precision (AP) is calculated by adding the inverse of the ranking, so the if we consider the top 5 results and the first relevant result is ranked third, and if the next one is fifth the AP for the run would be $\frac{\frac{1}{3}+\frac{1}{5}}{5}$. The MAP is then the sum of APs divided by the number of cross-validation runs. The outcomes are reported in Tables 2 and 5 for the binary, and in Section 5.4 for the multiclass and semi-supervised classification experiments.

5 Results and Discussion

Across the different experiments we can see that GPs either score higher or there is no significant difference between the GPs and SVMs. In the binary cross-validation experiments the NB has a high F-score, but a significantly lower AUC than either GPs or SVMs in all experiments. Likewise, in the binary experiments we demonstrate that using protein features increases classification performance regardless of whether proteins are identified manually or through automatic means.

We also examined the output from each of the classifiers. We found that the top ranked results for the NB and the SVM actually contained more negative results than the GPs. We assessed the quality of the ranking by using the MAP measure as defined in Section 4.6. The results in Table 3 show that GPs give higher probability and thus a better ranking to relevant documents than NB and SVM. The

No. of results	NB	GP	SVM
5	†0.1790 ± 0.0185	0.3063 ± 0.0273	0.2567 ± 0.0236
10	0.1870 ± 0.0147	0.2470 ± 0.0202	0.2267 ± 0.0193
30	0.1648 ± 0.0069	0.1910 ± 0.0177	0.1726 ± 0.0134
100	0.1367 ± 0.0027	0.1467 ± 0.0099	0.1399 ± 0.0085

Table 3 Mean average precision for top results of the cross-validation experiments with protein features. The † symbol indicates that the paired t-test significance analysis shows that the difference between the indicated value and the corresponding values from the other two algorithms is significant (P-value < 0.05).

NB	GP	SVM
ptngne1	ptngne1	ptngne1
ptngne2	ptngne2	ptngne2
ptngne3	rich	ptngne3
human	domain	cell
domain	motif	primary
factor	pathway	line
mutation	cysteine	interactio
transcript	binding	dependent
affinity	receptor	receptor
intermedia	limited	highly

Table 4 The words that occur the most in the 10 highest ranked sentences of each of the algorithms.

results are not significant except for when we examine just the 5 highest ranked sentences, because the MAP varies greatly from query to query. As we examine more and more top results, the difference in MAP between the algorithms reduces. In addition, we find that words that occur in the top ten sentences for each of the algorithms differ (Table 4). For example, GPs appear to rank sentences with only two proteins higher; and whilst the GP and SVM lists contain the words such as *binding* and *receptor*, which are considered highly indicative of a PPI, highly ranked NB sentences contain words that often occur in negative sentences, such as *transcript(or, ion)*.

In cross-corpora experiments PreBIND dat set continues to provide a stronger model and shows that using the correct protein features for the training data set and balanced, good quality training data can lead to an abstract-trained model that is able to identify PPI sentences with high accuracy. In the GP extension experiments we show that in the multiclass case the GPs and SVMs are equivalent; and finally, for semi-supervised classification we find that given enough training data to make a baseline model, addition of a large number of unlabelled points can improve performance.

5.1 Experiment 1: Cross-validation with plain features

The results in Table 2 show that in general the Bayesian methods are performing better on this task than the SVMs. NB has a consistently high F-score, mainly due to perfect recall. However, the precision is quite low, in turn influencing the accuracy and the AUC, both of which are significantly worse than GP and SVM across all of the cross-validation experiments. GP has the significantly highest AUC on plain features with the sentence data; however, on abstract data the difference between GPs and SVMs is not significant.

5.2 Experiment 2: Cross-validation with protein features

To assess the performance of the NE tagger and to judge which entities would be most appropriate as an alternative to the gold standard annotations, we evaluated the NE tagger against the UTexas corpus. We examined several alternative groupings of entities which were referring to proteins, and found that using *protein_molecule* (*pm*) was closest to the hand-annotations. Strict comparison, where the mark-up had to be perfectly aligned, demonstrated that the proteins were being located with high precision (P=0.7111), but lower recall (R=0.4764), leading to a fairly low F-score (F=0.5705). However, if we considered partial matches, both precision and recall were 0.12 higher, thus raising the F-score (P=0.8359, R=0.5937, and F=0.6943). Partial matches are still a good way to assess the fitness of the tagger for this problem because they would also contribute to the overall number of proteins found in a document. The main difference is that they would alter which word features were included, sometimes too many words would be considered part of the protein name, other times, too few. Partial matches occur because protein names often span several words, and there is no agreed way of annotating proteins. The HMM tagger was trained on the GENIA corpus, which has different annotation standards from the UTexas data. Furthermore, we observed that some of the proteins marked by the NE tagger that were considered false positives were in fact actual proteins that were missed by annotators. Consequently, this evaluation is just an approximation which has enabled us to chose the most appropriate automatically found entities for this data set. Cross-validation experiments using different entities mirrored the above findings by showing that *pm* entities helped with classification the most, while some even reduced F-scores below the plain-feature results. Unfortunately, what works for one data set does not translate to the other, and using *pm* entities alone decreases the F-score on the PreBIND corpus. Using them in conjunction with *protein_family_or_group* (*pfg*), provides a stable solution that does not reduce the cross-validation, but increases the cross-corpora results.

We discovered that using *pm* NE features dramatically increased the quality of sentence classification, while decreasing abstract classification. In longer abstracts, there is a larger number of protein names, making the lower recall of the NE tagging algorithm a problem; therefore, choosing a looser definition of the protein name, as reported above, increases the NE tagger recall and provides a less detrimental effect on the classification of abstracts. Substituting protein names for a place-holder string converts documents into a generalised template, increasing the prominence of interaction verbs and other surrounding words. Sentences contain more information relevant to interactions per document length, than the abstracts, and consequently these templates are more targeted.

Incidentally, the way the task is structured, whether the features are hand annotated features makes no difference, the performance increase still holds, and is consistent across all three algorithms. By comparison, Bunescu et al. (2005) used UTexas data to detect specific interactions, not just the sentences which describe them. They found that using automatically tagged NEs decreased performance of their methods compared to when they used the manually annotated protein names.

Corpus		Features	GP				SVM			
Train	Test		F	A	P	R	F	A	P	R
PB	UT	Plain	0.5425	50.7092	0.3814	0.9397	0.5674	59.4949	0.4242	0.8567
UT	PB	Plain	0.2157	44.0476	0.9767	0.1212	0.5697	60.7143	0.9342	0.4098
PB	UT	NER	**0.7031**	51.5981	0.5565	0.9544	0.6949	75.8147	0.5737	0.8811
UT	PB	NER	0.1491	41.4835	0.9655	0.0808	0.6222	63.1868	0.8922	0.4776

Table 5 Cross-corpora experiment results for GPs and SVMs. Each row shows whether the classifiers were trained or tested on the PreBIND (PB) or the UTexas (UT) corpus and what features were used (plain bag-of-words, or HMM NER tagged). The results are presented as F-score (F), accuracy (A), precision (P), and recall (R).

5.3 Experiment 3: Cross corpora evaluation

In order to test how a model created on one of these data sets would perform on a data set with a different distribution, we designed a cross-corpus experiment, the results of which are shown in Table 5. In this initial study we can observe that GPs learn from the abstract data better than from the sentence data, while for the SVMs it makes very little difference. While using PreBIND for training and UTexas for testing we find that GPs have very high recall but low precision, leading to a low F-score. The area under the ROC curve (AUC), however, is the same between the two algorithms, 0.72. Using NER features increases the AUC to 0.79 for the GP and 0.82 for the SVM, a result that is also observable in the F-scores and accuracies.

On the other hand, if we reverse the training and testing corpora, the precision-recall relationship is also inverted, and while *pm* NER features still increase the SVM F-score, the AUC for both classifiers decreases (from 0.75 to 0.70 for the GP and from 0.80 to 0.77 for the SVM). Considering the *pm+pfg* entities as proteins the PreBIND results in more effective training (as shown in Table 2), but in a smaller AUC increase (GP: 0.78, SVM: 0.79), and higher F-scores (F=0.4472, A=54.0241, P=0.9437, R=0.2930 for the GP and F=0.7420, A=29.6703, P=0.8277, R=0.6724 for the SVM). Thus, the choice of NER features that is more effective in cross validation for the training data leads to a stronger classification model, even if it is applied to data for which different settings are more applicable. This result is close to the UT cross-validation results, which means that it is possible to annotate only abstracts, but still retrieve sentences with high accuracy.

In summary, the abstract data is more conducive to training and the NER features have a positive effect given the correct choice of entities. Training with the abstracts, which have a slightly larger number of positive training data, leads to a high recall on the sentence data set, which has a low number of positive examples. The opposite holds when training on UTexas and testing on PreBIND. Finally, though the F-scores show a difference in performance between GPs and SVMs, the AUCs show that this difference is only a product of the default choice of decision boundary for the GPs.

5.4 Experiment 4: Multiclass and semi-supervised GP extensions

Multi-class and semi-supervised extensions of results indicate that GPs are particularly well suited for biomedical text classification. In the 10 fold cross-validation experiment, repeated ten times, on multiclass data NB was significantly worse than GP and SVM, while there was no difference between GPs and SVMs. The F-score for NB is 0.7169 ± 0.0023, for GPs it is 0.7649 ± 0.021 and 0l7655 ± 0.0016 for SVM. However, the GP algorithm required one single classifier for all 25 classes (Girolami and Rogers, 2006), while the one vs. one SVM multiclass application (Cawley, 2000) required $\frac{K\cdot(K-1)}{2}$, or in this case for $K = 25$ classes, 300 classifiers. Moreover, the simple bag-of-words model without named entity tagging applied here outperformed the model originally reported in Rosario and Hearst (2005). Their graphical model only achieved 60% accuracy in classifying this data, although it also performed named entity recognition at the same time.

The GP algorithm can also be extended to learn from partially-labelled data. Figure 2 shows the results from the semi-supervised learning experiment. Each of the points on the horizontal axis describes the percentage of training labels that were retained, while the vertical axis represents the difference between the accuracy of the GP trained only on the labelled portion of the data and the GP trained on both the labelled and unlabelled data. The actual accuracies for each of the experiments are different. The evaluation showed that with 1% labelled data there are too few labelled points to make accurate predictions and infer knowledge from unlabelled data. Furthermore, we found that training with 50% of the data reached the peak accuracy, and adding unlabelled data did not add any performance improvement. For this particular corpus, when 5% of the labels were available addition of 95% unlabelled documents lead to a significant gain in classification acuracy. Introduction of unlabelled training points never decreased the original classifier performance. This means that if a corpus was collected, but only a portion is annotated, the rest could still be used for training without the risk of performance decrease. Since the cost of annotation for biomedical texts is generally very high, this result shows that resources can be saved by leveraging small amounts of labelled data with large amounts of readily available unmarked text.

5.5 Algorithm performance

In this paper we used a very efficient SVM implementation programmed in C, accessed through the Matlab/Mex interface. Unoptimised Matlab code was used for the GPs, yet the difference in running time was small.

UTexas corpus required more time to run than PreBIND, which is consistent with the fact that both algorithms scale with the number of training documents and not with the number of features. Ten-fold cross-validation on UTexas data took 6 minutes and 18 seconds of user time for SVM and 7 minutes and 21 seconds for the GP. However, this measurement does not reflect the time which is needed to tune the SVM margin parameter C. In order to find the optimal value we ran 10 ten-fold cross-validation experiments requiring about 70 minutes. Had we used the RBF kernel, which would require calibration of the kernel hyperparameter θ that tuning time would be at least 700 minutes, unless experiments are run in parallel. In that same instance the tuning time for GP would be 70 minutes which would be significantly faster.

6 Conclusion

In this paper we have presented the first extensive evaluation of the Gaussian process classifier for protein interaction detection in biomedical texts. We examined three different data sets and found that GP performance is in general equal to, and depending on the data set it can be higher than, the state of the art SVM machines. Furthermore, we showed two useful variations of the basic algorithm, the multiclass and the semi-supervised, developed within the Bayesian framework. In our evaluation, one multiclass GP is equivalent to a combination of 300 binary SVM classifiers. In addition, the semi-supervised results show that the proposed classifier is capable of leveraging small amounts of labelled data with large amounts of unlabelled text, a result which has significant implications for biomedical texts, where annotation is expensive. More importantly, we have shown that the optimal choice of NE features can improve classification, not only in cross validation, but also when applying a model to data with a greatly different distribution of positive to negative examples. We believe that the flexibility of the probabilistic framework, the lack of margin parameter, and the availability of the optimised IVM algorithm make GP methods an attractive and efficient alternative to SVMs.

References

Airola, A., S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter and T. Salakoski (2008), 'All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning.', *BMC bioinformatics* **9 Suppl 11**.
URL: *http://dx.doi.org/10.1186/1471-2105-9-S11-S2*

Aizerman, A., E. M. Braverman and L. I. Rozoner (1964), 'Theoretical foundations of the potential function method in pattern recognition learning', *Automation and Remote Control* **25**, 821–837.

Albert, James H. and Siddhartha Chib (1993), 'Bayesian analysis of binary and polychotomous response data', *Journal of the American Statistical Association* **88**(422), 669.

Alias-I (2008), 'Lingpipe software package', WWW document.
URL: *http://alias-i.com/lingpipe/index.html*

Altun, Yasemin, Thomas Hofmann and Alexander J. Smola (2004), Gaussian process classification for segmenting and annotating sequences, *in* 'ICML'.

Berger, Adam L., Vincent J. Della Pietra and Stephen A. Della Pietra (1996), 'A maximum entropy approach to natural language processing', *Computational Linguistics* **22**, 39 – 71.

BIND (2007), 'Biomolecular interaction network database (BIND)', WWW document.
URL: *http://bind.ca/*

Boser, Bernhard E., Isabelle Guyon and Vladimir Vapnik (1992), A training algorithm for optimal margin classifiers, *in* 'Computational Learing Theory', pp. 144–

152.
URL: *http://citeseer.ist.psu.edu/boser92training.html*

Bunescu, R, R Ge, R J Kate, E M Marcotte, R J Mooney, A K Ramani and Y W Wong (2005), ‘Comparative experiments on learning information extractors for proteins and their interactions’, *Artif Intell Med* **33**(2), 139–155.
URL: *http://www.hubmed.org/display.cgi?uids=15811782*

Cawley, G. C. (2000), ‘MATLAB support vector machine toolbox (v0.55β)’, University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ.
URL: *http://theoval.sys.uea.ac.uk/˜gcc/svm/toolbox*

Chai, Kian Ming Adam, Hai Leong Chieu and Hwee Tou Ng (2002), Bayesian online classifiers for text classification and filtering, *in* ‘SIGIR ’02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval’, ACM Press, New York, NY, USA, pp. 97–104.

Chu, W. and Z. Ghahramani (2005*a*), ‘Gaussian processes for ordinal regression’, *Journal of Machine Learning Research* **6**, 1019–1041.

Chu, W, Z Ghahramani, F Falciani and D L Wild (2005), ‘Biomarker discovery in microarray gene expression data with gaussian processes’, *Bioinformatics* **21**(16), 3385–3393.
URL: *http://www.hubmed.org/display.cgi?uids=15937031*

Chu, Wei and Zoubin Ghahramani (2005*b*), Preference learning with gaussian processes, *in* ‘In Twenty-second International Conference on Machine Learning (ICML-2005)’.

Cohen, Aarom M. and William R Hersh (2005), ‘A survey of current work in biomedical text mining’, *Briefings in Bioinformatics* **6**(1), 51–71.

Crammer, Koby and Yoram Singer (2001), ‘On the algorithmic implementation of multiclass kernel-based vector machines’, *Journal of Machine Learning Research* **2**, 265–292.
URL: *http://jmlr.csail.mit.edu/papers/volume2/crammer01a/crammer01a.pdf*

Damoulas, T and M A Girolami (2008), ‘Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection’, *Bioinformatics* .
URL: *http://www.hubmed.org/display.cgi?uids=18378524*

Ding, C H and I Dubchak (2001), ‘Multi-class protein fold recognition using support vector machines and neural networks’, *Bioinformatics* **17**(4), 349–358.
URL: *http://www.hubmed.org/display.cgi?uids=11301304*

Donaldson, Ian, Joel Martin, Berry de Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D Bader, Katerina Michalickova, Tony Pawson and Christopher WV Hogue (2003), ‘PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine’, *BMC Bioinformatics* **4**(11).
URL: *http://www.biomedcentral.com/1471-2105/4/11*

Erkan, Gunes, Arzucan Ozgur and Dragomir R. Radev (2007), Semi-supervised classification for extracting protein interaction sentences using dependency parsing, *in* 'Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)', pp. 228–237.
URL: *http://www.aclweb.org/anthology/D/D07/D07-1024*

Girolami, Mark and Mingjun Zhong (2007), Data integration for classification problems employing gaussian process priors, *in* B.Schölkopf, J.Platt and T.Hoffman, eds, 'Advances in Neural Information Processing Systems 19', MIT Press, Cambridge, MA, pp. 465–472.

Girolami, Mark and Simon Rogers (2006), 'Variational bayesian multinomial probit regression with gaussian process priors', *Neural Computation* **18**(8), 1790–1817.

Giuliano, Claudio, Alberto Lavelli and Lorenza Romano (2006), Exploiting shallow linguistic information for relation extraction from biomedical literature, *in* 'In Proc. EACL 2006'.

Hersh, W (2005), 'Evaluation of biomedical text mining systems: lessons learned from information retrieval', *Briefings in Bioinformatics* **6**, 344–356.

Hsu, Chih-Wei and Chih-Jen Lin (2002), 'A comparison of methods for multiclass support vector machines', *IEEE Transactions on Neural Networks* **13**, 415–425.

Huang, Jin, Jingjing Lu and Charles X. Ling (2003), Comparing naive bayes, decision trees, and svm with auc and accuracy, *in* 'ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, p. 553.

Joachims, T. (1998), Text categorization with support vector machines: Learning with many relevant features, *in* 'Proceedings of the European Conference on Machine Learning', Springer.

Joachims, T. (2007), 'SVM-Light support vector machine', WWW document.
URL: *http://svmlight.joachims.org/*

Joachims, Thorsten (1999), *Advances in Kernel Methods - Support Vector Learning*, MIT-Press, chapter Making large-Scale SVM Learning Practical.

Keerthi, S. Sathiya, Olivier Chapelle and Dennis DeCoste (2006), 'Building support vector machines with reduced classifier complexity', *Journal of Machine Learning Research* **7**, 14931515.

Kim, J D, T Ohta, Y Tateisi and J Tsujii (2003), 'Genia corpus–semantically annotated corpus for bio-textmining', *Bioinformatics* **19 Suppl 1**, 180–182.
URL: *http://www.hubmed.org/display.cgi?uids=12855455*

Kote-Jarai, Z, L Matthews, A Osorio, S Shanley, I Giddings, F Moreews, I Locke, D G Evans, D Eccles, , R D Williams, M Girolami, C Campbell and R Eeles (2006), 'Accurate prediction of brca1 and brca2 heterozygous genotype using expression profiling after induced dna damage', *Clin Cancer Res* **12**(13), 3896–3901.
URL: *http://www.hubmed.org/display.cgi?uids=16818684*

Lama, N and M Girolami (2008), 'Vbmp: variational bayesian multinomial probit regression for multi-class classification in r', *Bioinformatics* **24**(1), 135–136.
URL: *http://www.hubmed.org/display.cgi?uids=18003643*

Lawrence, N. D., M. Seeger and R. Herbrich (2003), *Advances in Neural Information Processing Systems*, MIT Press, chapter Fast sparse Gaussian process methods: the informative vector machine, pp. 625–632.

Lawrence, N.D. and M.I. Jordan (2006), Gaussian processes and the null-category noise model, *in* O.Chapelle, B.Schölkopf and A.Zien, eds, 'Semi-supervised Learning', MIT Press, chapter 8.

Lawrence, Neil, John C. Platt and Michael I. Jordan (2005), Extensions of the informative vector machine, *in* J.Winkler, N. D.Lawrence and M.Niranjan, eds, 'Proceedings of the Sheffield Machine Learning Workshop', Springer-Verlag, Berlin.

Lee, K J, Y S Hwang, S Kim and H C Rim (2004), 'Biomedical named entity recognition using two-phase model based on svms', *J Biomed Inform* **37**(6), 436–447.
URL: *http://www.hubmed.org/display.cgi?uids=15542017*

Lee, Ki-Joong, Young-Sook Hwang and Hae-Chang Rim (2003), Two-phase biomedical ne recognition based on svms, *in* 'Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine', pp. 33 – 40.

Lee, Yoonkyung, Yi Lin and Grace Wahba (2004), 'Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data', *Journal of the American Statistical Association* **99**, 67–81(15).

Lewis, David D. (1998), Naive (Bayes) at forty: The independence assumption in information retrieval, *in* 'ECML '98: Proceedings of the 10th European Conference on Machine Learning', Springer-Verlag, London, UK, pp. 4–15.

Manning, Christopher D. and Hinrich Schütze (1999), *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts.
URL: *citeseer.csail.mit.edu/635422.html*

Mitchell, A L, A Divoli, J H Kim, M Hilario, I Selimas and T K Attwood (2005), 'Metis: multiple extraction techniques for informative sentences', *Bioinformatics* **21**(22), 4196–4197.
URL: *http://www.hubmed.org/display.cgi?uids=16159915*

Nigam, Kamal, Andrew McCallum and Tom Mitchell (2006), *Semi-Supervised Learning*, MIT Press, Boston, chapter Semi-supervised Text Classification Using EM.

NIH (2008), 'Statistical reports on MEDLINE/PubMed baseline data', WWW document.
URL: *http://www.nlm.nih.gov/bsd/licensee/baselinestats.html*

Opper, M. and O. Winther (1999), *Large Margin Classifiers*, MIT Press, chapter Gaussian Process Classification and SVM: Mean Field Results.

Platt, J.C. (1999), *Advances in Large Margin Classifiers*, MIT Press, chapter Probabilities for SV Machines, pp. 61–74.

Rasmussen, C. E. and C. K. I. Williams (2006), *Gaussian Processes for Machine Learning*, MIT Pres.

Rogers, S. and M. Girolami (2007), 'Multi-class semi-supervised learning with the ϵ- truncated multinomial probit gaussian process', *Journal of Machine Learning Research Workshop and Conference Proceedings* **1**, 17–32.

Rosario, Barbara and Marti Hearst (2005), Multi-way relation classification: Application to protein-protein interaction, *in* 'Proceedings of HLT-NAACL'05'.
URL: *http://biotext.berkeley.edu/papers/hlt-emnlp05-rosario.pdf*

Sculley, D. and Gabriel M. Wachman (2007), Relaxed online svms for spam filtering, *in* 'SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, New York, NY, USA, pp. 415–422.

Seeger, M. and M. I. Jordan (2004), Sparse gaussian process classification with multiple classes, Technical Report TR 661, Department of Statistics, University of California at Berkeley.
URL: *http://www.kyb.tuebingen.mpg.de/bs/people/seeger/*

Shawe-Taylor, John and Nello Cristianini (2004), *Kernel Methods for Pattern Analysis*, Cambridge University Press, New York, NY, USA.

Silva, Catarina, Ribeiro and Bernardete (2007), 'On text-based mining with active learning and background knowledge using svm', *Soft Computing* **11**(6), 519–530.
URL: *http://dx.doi.org/10.1007/s00500-006-0080-8*

Song, Min, Il-Yeol Song, Xiaohua Hu and Robert B. Allen (2005*a*), Integrating text chunking with mixture hidden markov models for effective biomedical information extraction, *in* 'International Conference on Computational Science (2)', pp. 976–984.

Song, Min, Il-Yeol Song, Xiaohua Hu and Robert B. Allen (2005*b*), 'Kxtractor: An effective biomedical information extraction technique based on mixture hidden markov models', *Transactions on Computational Systems Biology II* **2**, 68–81.

Stankovic, M., V. Moustakis and S. Stankovic (2005), Text categorization using informative vector machine, *in* 'The International Conference on Computer as a Tool, 2005. EUROCON 2005', pp. 209 – 212.

Sugiyama, Kazunari, Kenji Hatano and Shunsuke Uemura Masatoshi Yoshikawa (2003), Extracting information on protein-protein interactions from biological literature based on machine learning approaches, *in* M.Gribskov, M.Kanehis, S.Miyano and T.Takagi, eds, 'Genome Informatics 2003', Universal Academy Press, Tokyo, pp. 701–702.

Takeuchi, K and N Collier (2005), 'Bio-medical entity extraction using support vector machines', *Artif Intell Med* **33**(2), 125–137.
URL: *http://www.hubmed.org/display.cgi?uids=15811781*

Theodosiou, T, L Angelis, A Vakali and G N Thomopoulos (2007), 'Gene functional annotation by statistical analysis of biomedical articles', *Int J Med Inform* **76**(8), 601–613.
URL: *http://www.hubmed.org/display.cgi?uids=16781189*

Van Rijsbergen, C. J. (1979), *Information Retrieval, 2nd edition*, Dept. of Computer Science, University of Glasgow.
URL: *citeseer.csail.mit.edu/vanrijsbergen79information.html*

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer Verlag, New York.

Wang, Jason T. L. and Xiaoming Wu (2006), 'Kernel design for RNA classification using support vector machines', *Int. J. Data Min. Bioinformatics* **1**(1), 57–76.

Wang, T., Y. Li, K. Bontcheva, H. Cunningham and J. Wang (2006), Automatic extraction of hierarchical relations from text, *in* 'Proceedings of the Third European Semantic Web Conference (ESWC 2006)', Springer.
URL: *http://gate.ac.uk/sale/eswc06/eswc06-relation.pdf*

Xenarios, I, E Fernandez, L Salwinski, X J Duan, M J Thompson, E M Marcotte and D Eisenberg (2001), 'Dip: The database of interacting proteins: 2001 update', *Nucleic Acids Res* **29**(1), 239–241.
URL: *http://www.hubmed.org/display.cgi?uids=11125102*

Xu, Jun and Yalou Huang (2007), 'Using svm to extract acronyms from text', *Soft Comput.* **11**(4), 369–373.

Yu, W, M Clyne, S M Dolan, A Yesupriya, A Wulf, T Liu, M J Khoury and M Gwinn (2008), 'Gapscreener: An automatic tool for screening human genetic association literature in pubmed using the support vector machine technique', *BMC Bioinformatics* **9**(1), 205–205.
URL: *http://www.hubmed.org/display.cgi?uids=18430222*

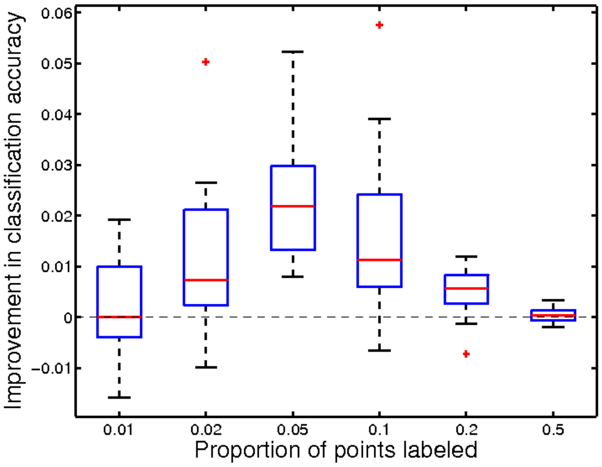

Figure 1 Semi-supervised GP results. The horizontal axis describes the proportion of the training data which is labelled. The vertical axis describes the improvement in accuracy which is achieved if unlabelled data is used in training.