Classification of Protein Interaction Sentences via Gaussian Processes

Tamara Polajnar, Simon Rogers, and Mark Girolami

University of Glasgow, Glasgow, Scotland, G12 8QQ
tamara@dcs.gla.ac.uk,
WWW home page: http://www.dcs.gla.ac.uk/inference/

Abstract. The increase in the availability of protein interaction studies in textual format coupled with the demand for easier access to the key results has lead to a need for text mining solutions. In the text processing pipeline, classification is a key step for extraction of small sections of relevant text. Consequently, for the task of locating protein-protein interaction sentences, we examine the use of a classifier which has rarely been applied to text, the Gaussian processes (GPs). GPs are a non-parametric probabilistic analogue to the more popular support vector machines (SVMs). We find that GPs outperform the SVM and naïve Bayes classifiers on binary sentence data, whilst showing equivalent performance on abstract and multiclass sentence corpora. In addition, the lack of the margin parameter, which requires costly tuning, along with the principled multiclass extensions enabled by the probabilistic framework make GPs an appealing alternative worth of further adoption.

1 Introduction

Biomedical research information is disseminated through several types of knowledge repositories. The foremost mode of academic communication are peer reviewed journals where results are evaluated and reported in a structure primarily aimed for human consumption. Alternative sources provide this information in a distilled format that is often designed for purposes of increasing the availability of particular types of results. This is typically achieved by accelerating the speed of access, cross-referencing, annotating with extra information, or restructuring the data for easier interpretation by both humans and computer programs. These resources often link the results directly to the citation in MEDLINE[1], a manually-curated publicly-available database of biomedical publication citations. Protein interactions, in particular, are a subject of many studies, the outcomes of which are stored in databases such as HPID[2], MIPS[3], and DIP[4].

[1] http://www.nlm.nih.gov/databases/databases_medline.html
[2] Human Protein Interaction Database (http://www.hpid.org/)
[3] Mammalian Protein-Protein Interaction Database (http://mips.gsf.de/proj/ppi/)
[4] Database of Interacting Proteins the Database of Interacting Proteins (http://dip.doe-mbi.ucla.edu/)

The electronic availability of these resources has lead to an increased interest in the automation of the process by which the relevant information is extracted from the original articles and entered into the specific knowledge repositories. We examine the task of locating sentences that describe protein-protein interactions (PPIs) using Gaussian processes (GPs) [35], a Bayesian analogue of the frequently applied support vector machine (SVM) [43] kernel-based classifier.

PPI detection is one of the key tasks in biomedical TM [13]. Proteins are essential parts of living organisms that, through interactions with cellular components (including other proteins) regulate many functions of the life-cycle. Approaches to PPI detection vary greatly, spanning information retrieval solutions to fully integrated parsing-based systems. For example, Chilibot is a search engine tool for finding PPIs in MEDLINE abstracts. Given a list of potential interactants, Chilibot first constructs a query that specifies combinations of the proteins, and then it processes the results to find interactions that co-occur in a sentence [9]. In a different approach, an automated pattern-based system described in [22] learns patterns from a corpus of example interaction sentences. Yet on a different track, a range of customised Bayesian methods is also available. For example, [33] present an approach that gives the likelihood that a MEDLINE abstract contains an interaction based on a dictionary of 80 discriminative words (*e.g. complex, interaction, two-hybrid, protein, domain, etc.*). [37] describe a Bayesian net model that is able to discriminate between multiple types of interaction sentences and detect protein entities at the same time. However, a non-probabilistic discriminative method has recently emerged as a highly-effective popular choice for PPI extraction. In the past ten years, SVMs have been frequently used for PPI sentence detection, where they have been proven to be highly effective [41, 18]. In particular, the kernel has been used to manipulate the input knowledge. For example, [6], [1], and [18] use structural features derived from dependency parses of the sentences with graph kernels, while [21], for example, uses kernel combinations of context-based features. In a comparative study between several classifiers, including decision trees and naïve Bayes, [23] find that SVMs perform the best on their PPI detection data set.

GPs are a Bayesian classification method analogous to the SVM that has rarely been applied to text classification; however, the probabilistic framework within which it is defined allows for elegant extensions that particularly suit TM tasks. For this reason we seek to evaluate GPs and compare them to the more frequently used SVMs and naïve Bayes (NB) [30] classifiers. Both GPs and SVMs are non-parametric, meaning that 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. Likewise the covariance function in the GP classifier corresponds to the kernel in the SVM algorithm, allowing for comparable data input and data transformations. Thus, while GPs have properties similar to SVMs [35, 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 [20, 27] and biomarker discovery in microarray data [11]. GPs have also

been applied to text classification in a few instances. Online Gaussian processes [8] and Informative Vector Machines were investigated for multiple classes on the Reuters collection in [40]. In addition, GPs and SVMs were compared for preference learning on the OHSUMED corpus [12] and an extension of GPs for sequential data such as named entities was proposed by [4].

In this article we will investigate the detection of sentences that describe PPIs in biomedical abstracts using GP classification with *bag-of-words* [30] and protein named entity (NE) features. The advantage of simpler features is that the test data does not have to be parsed or annotated in order for the model to be applied. Likewise, the model is more resilient to annotation errors. For example, in the sentence below, taken from the AImed [6] corpus, the number of interactions was correctly annotated, but the main interacting protein *IL-8* was marked in a way that is incorrect and 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**> .

In addition, we consider only PPI sentence detection and not full PPI extraction. This is a simplified view that yields a higher precision-recall balance than extraction of interacting pairs. It is a method that is not sufficient for automatic database population, but may be preferable for database curation and research purposes. The whole original sentence is returned and thus would allow the direct application of end-user relevance and quality judgments. If these judgments were logged, the system could be retrained for individual users.

2 Background

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* [2, 5] 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.

SVMs have benefited from widely available implementations, for example the C implementation SVM^{light} [24], whose algorithm uses only a subset of the training data. However, informative vector machines (IVMs) [28, 19], which are derived from GPs, now offer an analogous probabilistic alternative. A naïve implementation of SVM has a 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 [25]. 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.

2.1 Gaussian Process

Since it operates within a probabilistic framework, the GP classifier 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 $\mathbf{C}$ is 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 [19] 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 [3] enables exact Gibbs sampling or efficient variational approximations.

2.2 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, [34] 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 [35, p. 145]. On the other hand, the GP output probabilities give a more 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.

The Bayesian framework also allows for additional mathematical extensions of the basic algorithm, such as multiple classes [35, 19, 38], sequential data [4], and ordinal classes [10]. One advantage of the particular Gaussian process classifier used in this paper is its ability to effectively handle unlabelled training data (semi-supervised learning in the multiclass setting [36]). This is especially useful in text classification since there is a wealth of unlabelled documents available, but annotation can be expensive. SVMs can also be used for semi-supervised learning [39]; however difficulties often arise when multiple class data is used. There are theoretical extensions for SVMs but they are not as elegant as in the Bayesian case. For example [29] demonstrate the use of multiclass SVM on cancer microarray data; however, the implementation is $O(N^3K^3)$ [14], where K is the number of classes. Thus most applications of SVM to multiple class problems use combinations of multiple binary classifiers, for example 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, [16] use 351 SVM classifiers, per feature space, to predict 27 protein fold classes. For the same problem, [15] 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.

3 Results

3.1 Corpora and Experimental Setup

We use three main data sets. AImed is a corpus of abstracts where each individual sentence is annotated for proteins and interactions. We also examine the properties of PreBIND [17], which is only annotated for the presence of interaction within an abstract. We use these two data sets in cross validation experiments to compare the classifiers. In addition we examine if it is possible to train on the minimally annotated PreBIND data set and still classify on the sentence level. Finally, we use the BioText corpus, which is a compilation of 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* [37]. This is used to compare the algorithms in the multiclass setting.

Kernel Settings 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, 0.67 for the GP with the Gaussian and 0.80 with the cosine kernel).

Evaluation Measures 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): $precision = \frac{tp}{tp+fp}$, $recall = \frac{tp}{tp+fn}$, $F = \frac{2 \cdot precision \cdot recall}{precision+recall}$ [42]. 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. We also use the information retrieval standard mean average precision (MAP) [31] measure to assess the quality of the top ranked results from each of the classifiers.

Features Plain features were sequences of letters truncated at maximum length of 10 with stop words removed. We considered stemming and *term frequency - inverse document frequency* (*tf-idf*) [32, pp. 541–544] word weighting were examined as alternative representations, but both lead to a decrease in performance.

We examined the effect of individual proteins on classification and found that anonymisation of protein names increased performance on sentence data but decreased it for the PreBIND corpus. The features were constructed so that protein names were replaced by a placeholder string *ptngne* concatenated with the sequential number of the protein in the sentence. For example in the following sentence:

```
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 ] .
```

the extracted features are:

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

3.2 Binary Results

The results in Table 1 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 a significantly higher AUC on plain features with the sentence data; however, on abstract data the difference between GPs and SVMs is not statistically significant.

For AImed we found that using protein features increased the performance greatly regardless of whether they are gold standard annotations and automatically annotated NEs. The automatic annotation was done using the Lingpipe[5] HMM NE tagger trained on the GENIA [26] corpus. We found that considering *protein_molecule* (*pm*) features gave the highest quality of partial alignment between the annotations, which was still relatively low (P=0.8359, R=0.5937, and

[5] http://alias-i.com/lingpipe/

F=0.6943). However, in cross validation, for the PreBIND data set considering only *pm* features reduced performance, while also using *protein_family_or_group* (*pfg*) had less of a detrimental effect.

When we examined the rankings of the documents in the sentence data set with *pm* features, we found that the top results returned by the GP are significantly better than those returned by NB, as evaluated by MAP (Sect. 3.1). The variance of the MAP measure is large, so that, even though the numbers appear vastly different they are not statistically significant, except where indicated (Table 2). The quality converges as we consider more documents.

Data	Features	NB	GP	SVM
AIM	Plain	†F=**0.6785 ± 0.0080**	†F=0.6441 ± 0.0105	†F=0.6014 ± 0.0130
		†A=51.4009 ± 0.9111	†A=77.1309 ± 0.7102	†A=74.0353 ± 0.7717
		†P=0.5140 ± 0.0091	†P=0.6236 ± 0.0096	†P=0.5744 ± 0.0118
		†R=1.0000 ± 0.0000	†R=0.6679 ± 0.0160	†R=0.6336 ± 0.0194
		†AUC=0.2894 ± 0.0076	†AUC=**0.7365 ± 0.0126**	†AUC=0.7030 ± 0.0139
AIM	annotated	F=0.6915 ± 0.0108	†F=**0.7099 ± 0.0154**	F=0.6872 ± 0.0178
		†A=52.9561 ± 1.2742	†A=81.0926 ± 0.8885	†A=78.7958 ± 1.2361
		†P=0.5296 ± 0.0127	†P=0.6757 ± 0.0175	†P=0.6350 ± 0.0184
		†R=1.0000 ± 0.0000	R=0.7518 ± 0.0210	R=0.7532 ± 0.0237
		†AUC=0.2617 ± 0.0158	†AUC=**0.7898 ± 0.0102**	†AUC=0.7738± 0.0118
AIM	NER pm	†F=**0.7243 ± 0.0141**	†F=0.7117 ± 0.0087	†F=0.6611 ± 0.0141
		†A=56.9674 ± 1.7439	†A=81.4798 ± 0.3983	†A=78.1370 ± 0.7351
		†P=0.5697 ± 0.0174	†P=0.6878 ± 0.0133	†P=0.6345 ± 0.0129
		†R=1.0000 ± 0.0000	†R=0.7413 ± 0.0159	†R=0.6926 ± 0.0205
		†AUC=0.2399 ± 0.0057	†AUC=**0.7886 ± 0.0075**	†AUC=0.7500 ± 0.0097
AIM	NER pm+pfg	†F=**0.6455 ± 0.0153**	†F=0.5925 ± 0.0180	†F=0.5556 ± 0.0075
		†A=47.8439 ± 1.6409	†A=74.2450 ± 1.1850	†A=70.1948 ± 0.6240
		†P=0.4784 ± 0.0164	†P=0.5876 ± 0.0259	†P=0.5196 ± 0.0133
		†R=1.0000 ± 0.0000	R=0.6074 ± 0.0232	R=0.6052 ± 0.0198
		†AUC=0.3092 ± 0.0082	†AUC=**0.6942 ± 0.0173**	†AUC=0.6655 ± 0.0123
PB	Plain	†F=0.8350 ± 0.0095	F=**0.8621 ± 0.0114**	F=0.8547 ± 0.0091
		†A=71.7861 ± 1.4432	A=82.6097 ± 1.2976	A=81.7756 ± 1.1916
		†P=0.7179 ± 0.0144	P=0.8600 ± 0.0142	P=0.8656 ± 0.0165
		†R=1.0000 ± 0.0000	†R=0.8651 ± 0.0121	†R=0.8453 ± 0.0041
		†AUC=0.3590 ± 0.0140	AUC=**0.8069 ± 0.0157**	AUC=0.8033 ± 0.0158
PB	NER pm	†F=**0.8141 ± 0.0074**	F=0.7187 ± 0.0148	F=0.7264 ± 0.0115
		†A=68.7152 ± 1.0689	A=64.2192 ± 1.6666	A=65.1232 ± 1.0334
		†P=0.6872 ± 0.0107	P=0.7166 ± 0.0197	P=0.7205 ± 0.0119
		†R=1.0000 ± 0.0000	R=0.7251 ± 0.0188	R=0.7358 ± 0.0187
		†AUC=0.4131 ± 0.0170	AUC=0.6128 ± 0.0213	AUC=**0.6239 ± 0.0124**
PB	NER pm+pfg	F=0.8461 ± 0.0073	F=0.8535 ± 0.0099	F=**0.8575 ± 0.0130**
		†A=73.3874 ± 1.0987	A=81.4715 ± 1.1134	A=82.0506 ± 1.5046
		†P=0.7339 ± 0.0110	P=0.8530 ± 0.0131	P=0.8585 ± 0.0125
		†R=1.0000 ± 0.0000	R=0.8553 ± 0.0120	R=0.8578 ± 0.0169
		†AUC=0.3390 ± 0.0161	AUC=0.8009 ± 0.0196	AUC=**0.8163** ± 0.0217

Table 1. Results for NB, GPs, and SVMs ten-fold cross-validation experiment, repeated ten times. These 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*.

3.3 Cross-corpus evaluation

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 AImed for testing we find that GPs have

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 2. 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).

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

Table 3. Cross-corpora experiment results for GPs and SVMs. Each row shows whether the classifiers were trained or tested on the PreBIND (PB) or the AImed (AIM) 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).

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. This results in the AUC for both of the classifiers decreasing (from 0.75 to 0.70 for the GP and from 0.80 to 0.77 for the SVM), even though *pm* NER features still increase the SVM F-score. Considering the *pm+pfg* entities as proteins the PreBIND results in more effective training (as shown in Table 1), 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 when it is applied to data for which different settings are more applicable. This result is close to the AIM 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.

3.4 Multiclass Results

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 0.7655 ± 0.0016 for SVM. However, the GP algorithm required one single classifier for all 25 classes [19], while the one vs. one SVM multiclass application [7] required $\frac{K \cdot (K-1)}{2}$. For the case of $K = 25$ classes, it required 300 classifiers. Moreover, the simple bag-of-words model without named entity tagging applied here outperformed the model originally reported in [37]. Their graphical model only achieved 60% accuracy in classifying this data, although it also performed named entity recognition at the same time.

4 Conclusion

In this paper we have presented an extensive evaluation of the GP classifier for protein interaction detection in biomedical texts. Across the different experiments we can see that GPs either score higher than the SVMs, or that there is no significant difference between them. 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 demonstrated that using protein features increases classification performance regardless of whether proteins are identified manually or through automatic means. We have shown that the optimal choice of NE features can also improve cross-corpus classification even when applying a model to data with a greatly different distribution of positive to negative examples. In the multiclass setting we find the naïve Bayes classifier accuracy is much lower than that of the GPs and SVMs, whose accuracies are not significantly different. In our evaluation, one multiclass GP is equivalent to a combination of 300 binary SVM classifiers. We believe that the flexibility of the probabilistic framework, the lack of a margin parameter, and the availability of the optimised IVM algorithm are factors that make GP methods an attractive and efficient alternative to SVMs.

5 Acknowledgements

TP was funded by a Scottish Enterprise PhD studentship. SR and MG were funded by the EPSRC grant EP/E052029/1.

References

1. A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9 Suppl 11, 2008.
2. A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
3. James H. Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669, June 1993.

4. Yasemin Altun, Thomas Hofmann, and Alexander J. Smola. Gaussian process classification for segmenting and annotating sequences. In *ICML*, 2004.
5. Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.
6. R Bunescu, R Ge, R J Kate, E M Marcotte, R J Mooney, A K Ramani, and Y W Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artif Intell Med*, 33(2):139–155, Feb 2005.
7. G. C. Cawley. MATLAB support vector machine toolbox (v0.55β). University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000.
8. Kian Ming Adam Chai, Hai Leong Chieu, and Hwee Tou Ng. 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*, pages 97–104, New York, NY, USA, 2002. ACM Press.
9. H Chen and B M Sharp. Content-rich biological network constructed by mining pubmed abstracts. *BMC Bioinformatics*, 5:147–147, Oct 2004.
10. W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.
11. W Chu, Z Ghahramani, F Falciani, and D L Wild. Biomarker discovery in microarray gene expression data with gaussian processes. *Bioinformatics*, 21(16):3385–3393, Aug 2005.
12. Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *In Twenty-second International Conference on Machine Learning (ICML-2005)*, 2005.
13. Aarom M. Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):51–71, 2005.
14. Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.
15. T Damoulas and M A Girolami. Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection. *Bioinformatics*, Mar 2008.
16. C H Ding and I Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358, Apr 2001.
17. Ian Donaldson, 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. PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4(11), 2003.
18. Gunes Erkan, Arzucan Ozgur, and Dragomir R. Radev. 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)*, pages 228–237, 2007.
19. Mark Girolami and Simon Rogers. Variational bayesian multinomial probit regression with gaussian process priors. *Neural Computation*, 18(8):1790–1817, 2006.
20. Mark Girolami and Mingjun Zhong. Data integration for classification problems employing gaussian process priors. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 465–472. MIT Press, Cambridge, MA, 2007.

21. Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *In Proc. EACL 2006*, 2006.
22. Y. Hao, X. Zhu, M. Huang, and M. Li. Discovering patterns to extract protein-protein interactions from the literature: Part II. *Bioinformatics*, 21(15):3294–3300, August 2005.
23. Jin Huang, Jingjing Lu, and Charles X. Ling. Comparing naive bayes, decision trees, and svm with auc and accuracy. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 553, Washington, DC, USA, 2003. IEEE Computer Society.
24. Thorsten Joachims. *Advances in Kernel Methods - Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT-Press, 1999.
25. S. Sathiya Keerthi, Olivier Chapelle, and Dennis DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 7:14931515, 2006.
26. J D Kim, T Ohta, Y Tateisi, and J Tsujii. GENIA corpus–semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1:180–182, 2003.
27. N Lama and M Girolami. Vbmp: variational Bayesian Multinomial Probit Regression for multi-class classification in R. *Bioinformatics*, 24(1):135–136, Jan 2008.
28. Neil Lawrence, John C. Platt, and Michael I. Jordan. Extensions of the informative vector machine. In J. Winkler, N. D. Lawrence, and M. Niranjan, editors, *Proceedings of the Sheffield Machine Learning Workshop*, Berlin, 2005. Springer-Verlag.
29. Yoonkyung Lee, Yi Lin, and Grace Wahba. 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), 2004.
30. David D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, London, UK, 1998. Springer-Verlag.
31. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
32. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
33. Edward M. Marcotte, Ioannis Xenarios, and David Eisenberg. Mining literature for protein-protein interactions. *Bioinformatics*, 17:359 – 363, 2001.
34. J.C. Platt. *Advances in Large Margin Classifiers*, chapter Probabilities for SV Machines, pages 61–74. MIT Press, 1999.
35. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Pres, 2006.
36. S. Rogers and M. Girolami. 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, 2007.
37. Barbara Rosario and Marti Hearst. Multi-way relation classification: Application to protein-protein interaction. In *Proceedings of HLT-NAACL'05*, 2005.
38. M. Seeger and M. I. Jordan. Sparse gaussian process classification with multiple classes. Technical Report TR 661, Department of Statistics, University of California at Berkeley, 2004.
39. Silva, Catarina, Ribeiro, and Bernardete. On text-based mining with active learning and background knowledge using svm. *Soft Computing*, 11(6):519–530, April 2007.

40. M. Stankovic, V. Moustakis, and S. Stankovic. Text categorization using informative vector machine. In *The International Conference on Computer as a Tool, 2005. EUROCON 2005*, pages 209 – 212, 2005.
41. Kazunari Sugiyama, Kenji Hatano, and Shunsuke Uemura Masatoshi Yoshikawa. Extracting information on protein-protein interactions from biological literature based on machine learning approaches. In M. Gribskov, M. Kanehis, S. Miyano, and T. Takagi, editors, *Genome Informatics 2003*, pages 701–702. Universal Academy Press, Tokyo, 2003.
42. C. J. Van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.
43. V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.