8 Gaussian Processes and the Null-Category Noise Model

Neil D. Lawrence
Michael I. Jordan

With Gaussian process classifiers (GPC) we aim to predict the posterior probability of the class labels given an input data point, $p(y_i|x_i)$. In general we find that this posterior distribution is unaffected by unlabeled data points during learning. Support vector machines are strongly related to GPCs, but one notable difference is that the decision boundary in a SVM can be influenced by unlabeled data. The source of this discrepancy is the SVM's margin: a characteristic which is not shared with the GPC. The presence of the margin allows the support vector machine to seek low data density regions for the decision boundary, effectively allowing it to incorporate the cluster assumption (see Chapter 6). In this chapter we present the null category noise model*. A probabilistic equivalent of the margin. By combining this noise model with a GPC we are able to incorporate the cluster assumption without explicitly modeling the input data density distributions and without a special choice of kernel.*

8.1 Introduction

In this chapter we consider a Bayesian formulation of the classification problem and propose a solution to the semi-supervised learning problem within the Bayesian framework.

Bayesian methods are naturally used in the formulation of *generative* approaches to classification problems. Generative approaches explicitly model the class-conditional densities $p(x_i|y_i = 1)$ and $p(x_i|y_i = -1)$, combining these densities with the class prior probabilities $p(y_i = 1)$ and $p(y_i = -1)$ and using Bayes' theorem to form posterior predictions:

$$p(y_i = 1|x_i) = \frac{p(x_i|y_i = 1)\, p(y_i = 1)}{\sum_{y_i} p(x_i|y_i)\, p(y_i)}.$$

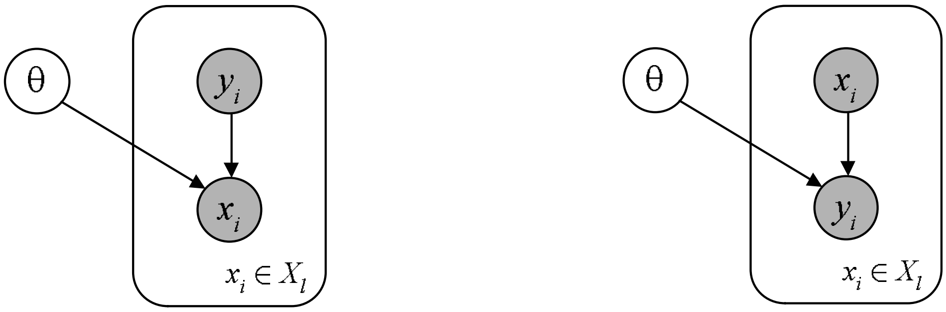

Figure 8.1 (Left) The graphical representation of a generative model. (Right) The graphical representation of a discriminative model.

In the semi-supervised setting in which a label y_i is not observed, the corresponding contribution to the likelihood is obtained by marginalizing over y_i. This yields a mixture density for x_i, a contribution to the likelihood that is handled readily within the Bayesian formulation as a standard missing-data problem. Thus semi-supervised learning is easily accommodated within a generative Bayesian framework.

Generative methods have well-known limitations in terms of prediction performance and most of the recent literature on classification has been devoted to the development of *discriminative* approaches to classification. Many of the well-known examples of discriminative classifiers are non-Bayesian, including the support vector machine (SVM), kernelized logistic regression (KLR) and boosting [Shawe-Taylor and Cristianini, 2004]. It is, however, also possible to develop discriminative classifiers within a Bayesian framework; in particular Gaussian process classifiers are discriminative classifiers that share with the SVM and KLR the use of a kernel function to form nonlinear discriminant boundaries in the input space [O'Hagan, 1992].

Bayesian approaches to discriminative classification focus on modeling the posterior probability $p(y_i|x_i)$. Naively, it would seem that semi-supervised learning is simply not accommodated within any such approach—marginalizing over y_i yields a constant contribution to the likelihood thus providing no information regarding the parameters. Non-Bayesian approaches can skirt this difficulty. In particular, statistical inference for a non-Bayesian classifier involves the specification of a "loss function" (e.g., a margin-based loss function) in addition to the specification of the model, and, as shown by several of the other chapters in this volume, loss functions can be concocted to capture information coming from unlabeled data points. A related point is that non-Bayesian approaches need not model the posterior probability and this provides additional flexibility in the design of the loss function. Bayesian approaches, on the other hand, start with the posterior probability and this appears to tie the hands of the designer, making it difficult if not impossible to develop discriminative, semi-supervised Bayesian learners.

In this chapter we show that it is in fact possible to solve the semi-supervised

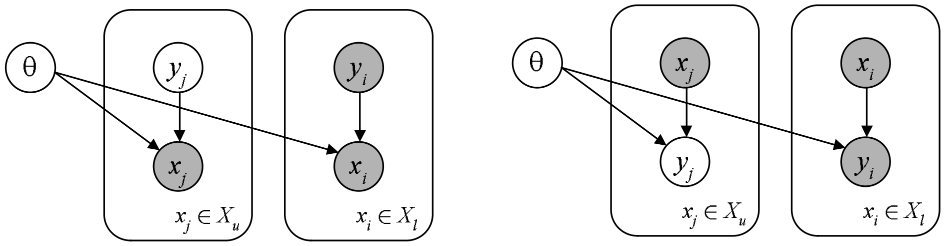

Figure 8.2 Graphical models for semi-supervised data in the generative framework (left) and the discriminative framework (right).

learning problem within a discriminative, Bayesian framework. To motivate the basic idea, let us first be more precise regarding the naive intuition referred to above by making use of graphical model representations of Bayesian classifiers. Consider first the graphical model representation of a generative classifier, shown in the left panel of Figure 8.1. In this diagram θ parameterizes the class-conditional densities. (We have omitted a node representing the class prior probabilities.) The rectangle, or "plate," captures replication. In particular, the diagram captures an assumption that the data $\{y_i\}$ are independent and identically distributed, conditional on $\{x_i\}$ and θ. Finally, shading represents conditioning; thus the graphical model in Figure 8.1 captures the standard supervised learning setting in which the labels $\{y_i\}$ are fully observed.

The right panel of Figure 8.1 presents a graphical representation of a discriminative classifier. In this case θ parameterizes the posterior probability (the meaning of θ is different in the two diagrams). Again the figure captures the standard supervised learning setting.

Let us now consider the semi-supervised learning problem. In semi-supervised learning the data X are split into a subset that is labeled, X_l, and a subset that is unlabeled, X_u. This is captured in the graphical model representations shown in Figure 8.2. Consider the model shown in the left panel for the generative case. Using the d-separation criterion for assessing conditional independence in graphical models [Pearl, 1988], we see that the parameter θ and the class label y_j are *dependent*, by virtue of the fact that their common descendant x_j is observed (shaded in the graph). This graphical motif is often referred to as a "v-structure". When two nodes point to a common descendant they are (necessarily) independent only when the descendant is *unobserved*.

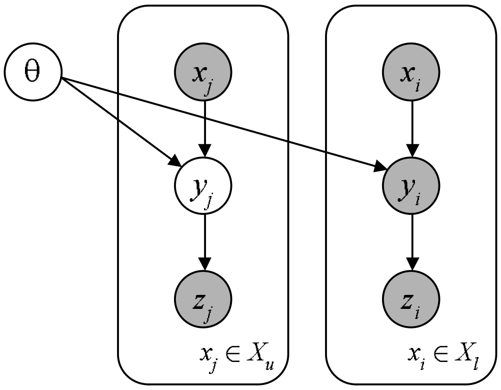

Figure 8.3 The augmented discriminative model. Even when y_j is unobserved x_j is no longer d-separated from θ because they have a common descendant which is observed. Note that for labeled examples the parameters are d-separated from the indicator variable z_j by the observation of y_j.

This holds true both when y_j is observed and when it is not. Thus both labeled and unlabeled data points will effect the Bayesian posterior distribution for θ.

Contrast this with the situation in the discriminative model (right panel of Figure 8.2). In this case the d-separation criterion shows that θ is *independent* of x_j for the unlabeled data. The observed value of x_j will not have an effect on the posterior distribution of θ when y_j is unobserved.

In the remaining sections of this chapter we will show how the discriminative model can be augmented to allow it to handle unlabeled data.

8.1.1 Augmenting the Model

In Figure 8.2 (Right) we saw how, in the discriminative approach, an unlabeled data point fails to influence the posterior for θ and thus the position of the decision boundary. This is because the unlabeled points and the parameters become d-separated (independent from one another) when the label y_j is unobserved. To restore the dependence we need to augment the model. This can be done by introducing an additional variable z_i that is a child of y_j and is always observed. As shown in Figure 8.3 this breaks the d-separation of x_j and θ and allows probabilistic dependence to flow between these variables—even when y_j is unobserved.

As a simple (and naive) example of such an augmented model, let z_j be an indicator variable that identifies whether or not the data point is labeled; i.e., z_j is taken to be 0 if the ith point is labeled and 1 if the point is unlabeled. Certainly z_j is itself always observed. Now by allowing the probability of a point being labeled to depend on its label y_i — i.e., by a particular specification of the model probability $p(z_i|y_i)$ — we can reintroduce the dependence of the parameters on the unlabeled data.

Of course this simple device does not solve the semi-supervised learning problem. Indeed, if we have no reason to believe that the probability of a data point being unlabeled is different in the two classes; i.e. if $p(z_j = 1|y_j = 1) = p(z_j = 1|y_j = -1)$, then we have $p(z_j) = p(z_j|y_j)$ and z_j is effectively decoupled from y_j, once again d-separating θ from x_j.

On the other hand, there is no need to restrict ourselves to binary indicator variables; we can be more clever about the augmentation. The remainder of the chapter develops the specific augmentation that we propose. As will be seen, our proposal is similar in spirit to the transductive SVM (see Chapter 6); we want to place the decision boundary in a region of low data density. The assumption that the inter-class regions have lower data density is known as the *cluster assumption* (see Chapter 1). We will show how an augmented model can capture the spirit of the cluster assumption—without implementing an explicit density model. This is known as low density separation.

8.2 The Noise Model

Our approach is based on the notion of a *null category*, a class for which we never observe any data. The null category can be viewed as a probabilistic interpretation of the "margin" in the SVM.[1]

To simplify our discussion of the null category noise model, we first introduce a latent process variable f_i. This variable will allow us to discuss the noise model independently of the "process model." The latent variable allows the probability of class membership to decompose as

$$p(y_i|x_i) = \int p(y_i|f_i)\, p(f_i|x_i)\, df_i.$$

where we refer to $p(y_i|f_i)$ as the *noise model* and $p(f_i|x_i)$ as the *process model*.

8.2.1 Ordered categorical models

The *null category noise model* derives from the general class of *ordered categorical models* [Agresti, 2002]. In the specific context of binary classification we will consider an ordered categorical model containing three categories.

$$p(y_i|f_i) = \begin{cases} \phi\left(-\left(f_i + \frac{a}{2}\right)\right) & \text{for} \quad y_i = -1 \\ \phi\left(f_i + \frac{a}{2}\right) - \phi\left(f_i - \frac{a}{2}\right) & \text{for} \quad y_i = 0 \\ \phi\left(f_i - \frac{a}{2}\right) & \text{for} \quad y_i = 1 \end{cases},$$

1. We are not the first to consider a probabilistic interpretation of the SVM loss function. Sollich [1999, 2000] treats the margin in terms of a "not sure" class, but this interpretation suffers from problems of normalization.

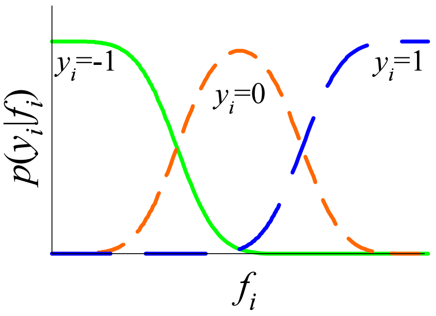

Figure 8.4 The ordered categorical noise model. The plot shows $p(y_i|f_i)$ for different values of y_i. Here we have assumed three categories.

where $\phi(x) = \int_{-\infty}^{x} \mathcal{N}(z|0,1)\,dz$ is the cumulative Gaussian distribution function and a is a parameter giving the width of category $y_i = 0$ (see Figure 8.4).

We can also express this model in an equivalent and simpler form by replacing the cumulative Gaussian distribution by a Heaviside step function,

$$H(x) = \begin{cases} 0 & \text{if} \quad x < 0 \\ 1 & \text{if} \quad x > 0 \end{cases},$$

and adding independent Gaussian noise to the process model:

$$p(y_i|f_i) = \begin{cases} H\left(-\left(f_i + \frac{1}{2}\right)\right) & \text{for} \quad y_i = -1 \\ H\left(f_i + \frac{1}{2}\right) - H\left(f_i - \frac{1}{2}\right) & \text{for} \quad y_i = 0 \\ H\left(f_i - \frac{1}{2}\right) & \text{for} \quad y_i = 1 \end{cases},$$

where we have standardized the width parameter to 1, by assuming that the overall scale is also handled by the process model.

8.2.2 The Null Category Noise Model

As stated previously, to induce a statistical dependence between the unlabeled data point, x_j, and the parameters, θ, we can augment the model with an additional variable which indicates whether the label is missing. For the null category noise model we also impose the constraint that

$$p(z_j = 1|y_j = 0) = 0; \tag{8.1}$$

in other words, a data point can not be from the category $y_j = 0$ and be unlabeled. We then parameterize the probabilities of missing labels to the other classes $p(z_i = 1|y_i = 1) = \gamma_+$ and $p(z_i = 1|y_i = -1) = \gamma_-$. For points where the label is present the latent process is updated as normal (because z_i is d-separated from θ by y_i), however, when the data point's label is missing, the posterior process is updated by,

$$p(z_i = 1|f_i) = \sum_{y_i} p(y_i|f_i)\, p(z_i = 1|y_i).$$

By marginalizing across y_i when the label is missing and otherwise using the standard likelihood, we recover the "effective likelihood function" for a single data point, $L(f_i)$. It takes one of three forms:

$$L(f_i) = \begin{cases} H\left(-\left(f_i + \frac{1}{2}\right)\right) & \text{for} \quad y_i = -1,\, z_i = 0 \\ \gamma_- H\left(-\left(f_i + \frac{1}{2}\right)\right) + \gamma_+ H\left(f_i - \frac{1}{2}\right) & \text{for} \quad z_i = 1 \\ H\left(f_i - \frac{1}{2}\right) & \text{for} \quad y_i = 1\, z_i = 0 \end{cases}.$$

The constraint imposed by (8.1) implies that an unlabeled data point never comes from the class $y_i = 0$. Since $y_i = 0$ lies between the labeled classes this is equivalent to a hard assumption that no data comes from the region around the decision boundary. We can also soften this hard assumption, if so desired, by injection of noise into the process model. If we also assume that our labeled data only comes from the classes $y_i = 1$ and $y_i = -1$ we will never obtain any evidence for data with $y_i = 0$; for this reason we refer to this category as the *null category* and the overall model as a *null category noise model* (NCNM).

8.3 Process Model and Effect of the Null Category

The noise model we have described can be used with a range of optimization frameworks. Indeed since it is a probabilistic interpretation of the SVM's margin, if we specify

$$f_i = w^{\mathrm{T}} x_i,$$

and prescribe a Gaussian prior distribution over w,

$$p(w) = \mathcal{N}(w|\mathbf{0}, \mathbf{I}),$$

where the multivariate Gaussian distribution over z with mean m and covariance Σ is given by

$$\mathcal{N}(z|m, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(z - m)^{\mathrm{T}} \Sigma^{-1} (z - m)\right),$$

then MAP solution for w is given by the linear SVM algorithm. Naturally f_i can then be 'kernelized' and the MAP solution for the model becomes equivalent to the non-linear SVM. However, in this domain the meaning of a prior distribution over w is less clear. It is often more convenient to consider a process prior over f_i. The process prior which leads to the SVM as a MAP solution is known as the Gaussian process (for two useful reviews of Gaussian processes see O'Hagan [1992], Williams [1998]). The Gaussian process prior assumes that the instantiations of the function, f_i, are jointly distributed as a zero mean Gaussian distribution with a covariance given by the kernel, K.

8.3.1 Gaussian Processes

In what remains of this chapter we will consider the use of a Gaussian process prior over f_i. The algorithms we consider update the process posterior is a sequential manner, incorporating a single data point at a time. It is therefore sufficient to consider a univariate distribution over f_i given x_i of the form

$$p(f_i|x_i) = \mathcal{N}(f_i|\mu(x_i), \varsigma(x_i)),$$

where the mean $\mu(x_i)$ and the variance $\varsigma(x_i)$ are functions of the input space. A natural consideration in this setting is the effect of our likelihood function on the distribution over f_i when incorporating a new data point. As we have already mentioned, if we observe y_i then the parameters are d-separated from z_i so the effect of the likelihood on the posterior process will be similar to that incurred in binary classification, in that the posterior will be a convolution of the step function and a Gaussian distribution. However, when the data point's label is missing the effect will depend on the mean and variance of $p(f_i|x_i)$. If this Gaussian has little mass in the null category region (i.e., the region between the classes), the posterior will be similar to the prior. However, if the Gaussian has significant mass in the null category region, the outcome may be loosely described in two ways:

1. If $p(f_i|x_i)$ "spans the likelihood," Figure 8.5 (Left), then the mass of the posterior can be apportioned to either side of the null category region, leading to a bimodal posterior. The variance of the posterior will be greater than the variance of the prior, a consequence of the fact that the effective likelihood function is not log-concave (as can be easily verified).
2. If $p(f_i|x_i)$ is "rectified by the likelihood," Figure 8.5 (Right), then the mass of the posterior will be pushed in to one side of the null category and the variance of the posterior will be smaller than the variance of the prior.

Note that for all situations when a portion of the mass of the prior distribution falls within the null category region it is pushed out to one side or both sides. The intuition behind the two situations is that in case 1, it is not clear what label the data point has, however it is clear that it shouldn't be where it currently is (in the null category). The result is that the process variance increases. In case 2 the data point is being assigned a label and the decision boundary is pushed to one side of the point so that it is classified according to the assigned label.

In an effort to demonstrate the effect of the null category, we sampled a vector $\{f_i\}_{i=1}^{500}$ from a Gaussian process with an RBF kernel. The input values $\{x_i\}_{i=1}^{500}$ were sampled uniformly from the two dimensional unit square. In the first plot

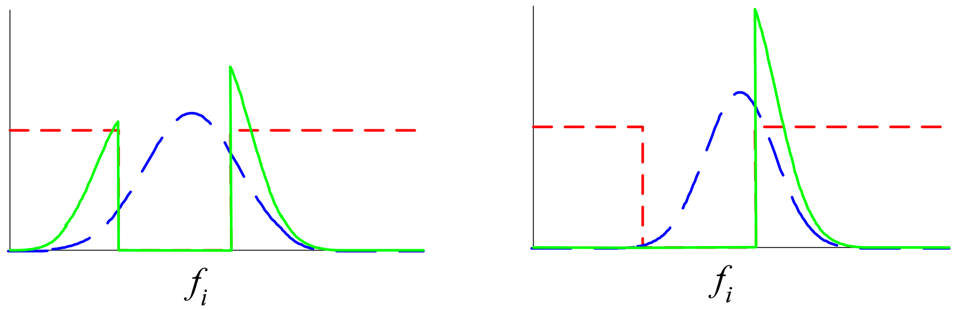

Figure 8.5 Two situations of interest. Diagrams show the prior distribution over f_i (long dashes) the effective likelihood function from the noise model when $z_i = 1$ (short dashes) and a schematic of the resulting posterior over f_i (solid line). *Left*: The posterior is bimodal and has a larger variance than the prior. *Right*: The posterior has one dominant mode and a lower variance than the prior. In both cases the process is pushed away from the null category.

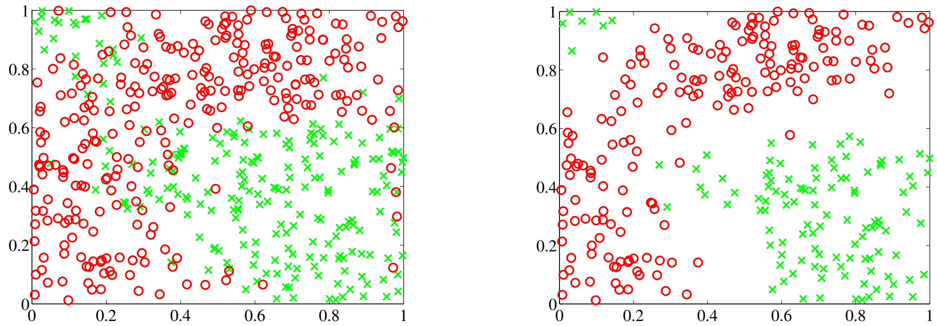

Figure 8.6 Samples from a standard Gaussian process classifier with a probit noise model (left) and a Gaussian process with the null category noise model (right). The input data was originally sampled uniformly from the unit square. The null category noise model has the effect of reducing the data density in the region of the decision boundary.

(Figure 8.6 (Left)), points were assigned to the class of $y_i = 1$ with probability $\phi(f_i)$ and were otherwise assigned a class of $y_i = -1$. In the second plot they were assigned the class $y_i = 1$ with probability $\phi\left(f_i - \frac{1}{2}\right)$ and the class $y_i = -1$ with probability $\phi\left(-f_i - \frac{1}{2}\right)$ all other points were assumed to have come from the null category and were removed. Note that this rejection of points has the effect of reducing the data density near the decision boundary.

8.4 Posterior Inference and Prediction

Broadly speaking the effects discussed above are independent of the process model: the effective likelihood will always force the latent function away from the null category. To implement our model, however, we must choose a process model and an inference method. The nature of the noise model means that it is unlikely that we will find a non-trivial process model for which inference (in terms of marginalizing f_i) will be tractable. We therefore turn to approximations which are inspired by "assumed density filtering" (ADF) methods; see, e.g., Csató [2002]. The idea in ADF is to approximate the (generally non-Gaussian) posterior with a Gaussian by matching the moments between the approximation and the true posterior. ADF has also been extended to allow each approximation to be revisited and improved as the posterior distribution evolves [Minka, 2001].

One further complication is that the 'effective likelihood' associated with the null category noise model is not log-concave, the implication of this is that the variance of the posterior process can increase when a point is included. This situation is depicted in Figure 8.5 (Left), the posterior depicted in this plot has a larger variance than the prior distribution. This increase in variance is difficult to accommodate within the approximating framework and in our implementation it was ignored.

One important advantage of the Gaussian process framework is that it is amenable to an empirical Bayesian treatment — the hyperparameters in the covariance function can be learned by optimizing the marginal likelihood. In practice, however, if the process variance is maximized in an unconstrained manner the effective width of the null category can be driven to zero, yielding a model that is equivalent to a standard binary classification noise model. The process variance controls the scale of the function. If the process variance is allowed to grow in an unconstrained manner the effective width of the null category region becomes zero, removing any effect from unlabeled data points. To prevent this from happening we regularize with an L1 penalty on the process variances (this is equivalent to placing an exponential prior on those parameters). The L1 penalty prefers smaller process variances thereby increasing the effective width of the null category region. The model therefore prefers a large null category region. This is analogous to maximizing the margin in a support vector machine.

8.4.1 Prediction with the NCNM

Once the parameters of the process model have been learned, we wish to make predictions about a new test-point x_* via the marginal distribution $p(y_*|x_*)$. For the NCNM an issue arises here: this distribution will have a non-zero probability of $y_* = 0$, a label that does not exist in either our labeled or unlabeled data. This is where the role of z_* becomes essential. The new point also has $z_* = 1$ so in reality the probability that a data point is from the positive class is given by

$$p(y_*|x_*, z_* = 1) \propto p(z_* = 1|y_*)\, p(y_*|x_*) . \tag{8.2}$$

The constraint that $p(z_* = 1|y_* = 0) = 0$ causes the predictions to be correctly normalized. So for the distribution to be correctly normalized for a test data point we must assume that we have observed $z_* = 1$.

An interesting consequence is that observing x_* will have an effect on the process

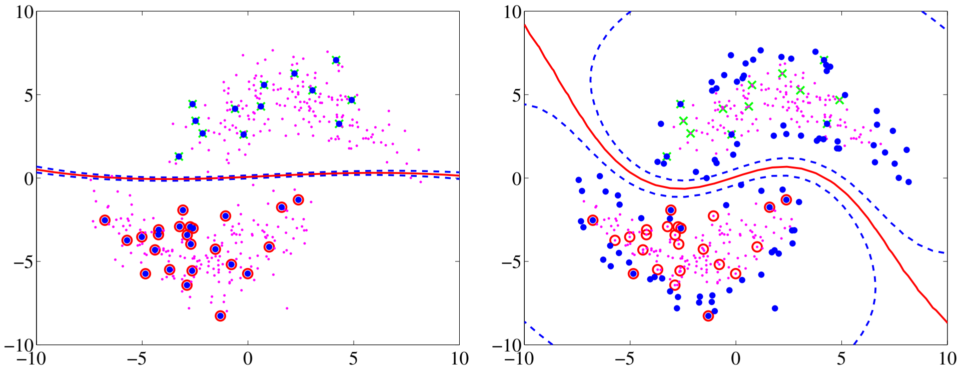

Figure 8.7 Results from the toy problem. There are 400 points, which are labeled with probability 0.1. Labeled data-points are shown as circles and crosses. Data-points in the active set are shown as large dots. All other data-points are shown as small dots. *Left*: Learning on the labeled data only with the IVM algorithm. All labeled points are used in the active set. *Right*: Learning on the labeled and unlabeled data with the NCNM. There are 100 points in the active set. In both plots decision boundaries are shown as a solid line; dotted lines represent contours within 0.5 of the decision boundary (for the NCNM this is the edge of the null category).

model. This is contrary to the standard Gaussian process setup (see, e.g., Williams [1998]) in which the predictive distribution depends only on the labeled training data and the location of the test point x_*. In the NCNM the entire process model $p(f_*|x_*)$ should be updated after the observation of x_*. This is not a particular disadvantage of the null category noise model, it is an inevitable consequence of any method that allows unlabeled data to affect the location of the decision boundary—a consequence the probabilistic framework makes explicit. In practice however we may disregard such considerations and make (possibly suboptimal) predictions of the class labels according to (8.2) without updating the location of the decision boundary.

8.5 Results

Sparse representations of the data set are essential for speeding up the process of learning. We made use of the informative vector machine (IVM) approach to greedily select an active set according to information-theoretic criteria. The informative vector machine is an approximation to a full Gaussian Process which is competitive with the support vector machine in terms of speed and accuracy. The IVM also enables efficient learning of kernel hyperparameters, and we made use of this feature in all of our experiments.

In all our experiments we used a kernel of the form

$$k_{nm} = \theta_2 \exp\left(-\theta_1 \left(\mathbf{x}_n - \mathbf{x}_m\right)^{\mathrm{T}} \left(\mathbf{x}_n - \mathbf{x}_m\right)\right) + \theta_3 \delta_{nm},$$

where δ_{nm} is the Kronecker delta function. The IVM algorithm selects an active set, and the parameters of the kernel were learned by performing type-II maximum likelihood over the active set. Since active set selection causes the marginalized likelihood to fluctuate it cannot be used to monitor convergence, we therefore simply iterated fifteen times between active set selection and kernel parameter optimization. The parameters of the noise model, $\{\gamma_+, \gamma_-\}$ can also be optimized, but note that if we constrain $\gamma_+ = \gamma_- = \gamma$ then the likelihood is maximized by setting γ to the proportion of the training set that is unlabeled.

We first considered an illustrative toy problem to demonstrate the capabilities of our model. We generated two-dimensional data in which two class-conditional densities interlock. There were 400 points in the original data set. Each point was labeled with probability 0.1, leading to 37 labeled points. First a standard IVM classifier was trained on the labeled data only (Figure 8.7, Left). We then used the null category approach to train a classifier that incorporates the unlabeled data. As shown in Figure 8.7 (Right), the resulting decision boundary finds a region of low data density and more accurately reflects the underlying data distribution.

8.5.1 USPS Digits

We next considered the null category noise model for learning of USPS handwritten digit data set. This data set is fully labeled, but we can ignore a proportion of the labels and treat the data set as a semi-supervised task. In the experiments that followed we used an RBF kernel with a linear component. We ran each experiment ten times, randomly selecting the data points that were labeled. The fraction of labeled points, r, was varied between 0.01 and 0.25. Each digit was treated as a separate 'one against the others' binary classification class. We also summarized these binary classification tasks with an overall error rate by allocating each test data point to the class with the highest probability. In the first of our experiments, we attempted to learn the parameters of the kernel by maximizing the IVM's approximation to the marginal likelihood. The results are summarized in Table 8.1.

It is immediately apparent that for values of r below 0.1 a sensible decision boundary is not found for many of the binary classification tasks. At first sight, this reflects badly on the approach: many semi-supervised learning algorithms give excellent performance even when the proportion of unlabeled data is so low. However here it must be borne in mind that the algorithm has the additional burden of learning the kernel parameters. Most other approaches do not have this capability and therefore results are typically reported for a given set of kernel parameters. For this reason we also undertook experiments using kernel parameters that were precomputed by an IVM trained on the fully labeled data set. These results are summarized in Table 8.2. As expected these results are much better in the range where $r < 0.1$. With the exception of the digit 2 at $r = 0.01$ a sensible decision boundary was learned for at least one of the runs even when $r = 0.01$.

We would certainly expect the results to be better in this second experiment as we are providing more information (in terms of precomputed kernel parameters)

r	**0**	**1**	**2**	**3**	**4**	
0.010	18 ± 0.0	8.0 ± 6.5	9.9 ± 0.0	8.3 ± 0.0	10 ± 0.0	
0.025	11 ± 8.8	0.98 ± 0.1	9.9 ± 0.0	6.5 ± 2.4	10 ± 0.0	
0.050	1.7 ± 0.2	1.0 ± 0.1	3.7 ± 0.4	5.4 ± 2.7	7.4 ± 3.5	
0.10	1.7 ± 0.1	0.95 ± 0.1	3.2 ± 0.2	3.2 ± 0.3	3.3 ± 0.3	
0.25	1.6 ± 0.2	0.97 ± 0.1	2.5 ± 0.2	2.9 ± 0.2	2.8 ± 0.1	
r	**5**	**6**	**7**	**8**	**9**	**Overall**
0.010	8.0 ± 0.0	8.5 ± 0.0	7.3 ± 0.0	8.3 ± 0.0	8.8 ± 0.0	83 ± 7.3
0.025	8.0 ± 0.0	8.5 ± 0.0	7.3 ± 0.0	8.3 ± 0.0	8.8 ± 0.0	64 ± 5.0
0.05	7.1 ± 1.9	1.7 ± 0.2	7.3 ± 0.0	7.4 ± 1.9	7.6 ± 2.7	33 ± 7.2
0.1	3.0 ± 0.3	1.5 ± 0.1	1.3 ± 0.1	3.4 ± 0.2	2.0 ± 0.3	7.7 ± 0.2
0.25	2.4 ± 0.2	1.3 ± 0.2	1.2 ± 0.1	2.6 ± 0.3	1.6 ± 0.2	6.4 ± 0.2

Table 8.1 Table of results for semi-supervised learning on the USPS digit data. For these results the model learned the kernel parameters. We give the results for the individual binary classification tasks and the overall error computed from the combined classifiers. Each result is summarized by the mean and standard deviation of the percent classification error across ten runs with different random seeds.

r	**0**	**1**	**2**	**3**	**4**	
0.010	3.2 ± 5.2	13 ± 14	9.9 ± 0.0	3.1 ± 0.2	8.3 ± 2.6	
0.025	1.5 ± 0.2	1.5 ± 0.9	5.2 ± 2.0	2.9 ± 0.2	4.4 ± 2.1	
0.050	1.5 ± 0.2	1.2 ± 0.2	3.4 ± 0.4	2.9 ± 0.1	3.3 ± 0.2	
0.10	1.5 ± 0.1	1.2 ± 0.1	2.8 ± 0.2	2.8 ± 0.2	3.0 ± 0.2	
0.25	1.4 ± 0.2	1.3 ± 0.2	2.4 ± 0.2	2.6 ± 0.2	2.8 ± 0.2	
r	**5**	**6**	**7**	**8**	**9**	**Overall**
0.010	7.5 ± 1.0	7.7 ± 8.5	12 ± 17	7.5 ± 1.2	35 ± 23	42 ± 10
0.025	5.0 ± 1.3	1.6 ± 0.2	1.9 ± 1.9	4.3 ± 0.5	9.9 ± 8.5	14 ± 6.1
0.050	3.6 ± 0.6	1.5 ± 0.1	1.3 ± 0.1	4.1 ± 0.4	2.6 ± 1.3	8.4 ± 0.7
0.10	2.8 ± 0.2	1.3 ± 0.1	1.3 ± 0.1	3.5 ± 0.3	2.0 ± 0.2	7.2 ± 0.5
0.25	2.3 ± 0.2	1.2 ± 0.1	1.2 ± 0.1	2.7 ± 0.2	1.6 ± 0.2	6.1 ± 0.4

Table 8.2 Table of results for semi-supervised learning on the USPS digit data. For these results the model was given the kernel parameters learned by the IVM on the standard fully labeled data. We give the results for the individual binary classification tasks and the overall error computed from the combined classifiers.

to the model. However it is very encouraging that for $r > 0.1$ the results of both experiments are similar.

8.6 Discussion

We motivated the work in this chapter through a brief review of the discriminative and the generative approaches to building classifiers from the perspective of probabilistic models. In general the discriminative approach is preferred, but in the particular context of semi-supervised learning it is not initially obvious how to take advantage of the unlabeled data: the parameters of the model are d-separated from the input position of the unlabeled data point. We showed how it was possible to restore the interaction between the parameters and the unlabeled data by augmenting our probabilistic model with an additional variable. Then, through inspiration from the support vector machine, we motivated a novel noise model: the null category noise model, through which we were able to take advantage of unlabeled data within the discriminative framework.

Our approach provides a practical approach to performing semi-supervised learning with Gaussian process without the requirement of designing bespoke kernels.

Code for recreating our experiments is available from `http://www.dcs.shef.ac.uk/~neil/ncnm`.