Regularization and transformation: CSE802 Project

Jonny Dowdall, James Peterkin II, Eric Alan Wayman

Submitted May 4, 2018

1 Introduction

1.1 Motivation

The fact that the performance of classifiers can be improved through methods that 1) impose some form of regularization on the estimated parameters, or 2) exploit useful transformations of the feature space, suggests that understanding of the intracies and interrelationships between these techniques is an important part of any practitioner's toolbox in the field of pattern recognition and machine learning. Our project seeks to build such an understanding and to test that understanding against the application of the techniques to real-world datasets.

The basic idea of regularization is that when fitted to a particular training data set models may learn the idiosyncracies of that data set (overfitting) and that will be reflected in poorer performance on an unseen test data set compared to the performance of the average of a series of models fit to various training sets. Regularization simulates this procedure by restricting the magnitude of components of the parameter vector, forcing the fitted model to be more "smooth" than if the parameter vector were unrestricted. Overfitting can also be addressed through sparsity-encouraging regularization (ℓ_1 regularization, which increases the "signal to noise ratio" by encouraging the parameter coefficients of features that do not contribute to reducing model error to go to zero).

Underfitting occurs when the model is too simple for the data. Since the models in this project produce linear decision boundaries, this will occur when features are not linearly separable in the original feature space. Transforming the data to a higher-dimensional feature space, in this project demonstrated by the use of the RBF kernel, may allow the data to become linearly separable in that feature space. Through the use of the kernel trick, the model can be fit in the higher-dimensional space through performing calculations in the original feature space, and the resulting decision boundary visualized in the original feature space.

Overfitting can also be addressed through fitting the data using a subset of the training points, and by choosing a linear boundary that separates the data points with as large a "margin", or distance from the closest points to the boundary, as possible. Both of these qualities are characteristics of the Support Vector Machine (SVM).

Another classifier, logistic regression, can be used in conjunction with the ℓ_1 and ℓ_2 norms for regularization, as well as with transformed data (kernelized logistic regression). The details of all these techniques will be explained in sections to follow.

1.2 Literature review

Regularization was first proposed by Tychonoff (Theodoridis 2015, 72) in 1977 (Tychonoff and Arsenin 1977). ℓ_1 and ℓ_2 regularization are common techniques described in many machine learning textbooks, for example Murphy 2015.

Logistic regression, the first of the two classifiers that were used in this project, was invented by David Cox in 1958 (Cox 1958).

According to Wikipedia, kernel classifiers were likely first mentioned in the 1960s (Aizerman et al. 1964) and gained widespread attention due to the introduction of the support vector machine (SVM) in the 1990s due to the SVM's competitive performance on tasks such as handwriting recognition. This SVM, which uses the "kernel trick" to be described below was invented by Boser et al. in 1992 (Boser et. al 1992).

The usage of kernels with logistic regression is demonstrated in Zhu and Hastie (2002).

The techniques explained and used in this project are widely used in pattern classification, and the exploration of their properties has been the subject of much research. Some examples of SVMs are described in Abe (2010). Keerthi et al. (2003) describe the performance of SVMs as the hyperparameter values become very large and very small. Lee et al. (2006) describe an efficient algorithm for fitting logistic regression with ℓ_1 regularization. Liu et al. (2011) propose an estimation procedure for the SVM with RBF kernel.

1.3 Problem statement

To explain how techniques of regularization and data transformation can be used in conjunction with the binary classifers logistic regression and SVM, use this understanding to hypothesize regarding the performance of the various techniques on datasets with differing characteristics, and test these hypotheses on real-world datsets. Our hypotheses are as follows:

On datasets which are linearly separable, if most of the features are useful towards classification we expect logistic regression with ℓ_2 regularization and SVM with a linear kernel to perform well. If only a few of the features are useful towards classification on such a dataset, we expect logistic regression with ℓ_1 regularization to perform better.

On datasets which are not linearly separable, if most of the features are useful towards classification we expect the SVM with an RBF kernel to perform well. If only a few of the features are useful towards classification on such a dataset, we expect logistic regression with an RBF kernel to perform better.

2 Approach

2.1 Logistic regression with ℓ_1 and ℓ_2 regularization

Logistic regression is a discriminative classifier. It corresponds to a binary classification model:

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathrm{Ber}(y|\mathrm{sigm}(\boldsymbol{w}^T \boldsymbol{x}))$$

(Murphy 2012, 245) where sigm is the sigmoid function. If the possible values of y are either -1 or $+1$, then $p(y = 1) = 1/(1 + \exp(-\boldsymbol{w}^T \boldsymbol{x}))$ and $p(y = -1) = 1/(1 + \exp(\boldsymbol{w}^T \boldsymbol{x}))$. We minimize the error by maximizing the negative log-likelihood:

$$NLL(\boldsymbol{w}) = \log(1 + \exp(-y_i \boldsymbol{w}^T \boldsymbol{x}))$$

(Murphy 2012, 245). There is not a closed-form solution for the MLE of $\boldsymbol{w}$, so it must be estimated by an optimization algorithm. However, we must often design constraints to prevent the parameters from overfitting the training data and losing generality.

2.1.1 The effects of ℓ_1 and ℓ_2 regularization

When the ℓ_2 regularization term is included, maximizing the NLL function with respect to $\boldsymbol{w}$ and λ tries to reduce the norm of $\boldsymbol{w}$ (the vector of parameters) while at the same time minimizing the error given by the log-likelihood cost function (maximizing the negative of this function). This helps prevent overfitting: by restricting the ℓ_2 norm of $\boldsymbol{w}$, the "complexity" of the model is restricted, so it is prevented from "learning too much about the idiosyncrasies of the specific training data set" (Theodoridis 2015, 74).

If only a few features contain significant information and there are a large number of features, the "true" model generating the data will have the coefficients of most components of $\boldsymbol{w}$ equal to zero. Therefore it

The following figure (Figure 1, taken from Theodoridis 2015, 406, Figure 9.2) shows the relationship between a given component θ of the parameter vector $\boldsymbol{\theta}$ (what we call $\boldsymbol{w}$) and its contribution to $\|\boldsymbol{\theta}\|_p$, $|\theta|^p$, for given levels of p. For ℓ_p norms with $p \geq 1$, components θ with larger $|\theta|^p$ give a larger contribution to the norm, so assuming for example's sake that two components θ_1 and θ_2 have the same effect on the fit of the model and $|\theta_1|^p > |\theta_2|^p > 1$, the minimization will try to reduce the size of θ_1 more than θ_2. Conversely, for $p > 1$, any θ_j with $|\theta_j|^p < 1$ will not have its size reduced very much at all, irrespective of the amount to which it contributes to minimizing the error of the model.

2.1.2 Applying regularization to logistic regression

ℓ_1 regularization is achieved by adding the term $\lambda \|\boldsymbol{w}\|_1{}^2$ where $\|\boldsymbol{w}\|_1 = \sum_{i=1}^{c} |\boldsymbol{w}_i|$ (Theodoridis 2015, 404), so

$$NLL(\boldsymbol{w}, \lambda) = \log(1 + \exp(-y_i \boldsymbol{w}^T \boldsymbol{x})) + \lambda \|\boldsymbol{w}\|_1$$

ℓ_2 regularization is achieved by adding the term $\frac{\lambda}{2} \|w\|_2{}^2$ to $NLL(\boldsymbol{w})$ above, giving

$$NLL(\boldsymbol{w}, \lambda) = \log(1 + \exp(-y_i \boldsymbol{w}^T \boldsymbol{x})) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2{}^2$$

However, for $p = 1$, even components θ_j with $|\theta_j|^1 < 1$ will have the regularization applied to them. Therefore irrespective of the size of a true θ_j, the regularization will force θ_j to 0 if it does not contribute to minimizing model error.

(The above discussion was based Theodoridis 2015, 406-407)

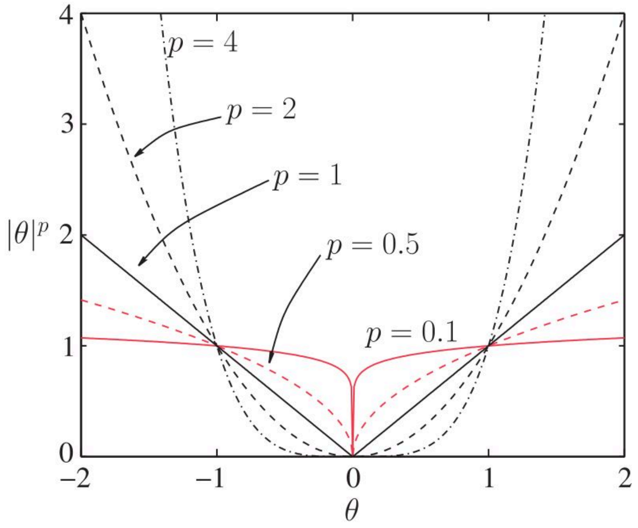

Figure 1: The approximate effect of the ℓ_p norm on a given component of the parameter vector

2.2 Kernels: linear vs RBF

Kernels are commonly used to model similarities over pairs of data points. A Mercer kernel is a kernel whose Gram matrix

$$\boldsymbol{K} = \begin{pmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ & \vdots & \\ \kappa(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix}$$

is positive semi-definite for any set of inputs $\{\boldsymbol{x}_i\}_{i=1}^N$ (Murphy 2012, 481). For any Mercer kernel there exists a function $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^D$ for which then $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x})$. Note that D can be infinite, as explained in the section "SVM and RBF kernel relationship explanation."

In this project we use two kernels, linear kernels and the RBF kernel, both of which are Mercer kernels. The kernels will be used in this project as transformations of data to be input to classifiers which produce a linear decision boundary (if transformed data is input to a classifier, the resulting decision boudary will be linear in that transformed space).

Note that usually it is hard to derive the feature vector $\boldsymbol{\phi}(\boldsymbol{x})$ from a Kernel $\kappa(\boldsymbol{x}, \boldsymbol{x}')$, but the reverse is not difficult for a Mercer kernel since $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})$.

The linear kernel is $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$, which corresponds to the case where $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$, so $\boldsymbol{\phi}(\boldsymbol{x})$ takes points in $\mathcal{X}$ to $\mathcal{X}$. This kernel is useful in the case where the decision boundary is linear in the original feature space, so transforming them to a higher-dimensional feature space is not necessary (Murphy 2012, 482).

The RBF kernel is defined as follows:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\gamma \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|\right)$$

As noted above, the D in $\boldsymbol{\phi}(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}^D$ is infinite in the case of the RBF kernel. To understand the transformation, following Abu-Mostafa et al. (8-37), let $\gamma = 1$ and $\boldsymbol{x}$ be a scalar. Then

$$\begin{aligned} K(x, x') &= \exp\left(-\left\|x - x'\right\|^2\right) \\ &= \exp\left(-(x)^2\right) \cdot \exp\left(2xx'\right) \cdot \exp(-\left(x'\right)^2) \\ &= \exp\left(-(x)^2\right) \cdot \left(\sum_{k=0}^{\infty} \frac{2^k (x)^k \left(x'\right)^k}{k!}\right) \cdot \exp\left(-\left(x'\right)^2\right) \end{aligned}$$

Defining

$$\boldsymbol{\phi}(x) = \exp(-x^2) \cdot \left(1, \sqrt{\frac{2^1}{1!}}x, \sqrt{\frac{2^1}{2!}}x^2, \sqrt{\frac{2^1}{3!}}x^3, \dots\right)$$

we see that $K(x, x') = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x)$. The right hand side is an inner product in an infinite-dimensional feature space, which shows that the D in the range of K can be infinite.

2.2.1 The "kernel trick"

If it is difficult to compute $\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x})$, instead we can compute $K(\boldsymbol{x}, \boldsymbol{x}')$ in the original $\mathcal{X}$ space since the results are equal. For the kernels used in this project, this is useful for the RBF kernel, as exact calculation of $\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x})$ in the range space of $\boldsymbol{\phi}$ is impossible.

2.3 SVMs

The SVM is a classifier that incorporates sparsity of data points (as opposed to features) into its loss function (Murphy 2012, 497). SVMs for classification use a loss function called hinge loss, which is of the form $L_{\text{hinge}}(y, \eta) = \max(0, 1 - y\eta) = (1 - y\eta)_+$ where $\eta = f(\boldsymbol{x})$ is the "confidence" (not necessarily a probability) in choosing label $y = 1$ (Murphy 2012, 499). The objective function is

$$\min_{\boldsymbol{w}, w_0} \frac{1}{2} \left\|\boldsymbol{w}\right\|_2{}^2 + C \sum_{i=1}^{N} (1 - y_i f(\boldsymbol{x}_i))_+$$

This is non-differentiable, but by introducing slack variables, the minimization problem can be transformed to one solvable by quadratic programming (Murphy 2012, 499).

2.3.1 Generalization and the large-margin principle

The minimization problem mentioned in the previous paragram can be obtained through a different approach, namely maximizing the size of the margin $f(\boldsymbol{x}) / \left\|\boldsymbol{w}\right\|_2$. This approach also depends on the introduction of slack variables which allows the problem to handle certain cases. The resulting objective function is the same as the approach from minimizing the hinge loss function.

The importance of the large-margin is that it helps the model's generalization performance (Theodoridis 2015, 550). An intuitive way to see this is by Figure 2 (this is Figure 14.11 from Murphy 2012, 500).

2.3.2 Generalization and support vectors

The solution for the weights for the SVM has the form $\widehat{\boldsymbol{w}} = \sum_i \alpha_i \boldsymbol{x}_i$ where $\boldsymbol{\alpha}$ has many entries equal to 0; the $\boldsymbol{x}_i$ corresponding to non-zero α_i are called support vectors. Since the parameter vector for the fitted SVM depends only on a subset of data points, this helps model generalizability (Theodoridis and Koutroumbas 2009, 206).

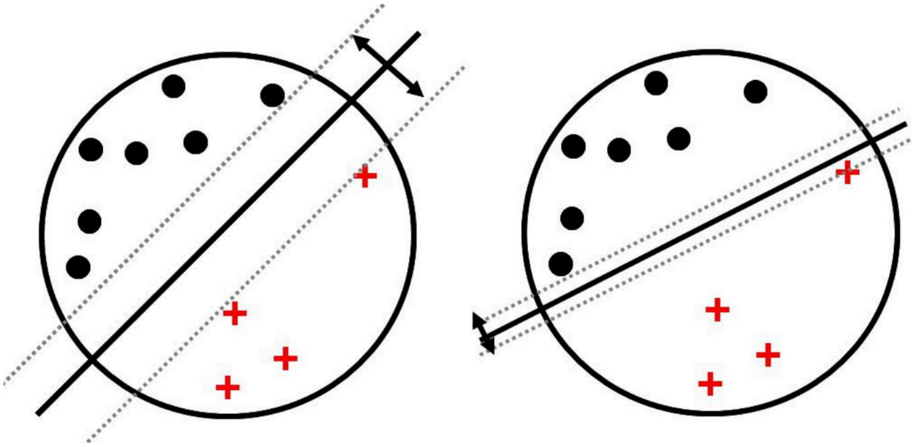

Figure 2: Visualization of the large margin principle

The SVM is used in this project with both the linear and RBF kernels.

2.4 SVM and logistic regression with the RBF kernel: a close relationship

In this section, we explain the effects of using logistic regression on data that has been transformed with the RBF kernel, and how this relates to the case where an SVM is used with such transformed data.

The optimal $f(\boldsymbol{x})$ in fitting an SVM is of the form $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i')$ (Zhu and Hastie 2002, 186). Also since the negative log-likelihood (NLL) for logistic regression has a similar shape to the NLL of the SVM, replacing the NLL of the SVM with the NLL of the logistic regression gives roughly the same solution (Zhu and Hastie 2002, 186). Then for a Mercer kernel, the interpretation of the probability $p(\boldsymbol{x})$ (which equals $P(y = 1|\boldsymbol{X} = \boldsymbol{x})$, Lin 2002) is

$$
\begin{aligned}
p(\boldsymbol{x}) &= \frac{e^{f(\boldsymbol{x})}}{1+e^{f(\boldsymbol{x})}} = \frac{1}{1+\exp(-f(\boldsymbol{x}))} \\
&= \frac{1}{1+\exp(-\sum_{i=1}^{n}\alpha_i K(\boldsymbol{x},\boldsymbol{x}_i'))} && \text{plugging in the optimal solution} \\
&= \frac{1}{1+\exp(-\sum_{i=1}^{n}\alpha_i \boldsymbol{\phi}(\boldsymbol{x}_i)^T\boldsymbol{\phi}(\boldsymbol{x}))} && \text{using the kernel trick} \\
&= \frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}))}
\end{aligned}
$$

where the last step is by defining $\boldsymbol{w} = \sum_i \alpha_i \boldsymbol{\phi}(\boldsymbol{x}_i)$ is the weighted sum of transformed support vectors. The last two steps here were taken from Guestrin (2007). This implies that the kernel trick can be used to run logistic regression on data that has been transformed to an infinite-dimensional feature space using the $\boldsymbol{\phi}$ corresponding to the RBF kernel.

2.5 Summary of model fitting strategies and data transformations

The following table summarizes the combinations of model fitting strategies and data transformations used in this project. Each column indicates a different model fitting strategy (used in conjunction, of course, with minimizing model error as represented by a loss function), while each row indicates kernel, in other words, a feature transformation. Each cell indicates the classifier that was used in conjuction with the fitting strategy and data transformation. Note that the model used for any particular combination is deterministic: in other words, the desired model fitting strategy and data transformation indicate a model choice.

	Simple loss function	Loss function with ℓ_1 regularization	Loss function with ℓ_2 regularization	Few data points & large margin
Linear		Logistic regression	Logistic regression	SVM
RBF	Logistic regression			SVM

Table 1: Models used with different fitting strategies and feature transformations

3 Datasets

The following subsections describe the datasets used, and they are followed by a summary table.

3.1 Breast Cancer Dataset

1. Wisconsin Diagnostic Breast Cancer (WDBC)
2. Source Information

  - Creators:
    - Dr. William H. Wolberg, General Surgery Dept., University of Wisconsin, Clinical Sciences Center, Madison, WI 53792. wolberg@eagle.surgery.wisc.edu
    - W. Nick Street, Computer Sciences Dept., University of Wisconsin, 1210 West Dayton St., Madison, WI 53706. street@cs.wisc.edu 608-262-6619
    - Olvi L. Mangasarian, Computer Sciences Dept., University of Wisconsin, 1210 West Dayton St., Madison, WI 53706. olvi@cs.wisc.edu
  - Donor: Nick Street
  - Date: November 1995

3. Relevant information

   - Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. A few of the images can be found at http://www.cs.wisc.edu/ street/images/

4. Number of instances: 569

5. Number of attributes: 32 (ID, diagnosis, 30 real-valued input features)

6. Attribute information:

   (a) ID number
   (b) Diagnosis (M = malignant, B = benign)
   (c) 3-32)
      - Ten real-valued features are computed for each cell nucleus such as radius, texture, etc.
   (d) Class distribution: 357 benign, 212 malignant

7. W.N. Street et al. (1993)

3.2 Letter Dataset

1. Letter Image Recognition Data

2. Source Information

   - Creator: David J. Slate
   - Odesta Corporation; 1890 Maple Ave; Suite 115; Evanston, IL 60201
   - Donor: David J. Slate (dave@math.nwu.edu) (708) 491-3867
   - Date: January, 1991

3. Past Usage:

   - P.W. Frey and D. J. Slate (Machine Learning Vol 6 #2 March 91):
     - ”Letter Recognition Using Holland-style Adaptive Classifiers”.

4. Number of instances: 20000.

- However, we only used 1543 since we are testing binary classifiers and we chose the letters E and F, whose counts are listed below.
- The data was split 60/40 for a training/testing set.

5. Number of Attributes: 17 (Letter category and 16 numeric features)

6. Attribute Information:

   - lettr capital letter (26 values from A to Z)
   - x-box horizontal position of box (integer)
   - y-box vertical position of box (integer)
   - width width of box (integer)
   - high height of box (integer)
   - onpix total # on pixels (integer)
   - x-bar mean x of on pixels in box (integer)
   - y-bar mean y of on pixels in box (integer)
   - x2bar mean x variance (integer)
   - y2bar mean y variance (integer)
   - xybar mean x y correlation (integer)
   - x2ybr mean of x * x * y (integer)
   - xy2br mean of x * y * y (integer)
   - x-ege mean edge count left to right (integer)
   - xegvy correlation of x-ege with y (integer)
   - y-ege mean edge count bottom to top (integer)
   - yegvx correlation of y-ege with x (integer)

7. Class Distribution:

   - 789-A 766-B 736-C 805-D **768-E 775-F** 773-G
   - 734-H 755-I 747-J 739-K 761-L 792-M 783-N
   - 753-O 803-P 783-Q 758-R 748-S 796-T 813-U
   - 764-V 752-W 787-X 786-Y 734-Z

8. P.W. Fray and D.J. Slate (1991)

3.3 Leukemia Dataset

1. This dataset set was built to help predict new classes of cancer. The original dataset was built from 38 leukemia bone marrow samples. The bone marrow samples were obtained from acute leukemia patients at diagnosis.

2. Number of samples: 38

3. Number of features: 7129

4. Number of classes: 2. Acute myeloid leukemia (AML) & Acute lymphoblastic leukemia (ALL)

5. Training/Testing: The data was split 60/40

6. Golub et al. (1999)

3.4 MNIST Dataset

1. The MNIST database is a large collection of handwritten digits from 0 to 9. This dataset is a subset of the larger dataset NIST (Grother, 1995), which contains 62 classes of handwritten characters. Each sample is a 28x28 black and white image.

2. We represent each sample as a 1-dimensional vector with 784 features, each feature corresponding to a particular pixel. Each feature is an integer value between 0 and 255.

3. Since we are testing binary classifiers, we only consider two digits out of the ten in the dataset. We chose ”8” and ”0” because they have many overlapping features, making them more difficult to classify. After a 60/40 train/test split, the resulting dataset has 8000 training samples and 5500 test samples with an approximately equal amount of samples representing each class.

4. LeCun et al. (1998)

3.5 Wilt Dataset

1. Was created from satellite images taken over the forest in Japan that contains Japanese Oak Wilt and Japanese Pine Wilt trees B.A. Johnson et al (2013). The images were used to determine whether or not the trees were diseased, which means the color of their canopy was more on the red side than the green side B.A. Johnson et al (2013).

2. Features (6 total, 5 features, 1 class per sample):

   - class: ”w” (diseased trees), ”n” (all other land cover)
   - GLCM_Pan: GLCM mean texture (Pan band)
   - Mean_G: Mean green value
   - Mean_R: Mean red value
   - Mean_NIR: Mean NIR value
   - SD_Pan: Standard deviation (Pan band)

3. Number of classes: 2

4. Number of samples:

   - Training set: 4339
   - Testing set: 500

5. B.A. Johnson et al. (2013)

3.6 Summary of datasets

The following table shows the number of training samples in each class (A or B), for each dataset:

Dataset	Features	Train [classA, classB]	Test [classA, classB]
Wilt	5	[74, 4265]	[187, 313]
MNIST	784	[4132, 4104]	[2771, 2721]
Leukemia	7129	[27, 11]	[20, 14]
Letters	16	[455, 470]	[313, 305]
Breast Cancer	30	[132, 209]	[80, 148]

Table 2: Number of features and number of samples in classes A and B by dataset

4 Experimental analysis

All models were evaluated on each of the five datasets discussed.

To choose the γ function of the RBF kernel (where $\gamma = 1/(2\sigma^2)$) we follow the heuristic choice mentioned in (Gretton et al. 2012, 748) of setting σ to equal the median distance between points of the training data.

We performed grid-search to determine the optimal C margin value for the SVM models (Hsu et al. 2016, 5). Specifically, we performed 5-fold cross-validation over the 1D space $\gamma \in 0.01, 0, 100, 1000$ to select the best value for each individual model.

A value of $\lambda = 1$ was used for regularization.

We built our models using the scikit-learn package with Python. The linear regression models have built in regularization, so we set $\lambda = 0.0001$ so we could observe the effects of the RBF kernel without the influence of regularization.

SVM Kernel	Breast Cancer	MNIST	Leukemia	Letters	Wilt
RBF	100	100	100	100	100
Linear	0.01	0.01	0.01	0.01	100

Table 3: The C values chosen from the cross-validation grid-search for SVM models.

Model	Breast Cancer	MNIST	Leukemia	Letters	Wilt
Logistic Regression ℓ_1	0.0351	0.0133	0.0000	0.0146	0.3080
Logistic Regression ℓ_2	0.0351	0.0127	0.0178	0.0146	0.3080
Logistic Regression RBF	0.0351	0.0057	0.0294	0.0113	0.1340
SVM RBF	0.0236	0.0038	0.0294	0.0049	0.1440
SVM Linear	0.0439	0.0144	0.0294	0.0243	0.2960

Table 4: Empirical error when testing the five models on the five datasets.

5 Findings

Performance was poor on the Wilt dataset because the training samples were almost all in class B, so none of the classifiers could be properly fit. Even so, the RBF kernel was able to transform the data so it could be more effectively linearly separated than it could using the other methods.

Our highest ratio of features to observations, the Leukimia data set, has 7129 features. Since these are genes we would expect the true model to be sparse in features (i.e. only a few genes causing the disease). Our hypothesis that ℓ_1 performs better in such cases certainly holds here. Transforming the feature space into an even higher dimension had little effect compared to that of inducing sparsity.

The next highest ratio of features to observations is the MNIST data set. We expected that MNIST would not be linearly separable, and it turns out that transforming the feature space into a very high dimensional one via the RBF kernel gave good performance on both models which used that kernel. The encouraging of sparsity did not seem to make a major difference here.

When applied to the Letters data set, the various models performed similarly as they did on MNIST, indicating that Letters is not linearly separable, nor is the true model sparse enough in features to exhibit good performance under ℓ_1 regularization.

The Breast Cancer dataset has a fairly balanced ratio of features to observations compared to the other datasets. SVM RBF performs the best on this dataset. The SVM incorporates few data points in its fitting, has good generalization from the large-margin principle, and with the RBF kernel is able to fit data that is not linearly-separable in the original feature space, so the combination of these attributes likely leads to its superior performance here and in fact on most of the other data sets as well.

6 Summary and future work

There are many different ways to combat non-linearity in data. There are also many ways to combat overfitting. However, its a delicate process combining the techniques. We explored non-linear transformations and regularization, but there is still much to understand about how these techniques behave with feature reduction strategies like principal component analysis. Feature reduction offers a whole new approach to reduce overfitting and can produce interesting results when combined with subsequent kernel transformation.

While there seems to be some reliable go-to methods, it has been shown that there is no clear one-size-fits-all solution to accurately modeling and generalizing complex distributions. The safest method is to build an intuition of how these techniques behave on different sizes and shapes of data.

7 Group member contributions

Eric wrote the original code to estimate the various models, wrote code to load one of the datasets, wrote the Introduction and Approach sections of the paper, and handled communication with Professor Ross. James gathered the information for and wrote the Datsets section of the report, explained datset features to the team, wrote code to load two datasets, and contributed to the literature review. Jonny wrote code to process and load datasets, reformatted and modularized Eric's original code so it worked for all the datasets, handled experimenting and training the models, and wrote the Experimental Analysis and Summary and Future Work section of the report.

8 Works cited

8.1 Works cited in main report

Abu-Mostafa, Malik Magdon-Ismail and Hsuan-Tien Lin. 2012. *Learning with Data*, e-Chapter 8 ("Support Vector Machines") AMLBook.

Gretton, Arthur et al. 2012. "A Kernel Two-Sample Test." Journal of Machine Learning Research. Vol 13, p. 723-773.

Guestrin, Carlos. 2007. "Support Vector Machines." Lecture slides for "Machine Learning – 10701/15781" at Carnegie Mellon University.

Hsu, Chih-Wei et al. 2016. "A Practical Guide to Support Vector Classification." Department of Computer Science, National Taiwan University.

Lin, Yi. 2002. "Support Vector Machines and the Bayes Rule in Classification." *Data Mining and Knowledge Discovery* (6): 259–275.

Murphy, Kevin. 2012. *Machine Learning: A Probabilistic Perspective.* MIT Press: Cambridge, MA.

Theodoridis, Sergios and Konstantinos Koutroumbas. 2009. *Pattern Recognition.* Academic Press: Burlington, MA.

Theodoridis, Sergios. 2015. *Machine Learning: A Bayesian and Optimization Perspective.* Academic Press: London, United Kingdom.

Zhu, Ji and Trevor Hastie. 2004. "Kernel Logistic Regression and the Import Vector Machine." *Journal of Computational and Graphical Statistics.* Volume 14, 2005 - Issue 1.

8.2 Works cited in literature review

Abe S. 2010 "Two-Class Support Vector Machines". In *Support Vector Machines for Pattern Classification. Advances in Pattern Recognition.* Springer, London.

Aizerman, M. A., Emmanuel M. Braverman and Rozoner, L. I. 1964. "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning". *Automation and Remote Control.* 25: 821–837.

Boser, Bernhard E., Isabelle M. Guyon and Vladimir N. Vapnik. 1992. "A Training Algorithm for Optimal Margin Classifiers". *Proceedings of the fifth annual workshop on Computational learning theory – COLT '92.* p. 144.

Cox, DR. 1958. "The regression analysis of binary sequences (with discussion)". *J Roy Stat Soc B.* 20: 215–242.

Keerthi, S. Sathiya and Lin, Chih-Jen. 2003. "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel". Journal on Neural Computation. http://dx.doi.org/10.1162/089976603321891855

S.I. Lee, H. Lee, P. Abbeel, and A. Y. Ng. 2006. ”Efficient L1 Regularized Logistic Regression”. AAAI. https://www.aaai.org/Papers/AAAI/2006/AAAI06-064.pdf

Liu, Q., Chen, C., Zhang, Y. et al. 2011. “Feature Selection for Support Vector Machines with RBF Kernel.” *Artif Intell Rev* 36: 99. https://doi.org/10.1007/s10462-011-9205-2

Tychonoff, A.N. and V.Y. Arsenin. *Solution of ill-posed problems.* Winston & Sons: Washington, 1977.

8.3 Datasets

Brian Alan Johnson, Ryutaro Tateishi and Nguyen Thanh Hoan. 2013. ”A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees” International Journal of Remote Sensing, 34:20, 6969-6982.
https://doi.org/10.1080/01431161.2013.810825

P. W. Frey and D. J. Slate. 1991. ”Letter Recognition Using Holland-style Adaptive Classifiers”. Machine Learning Vol 6 #2. https://link.springer.com/article/10.1007/BF00114162

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, et al. 1999. ”Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring.” Science 286 (5439): 531–527. doi:10.1126/science.286.5439.531.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998 ”Gradient-based learning applied to document recognition.” Proceedings of the IEEE, 86(11):2278-2324. W.N. Street, W.H. Wolberg and O.L. Mangasarian. 1993. ”Nuclear feature extraction for breast tumor diagnosis.” IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870.