Training SVMs- and kNN Classifiers on a reduced MNIST data set

Philip Neuhart

Email: philipneuhart@live.at

Betreuer: Univ.-Prof. Dr. Arnold Neumaier

Seminar Optimisation

Contents

Classification

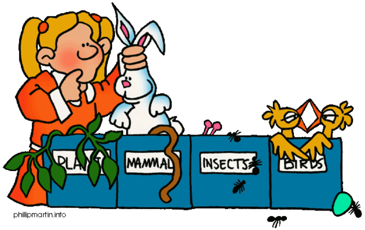


Figure: Data Scientist at work

Definition

Classification is the task of assigning categories/classes to observations.

- Observations also called instances, inputs
- Classes also referred to as labels or outputs

Statistical Classification

Definition

- Observations are elements of the input space $X \subseteq \mathbb{R}^n$
- Labels are elements of the output domain $Y = \{1, ...m\}$
- Training set $S = ((x_1, y_1), ..., (x_l, y_l)) \subseteq (X \times Y)^l$
- $(x, y) \in S \implies (x, y)$ training example
- $n = \#$ of attributes, $m = \#$ of classes, $l = \#$ of training examples

Definition

Statistical Classification models try to predict an output y given an input x. They use the information contained in the training data to form a decision rule for classifying new unlabeled input data, also referred to as test data.

Statistical Classification

Model Examples

- Support Vector Machines (SVMs)
- k-Nearest Neighbour (kNN)
- Random Forest
- etc.

Applications

- Handwritten Digit/Character Recognition
- Image Classification
- Market Forecasting
- etc.

Linear Classification

Binary Classification

Definition

$f : X \subseteq \mathbb{R}^n \to \mathbb{R}, x = (x^1, ..., x^n)^T \in X$

$$f(x) = \langle w, x \rangle + b = \sum_{i=1}^{n} w^i x^i + b$$

Decision function

$$h(x) = \begin{cases} \text{Class A} & \text{sign}(f(x)) = 1 \\ \text{Class B} & \text{sign}(f(x)) = -1 \end{cases} \quad (\text{sign}(0) = 1)$$

$f(x) \ldots$ hypothesis
$w \ldots$ weight vector
$b \ldots$ bias

Geometric Interpretation

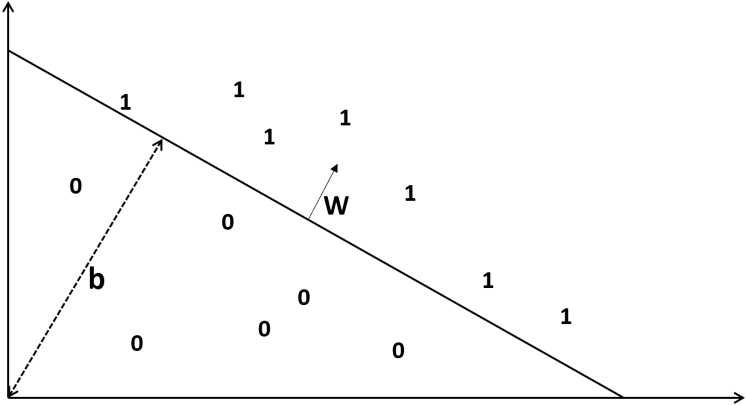


Figure: A separating hyperplane (w, b) for $dim(X) = 2$

Dual representation

Hypothesis f can also be expressed in dual representation:

$$
\begin{aligned}
f(x) &= \langle w, x \rangle + b \\
&= \left\langle \sum_{j=1}^{l} \alpha_i y_i x_i, x \right\rangle + b \\
&= \sum_{j=1}^{l} \alpha_i y_i \langle x_i, x \rangle + b,
\end{aligned}
$$

$w = \sum_{j=1}^{l} \alpha_i y_i x_i$.

Linear Classification

Definition

A training set is said to be linearly separable if the data can be separated into its classes in the input space by a hyperplane.

Properties of Linear Classifiers

- Computationally efficient
- BUT: Training data must be linearly separable

Learning in Feature Space

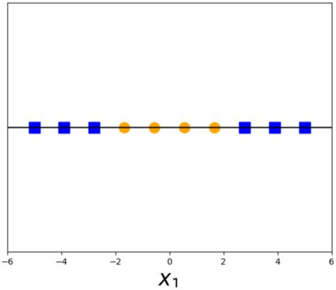
Feature Space

Linearly inseparable data $\Longrightarrow$ no Linear Classifiers? Not quite!
Solution: Feature Space
Can map input space X into a new space $F = \{\phi(x) : x \in X\}$:

$$x = (x_1, ..., x_n) \mapsto \phi(x) = (\phi_1(x), ..., \phi_N(x))$$

Components of x . . . attributes
Components of $\phi(x)$ . . . features
ϕ . . . feature map

Learning in Feature Space



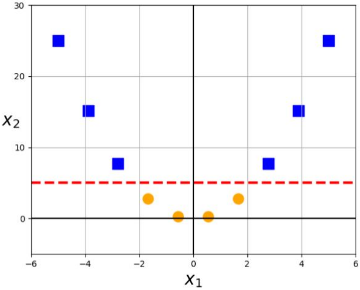


Figure: feature map $\phi(x) = x^2$

Learning in Feature Space

Hypothesis

Extending hypothesis space:

$$f(\mathrm{x}) = \sum_{i=1}^{N} w_i \phi_i(\mathrm{x}) + b$$

Dual representation

$$f(\mathrm{x}) = \sum_{i=1}^{\ell} \alpha_i y_i \langle \phi(\mathrm{x}_i) \cdot \phi(\mathrm{x}) \rangle + b$$

$\phi(x)$ can be non-linear!

Learning in Feature Space

Implicit Mapping into Feature Space

- Want to make the data linearly separable through feature mapping
- Don't want to pay the computational costs
- Solution: Kernel functions!

Example

$$K(x,z) = (\langle x \cdot z \rangle + c)^2 = \left(\sum_{i=1}^{n} x_i z_i + c\right)\left(\sum_{j=1}^{n} x_j z_j + c\right)$$

$$= \sum_{(i,j)=(1,1)}^{(n,n)} (x_i x_j)(z_i z_j) + \sum_{i=1}^{n}\left(\sqrt{2c}x_i\right)\left(\sqrt{2c}z_i\right) + c^2$$

Learning in Feature Space

Example

$$\phi(x) = \underbrace{(x_1x_1, x_1x_2, \ldots, x_nx_n, \sqrt{2}cx_1, \ldots, \sqrt{2}cx_n, c)}_{\binom{n+2}{2} \text{ features}}$$

Evaluating Kernel: $O(n)$ operations
Evaluating ϕ, computing inner product: $O(n^2)$ operations
$\implies$ Computationally efficient!

Kernels

Definition

A kernel is a function $K : X \times X \to \mathbb{R}$, such that for all $\mathrm{x}, \mathrm{z} \in X$

$$K(\mathrm{x}, \mathrm{z}) = \langle \phi(\mathrm{x}) \cdot \phi(\mathrm{z}) \rangle$$

where ϕ is a mapping from X to a feature space F.

Characterisation of Kernels

(Mercer) Let X be a finite input space with $|X| = n$. Then $K(\mathrm{x}, \mathrm{z})$ is a kernel function if and only if the matrix

$$\mathrm{K} = \left(K\left(\mathrm{x}_i, \mathrm{x}_j\right)\right)_{i,j=1}^{n}.$$

is symmetric and positive semi-definite (has only non-negative eigenvalues).

Examples

Linear kernel	$K(x, z) = \langle x, z \rangle + c$
Radial basis function kernel	$K(x, z) = exp(-\gamma \|x - z\|^2)$
Polynomial kernel	$K(x, z) = (x^T z + c)^d$

$\gamma > 0$, $c \in \mathbb{R}$ and $d \in \mathbb{Z}^+$ parameters

Maximal Margin Classifier

Definition

Separating hyperplane (w, b) for a training set S, $|S| = l$
Functional margin of an example (x_i, y_i):

$$\gamma_i^f = y_i \cdot (\langle w, x_i \rangle + b)$$

Geometric margin of (x_i, y_i):

$$\gamma_i = \gamma_i^g := \frac{1}{\|w\|} \gamma_i^f$$

Functional & Geometric margin of the hyperplane:

$$\gamma^f = \min_{1 \leq i \leq l} \gamma_i^f \qquad \gamma = \min_{1 \leq i \leq l} \gamma_i$$

Note: Rescaling the hyperplane $(\lambda w, \lambda b), \lambda \in \mathbb{R}$, doesn't change it!

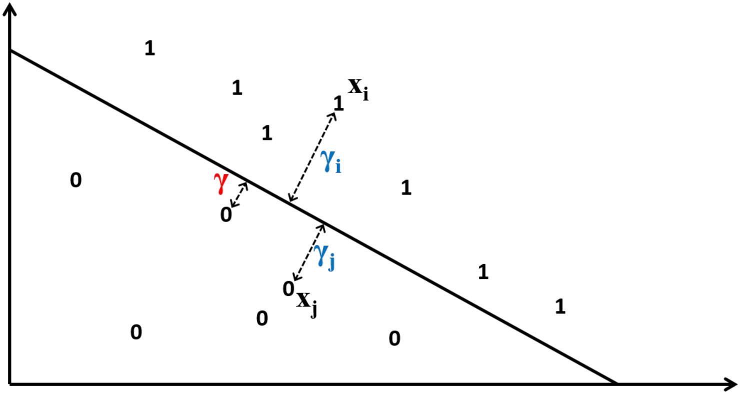


Figure: Geometric Margins

Definition

The maximal margin hyperplane is the hyperplane realising the maximum geometric margin over all hyperplanes.

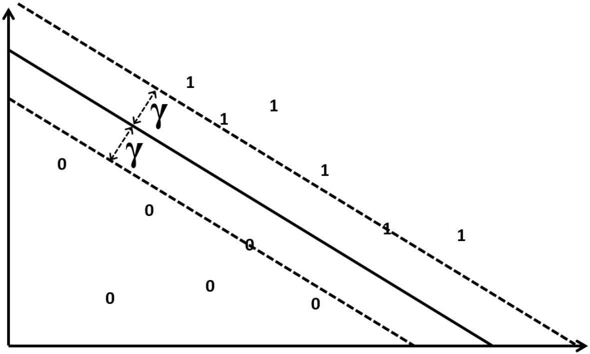


Figure: Maximal Margin Hyperplane

Maximal Margin Classifier

Maximal Margin Hyperplane

Given a linearly separable training sample $S = ((\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_\ell, y_\ell))$, the hyperplane (w, b) that solves the optimisation problem

$$\begin{array}{ll} \text{minimise}_{\,\mathrm{w},b} & \langle \mathrm{w} \cdot \mathrm{w} \rangle \\ \text{subject to} & y_i \left(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b \right) \geq 1 \\ & i = 1, \ldots, \ell \end{array}$$

realises the maximal margin hyperplane with geometric margin $\gamma = 1/\|\mathrm{w}\|_2$.

Proof

Let (w^*, b^*) be the solution of the above optimisation problem. At least one constraint must be active for (w^*, b^*), since otherwise we can find $\lambda > 1$ s.t. $(\frac{w^*}{\lambda}, \frac{b^*}{\lambda})$ is also a feasible solution and

Maximal Margin Classifier

$$\left\langle \frac{w^*}{\lambda} \cdot \frac{w^*}{\lambda} \right\rangle < \langle w^* \cdot w^* \rangle \;\lightning$$

It also follows that (w^*, b^*) has a geometric margin $\gamma = \frac{1}{\|w^*\|}$.
Now, let $(\tilde{w}, \tilde{b})$ be the maximal margin hyperplane with functional margin $\tilde{\gamma}^f = 1$ and geometric margin $\frac{1}{\|\tilde{w}\|}$. Then the following holds

$$\frac{1}{\|w^*\|} = \gamma \leq \tilde{\gamma} = \frac{1}{\|\tilde{w}\|}.$$

But since $(\tilde{w}, \tilde{b})$ is also a feasible solution, we find that

$$\|w^*\|^2 = \langle w^* \cdot w^* \rangle \leq \langle \tilde{w} \cdot \tilde{w} \rangle = \|\tilde{w}\|^2.$$

It follows that

$$\frac{1}{\|w^*\|} = \gamma \geq \tilde{\gamma} = \frac{1}{\|\tilde{w}\|}.$$

The primal Lagrangian,

$$W(\alpha) := L(\mathrm{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathrm{w} \cdot \mathrm{w} \rangle - \sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b \right) - 1 \right],$$

is stationary at the optimum. Differentiate with respect to w and b,

$$\frac{\partial L(\mathrm{w}, b, \alpha)}{\partial \mathrm{w}} = \mathrm{w} - \sum_{i=1}^{\ell} y_i \alpha_i \mathrm{x}_i = 0$$

$$\frac{\partial L(\mathrm{w}, b, \alpha)}{\partial b} = \sum_{i=1}^{\ell} y_i \alpha_i = 0$$

to obtain

$$\mathrm{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathrm{x}_i,$$

$$0 = \sum_{i=1}^{\ell} \alpha_i y_i$$

Resubstituting into the primal Lagrangian:

$$W(\alpha) = L(\mathrm{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathrm{w} \cdot \mathrm{w} \rangle - \sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\langle \mathrm{w} \cdot \mathrm{x}_i \rangle + b \right) - 1 \right]$$

$$= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \left\langle \mathrm{x}_i \cdot \mathrm{x}_j \right\rangle$$

Note: Training data only appears in the inner product!

Maximal Margin Classifier

The following proposition follows by the strong duality theorem.

Dual representation

Consider a linearly separable training sample

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell))$$

and suppose the parameters α^* solve the following quadratic optimisation problem:

$$\begin{array}{ll} \text{maximise} & W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \\ \text{subject to} & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & \alpha_i \geq 0, i = 1, \ldots, \ell \end{array}$$

Then the weight vector $\mathbf{w}^* = \sum_{i=1}^{\ell} y_i \alpha_i^* \mathbf{x}_i$ realises the maximal margin hyperplane with geometric margin

$$\gamma = 1 / \|\mathbf{w}^*\|_2$$

$b*$ must be found with the help of the primal constraints:

$$b^* = -\frac{\max_{y_i=-1}\left(\langle \mathrm{w}^* \cdot \mathrm{x}_i \rangle\right) + \min_{y=-1}\left((\mathrm{w}^* \cdot \mathrm{x}_i \rangle\right)}{2}$$

KKT complementarity conditions: α^*, (w^*, b^*) must satisfy

$$\alpha_i^* \left[y_i \left(\langle \mathrm{w}^* \cdot \mathrm{x}_i \rangle + b^* \right) - 1 \right] = 0, \quad i = 1, \ldots, \ell.$$

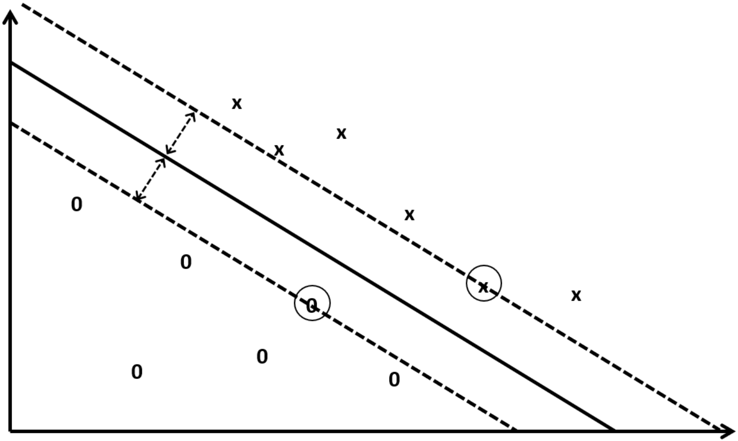


Figure: Maximal Margin Hyperplane and Support Vectors

Maximal Margin Hyperplane

Support Vectors

$$f(x, \alpha^*, b^*) = \sum_{i=1}^{l} y_i \alpha_i^* \langle x_i \cdot x \rangle + b^*$$
$$= \sum_{i \in sv} y_i \alpha_i^* \langle x_i \cdot x \rangle + b^*$$

Now the sum only runs over the support vectors which are generally far fewer than there are training examples.

Main result

Consider a training sample

$$S = ((\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_\ell, y_\ell))$$

that is linearly separable in the feature space implicitly defined by the kernel $K(\mathrm{x}, \mathrm{z})$

Maximal Margin Hyperplane

Main result

Suppose the parameters α^* and b^* solve the following quadratic optimisation problem:

$$\text{maximise } W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{ij=1}^{l} y_i y_j \alpha_i \alpha_j K\left(\mathrm{x}_i, \mathrm{x}_j\right),$$

$$\text{subject to } \sum_{i=1}^{\ell} y_i \alpha_i = 0$$

$$\alpha_i \geq 0, i = 1, \ldots, \ell$$

Then the decision rule given by $\operatorname{sgn}(f(\mathrm{x}))$, where $f(\mathrm{x}) = \sum_{i \in sv} y_i \alpha_i^* K\left(\mathrm{x}_i, \mathrm{x}\right) + b^*$, is equivalent to the maximal margin hyperplane in the feature space implicitly defined by the kernel $K(\mathrm{x}, \mathrm{z})$.

Remarks

Separability

Although there are ways to force separability for any training set in an accordingly chosen feature space it is most often not desirable to do so, since this approach leads to overfitting. Instead, one could allow some misclassifications in the learning process, i.e., not force the training data to be linearly separable in the feature space. Although out of the scope of this assignment, the soft margin classifier poses a well-known example for such an approach.

Remarks

Multi-class case

A common way of generalizing the binary classification theory to the multi-class case is to use the One-vs-all approach. Given an m-class training set S. For each class $j \in \{1, ..., m\}$, the classifier is trained on a binary training set, formed by aggregating the examples from all classes, except class j, to one single class. This way, one obtains m hypothesis. In the case of linear classifiers, and thus also for SVMs, the decision rule is then given by assigning the class, for which the corresponding hypothesis outputs the maximum value, when evaluated on an instance.

k-Nearest Neighbour

kNN

- Training set $S = \{(x_i, y_i)\}_{i=1}^{l} \subseteq (X \times Y)^l$
- $N_x^k \ldots$ set of k nearest training examples of x
- $P(Y = j | X = x) = \frac{1}{k} \sum_{x_i \in N_x^k} \delta_{y_i j}$
- Assign the class with highest probability to x

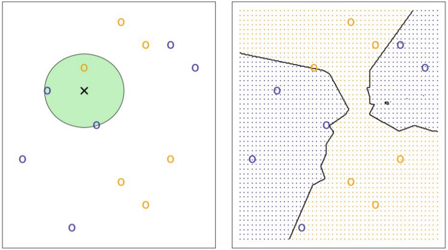


Figure: kNN classification, k=3, Euclidean distance

MNIST data set

Classifier	TER(%)
K-nearest-neighbors, Eucl. (L2)	3.09
K-NN, shape context matching*	0.63
SVM, Gaussian Kernel	1.4
Virtual SVM, deg-9 poly, [...]*	0.56
committee of 35 conv. net [...]*	0.23

Table: Benchmarks for test error rates (TER) different classifiers

* with preprocessing of data

Application

Procedure

- MNIST/1, MNIST/5, MNIST/10
- Extend training set
- Find super-trainers
- Test accuracy
- Analyse results

Extending the training set

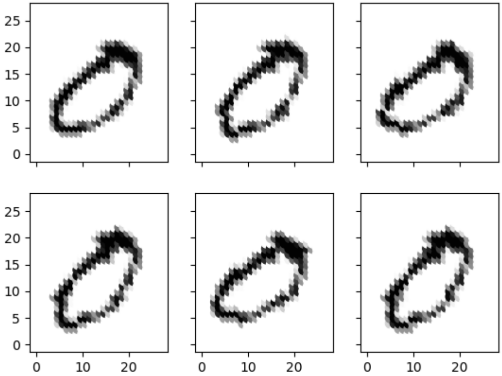


Figure: Rotations of a training example

Extending the training set

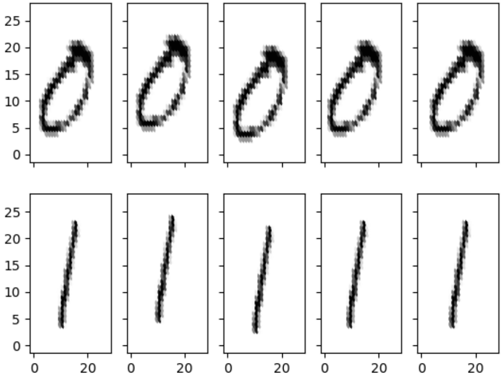


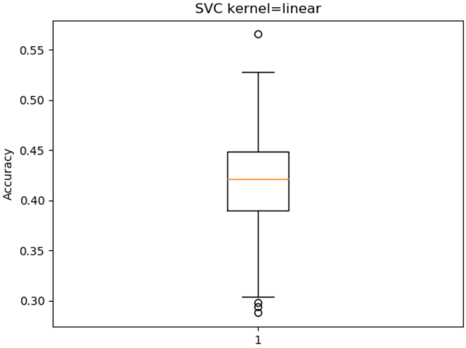
Figure: Shift of a training example

Super-trainers

Definition

The term super-trainers is used to describe either the best available MNIST/1, MNIST/5 or MNIST/10 data set.

- Pool of possible super-trainers: 1000 examples for each digit
- Sample 1000 MNIST/1 data sets
- Test accuracy



Super-trainers

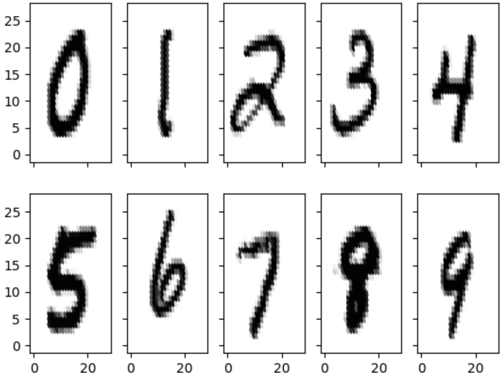


Figure: Super-trainers

Super-trainers

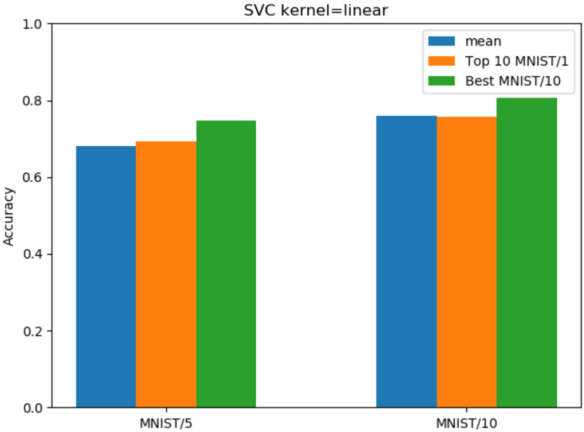


Figure: Comparison Top10 MNIST/1 vs. Best MNIST/10

Final Results

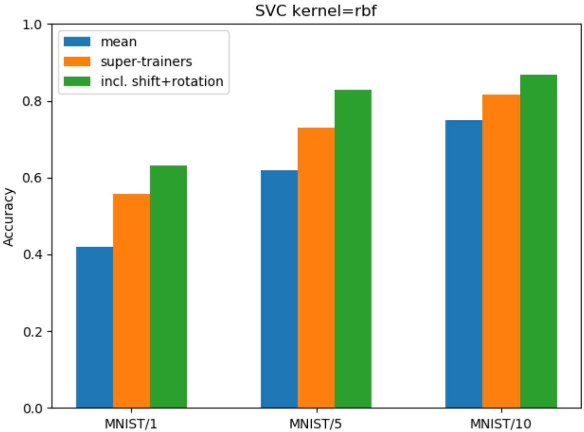


Figure: Performance Increases - SVC - rbf kernel

Final Results

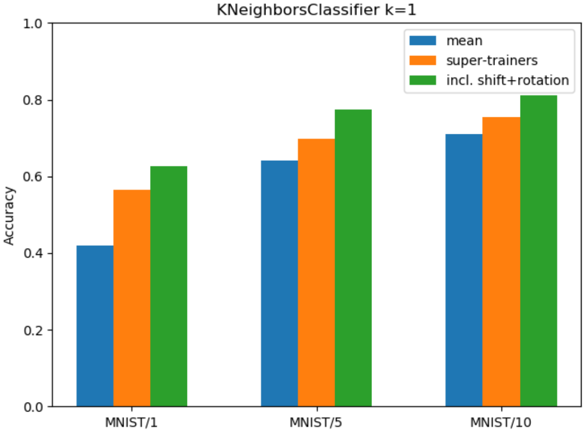


Figure: Performance Increases - kNN - k=1

The training sets are relatively small, thus small values of *k* yielded better results.

References

- James; Witten et al.An Introduction to Statistical Learning.Springer Science+Business Media New York, 2017.isbn: 978-1-4614-7137-0.
- Beneschi.MNIST - EDA, Preprocessing & Classifiers.url:https://www. kaggle.com / damienbeneschi / mnist - eda - preprocessing -classifiers.
- Boser, Guyon, and Vapnik. A Training Algorithm for Optimal Margin Classifiers. COLT '92. Pittsburgh, Penn-sylvania, USA: Association for Computing Machinery, 1992, pp. 144–152.isbn: 089791497X.url:https://doi.org/10.1145/130385.130401.
- Cristianini and Shawe-Taylor. An Introduction to Support VectorMachines and Other Kernel-based Learning Methods. Cambridge UniversityPress, 2000.isbn: 9781139643634.url:https://books.google.at/books?id=I%5C_0gAwAAQBAJ.23

References

- scikit-learn developers.Nearest Neighbors. Accessed: 2021-05-31.url:https://scikit-learn.org/stable/modules/neighbors.html #nearest-neighbors-classification.
- Dietterich. "Overfitting and Undercomputing in Machine Learning".In:ACM Comput. Surv.27.3 (Sept. 1995), pp. 326–327.issn: 0360-0300.url:https://doi.org/10.1145/212094.212114.
- Digit Recogniser. Accessed: 2020-05-31.url:https://www.kaggle.com/c/digit-recognizer/overview.
- Dye. "A primer on kernel methods". In: (2019).url:https://towardsdatascience.com/an-intro-to-kernels-9ff6c6a6a8dc.

References

- Kotsiantis, Dimitris Kanellopoulos, and Panagiotis E Pintelas. "Data preprocessing for supervised leaning". In:International Journal ofComputer Science1.2 (2006), pp. 111–117.url:`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.8413&rep=rep1&type=pdf`.
- Fix; Hodges "Discriminatory Analysis. NonparametricDiscrimination: Consistency Properties". In: (1951).
- LeCun, Cortes, and Burges. "MNIST handwritten digitdatabase". In:ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist/(2010).
- Marques de Sá. "Statistical Classification". In:Applied StatisticsUsing SPSS, STATISTICA and MATLAB. Berlin, Heidelberg: SpringerBerlin Heidelberg, 2003.isbn: 978-3-662-05804-6.url:https://doi.org/10.1007/978-3-662-05804-6_6.

References

- Neuhart.Linear Learning Machines. University of Vienna, 2020.
- Nielsen.Reduced MNIST: how well can machines learn from smalldata?Accessed: 2021-04-03.url:http://cognitivemedium.com/rmnist.
- Pietersma. “Feature space learning in Support Vector Machinesthrough Dual Objective optimization”. In: (2010).url:http://www.ai.rug.nl/ mwiering/AD_Pietersma_Thesis.pdf.
- Pillow library. Accessed: 2020-05-31.url:https://pillow.readthedocs.io/en/stable/.

References

- Wilimitis. "The Kernel Trick in Support Vector Classification". In:(2018).url:https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f
- Recent Advances of Large-Scale Linear Classification. Accessed: 2020-05-31.url:https://dmkd.cs.vt.edu/TUTORIAL/Bigdata/Papers/IEEE12.pdf.
- sklearn library.url:https://scikit-learn.org/stable/.24