Math 156 - Machine Learning

Seed Classification Project

Osman Akar & Cory Ye

UCLA Applied Mathematics

Date: 12/11/2017

Problem Statement

Consider the **multi-class Kernel classifier**, the generalization of the 2-class Kernel classifier to datasets $\mathcal{D}$ with multiple classes $\mathcal{C}$. In this project, the data $\mathcal{D}$ that we analyze is the *seeds* dataset found in the UCI Machine Learning Repo [1], a set of attributes and classifications derived from the kernels of three types of wheat: Kama, Rosa, and Canadian. $\mathcal{D}$ was robustly collected via the analysis and gradient clustering of wheat X-ray images in [2], defined by the collection:

$$\mathcal{D} = \left\{ \mathbf{x} \in \mathcal{D} \hookrightarrow \begin{bmatrix} \mathbf{A} & \mathbf{P} & \mathbf{c} & \mathbf{l} & \mathbf{w} & \mathbf{a}_s & \mathbf{l}_g \end{bmatrix}^T \in (\mathbb{R}^{1\times|\mathcal{D}|})^7 \sim \mathbb{R}^{7\times|\mathcal{D}|} \;\middle|\; \mathcal{C} \in \{0,1\}^{|\mathcal{D}|\times 3} \right\}$$

$$\mathcal{C} = \begin{bmatrix} \mathcal{C}_{(i,j)} = 1 \overset{\chi}{\iff} \mathbf{x}_i \in \mathcal{C}_j \subset \mathcal{D} \end{bmatrix} \in \{0,1\}^{|\mathcal{D}|\times 3}$$

The collection $\mathcal{D}$ contains the classification matrix $\mathcal{C}$ of the three classes corresponding to wheat species, A the cross-sectional area of the seed kernel, P the perimeter of the seed kernel, c the compactness of the seed kernel computed by the nonlinear function $c(A,P) = 4\pi \cdot A/P^2$, l the length of the kernel, w the width of the kernel, a_s the asymmetry coefficient, and l_g the length of the kernel groove with $|\mathcal{D}| = 210$ and classes $|\mathcal{C}_i| = 70$ for $i \in \{1,2,3\} \subset \mathbb{Z}$. $\mathcal{D}$ is a separable dataset (as seeds of equivalent wheat species share approximately similar attributes), so that the technique of Kernel classification/clustering is hypothetically effective. Hence, we aim to train a multi-class Kernel classifier to accurately classify/partition the *seeds* dataset $\mathcal{D} = \bigvee_i \mathcal{C}_i$ as a function of attributes $\mathbf{x} = (A, P, c, l, w, a_s, l_g) \in \mathbb{R}^7$.

Machine Learning Model

To derive the multi-class Kernel classifier, we apply the duality of representations between linear classification model and multi-class Kernel classifier.

$$\mathbf{y}(\mathbf{x}) = \mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}) \iff \mathbf{y}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T}$$

Training the weight matrix $\mathbf{W} \in \mathbb{R}^{7\times 3}$ and taking $\mathbf{x} \in \mathcal{C}_i \iff i = \underset{k}{\mathbf{argmax}}\ y_k(\mathbf{x})$ classifies $\mathbf{x} \in \mathcal{D}$. Define the classification vector/target $\mathbf{t}_n = \mathcal{C}^T_{(:,n)} \in \mathbb{R}^3$ of $\mathbf{x}_n \in \mathcal{D}$ and the λ-regularized least-squares error/optimization measure on the training data $\mathcal{D}$.

$$J(\mathbf{W}) = \frac{1}{2}\sum_{n=1}^{|\mathcal{D}|} \|\mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n) - \mathbf{t}_n\|^2 + \frac{\lambda}{2}\|\mathbf{W}\|_F^2$$

$$\boldsymbol{\alpha}_n = -\frac{1}{\lambda}\left[\mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n) - \mathbf{t}_n\right] \in \mathbb{R}^3 \implies \left[\mathbf{W}_{optimal} = -\frac{1}{\lambda}\sum_n \boldsymbol{\phi}(\mathbf{x}_n)(\mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n) - \mathbf{t}_n)^T = \sum_n \boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\alpha}_n^T = \boldsymbol{\Phi}\mathbf{A}^T\right]$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}_1) & \cdots & \boldsymbol{\phi}(\mathbf{x}_n) \end{bmatrix} \in \mathbb{R}^{7\times|\mathcal{D}|} \qquad \mathbf{A} = \begin{bmatrix} \boldsymbol{\alpha}_1 & \cdots & \boldsymbol{\alpha}_n \end{bmatrix} \in \mathbb{R}^{3\times|\mathcal{D}|}$$

Reformulating the least-squares algorithm by substitution of $\mathbf{W}_{optimal} = \mathbf{\Phi}\mathbf{A}^T$ into $J(\mathbf{W})$ implies that:

$$\mathbf{T} = \mathcal{C} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_n^T \end{bmatrix} \in \{0,1\}^{|\mathcal{D}|\times 3} \implies J(\mathbf{A}) = \frac{1}{2}\mathbf{A}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{A}^T - \mathbf{A}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{T} + \frac{1}{2}\mathbf{T}^T\mathbf{T} + \frac{\lambda}{2}\mathbf{A}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{A}^T$$

$$\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi} = \left[\mathbf{K}_{nm} = \boldsymbol{\phi}(\mathbf{x}_n)^T\boldsymbol{\phi}(\mathbf{x}_m)\right] \in \mathbb{R}^{|\mathcal{D}|^2} \implies J(\mathbf{A}) = \frac{1}{2}\mathbf{A}\mathbf{K}\mathbf{K}\mathbf{A}^T - \mathbf{A}\mathbf{K}\mathbf{T} + \frac{1}{2}\mathbf{T}^T\mathbf{T} + \frac{\lambda}{2}\mathbf{A}\mathbf{K}\mathbf{A}^T$$

Apply the previous equations $\lambda\boldsymbol{\alpha}_n = \mathbf{t}_n - \mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n)$ and $\mathbf{W} = \mathbf{\Phi}\mathbf{A}^T$ with Kernel matrix $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ and kernel vector $\mathbf{k}(\mathbf{x}) = \left[\left\{\boldsymbol{\phi}(\mathbf{x}_n)^T\boldsymbol{\phi}(\mathbf{x})\right\}_{n=1}^{|\mathcal{D}|}\right] = \mathbf{\Phi}^T\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{|\mathcal{D}|}$ implies that:

$$\begin{aligned} \lambda\boldsymbol{\alpha}_n + \mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n) = \mathbf{t}_n &\implies \lambda\boldsymbol{\alpha}_n + \mathbf{A}\mathbf{\Phi}^T\boldsymbol{\phi}(\mathbf{x}_n) = \mathbf{t}_n \\ \xrightarrow{\forall\,\mathbf{x}_n\in\mathcal{D}} \lambda\mathbf{A} + \mathbf{A}\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{T}^T &\implies \mathbf{A}_{optimal}^T = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T} \\ \mathbf{A}_{optimal}^T &\implies \left[\ \mathbf{y}(\mathbf{x}) = \mathbf{A}\mathbf{\Phi}^T\boldsymbol{\phi}(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T\mathbf{\Phi}\mathbf{A}^T = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T}\ \right] \quad \square \end{aligned}$$

Hence, the duality of optimization methods between multi-class linear basis classifiers and multi-class Kernel classifiers is proved. In the context of this project, we design the τ-normalized Kernel matrix/function with the Vandermonde polynomial term vector $\mathcal{V}(\mathbf{x})$ and component-wise Hadamard product $\circ : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ as the *Gaussian radial basis kernel function* $k(\mathbf{x}_n, \mathbf{x}_m)$:

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{x}) &= e^{-\frac{\|x\|^2}{\tau\sigma_x^2}} \cdot \left[\left(\sqrt{\frac{1}{n!\cdot\sigma_x^{2n}}} \;\middle|\; n = \mathbf{order}[\mathcal{V}(\mathbf{x})_i]\right)_i\right] \circ \mathcal{V}(\mathbf{x}) \\ \implies &\left[\ \mathbf{K}_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\phi}(\mathbf{x}_n)^T\boldsymbol{\phi}(\mathbf{x}_m) = e^{-\|\mathbf{x}_n - \mathbf{x}_m\|^2/\tau\sigma_n\sigma_m}\ \right] \end{aligned}$$

σ_n is chosen as the distance $\|\mathbf{x}_n - \mathbf{x}_n(k)\| \in \mathbb{R}$ of $\mathbf{x}_n \in \mathcal{D}$ to the k-th nearest neighbor $\mathbf{x}_n(k)$ of $\mathbf{x}_n$ for some optimal $k \in \{2, \ldots, 20\} \subset \mathbb{Z}$ and parameters $\lambda, \tau \in \mathbb{R}$.

The Kernel classifier model $\mathbf{y}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T}$ of complexity $\mathcal{O}(|\mathcal{D}|^{2.376})$ (because of the inversion $(\mathbf{K} + \lambda\mathbf{I})^{-1}$ through *Coppersmith–Winograd inversion algorithms*) is chosen over support vector machines (SVM), linear classifiers, and perceptrons/neural networks (NN) because it balances computational complexity and optimization precision. $\mathcal{D}$ is not linearly separable, so linear models rely on previous knowledge of the nonlinear, non-convex feature map $\boldsymbol{\phi}$ that is difficult to determine. SVMs require a combination of nonlinear-iterative, kernel, and maximum margin (one-versus-all) methods with complexity $\mathcal{O}\left[\mathbf{max}(|\mathcal{D}|, \mathbf{dim}(\mathbb{R}^7)) \cdot \mathbf{min}(|\mathcal{D}|, \mathbf{dim}(\mathbb{R}^7))^2\right] \sim \mathcal{O}(|\mathcal{D}|^{2.376})$ and multiple convex inequality constraints, which are difficult to solve for multi-class datasets $\mathcal{D}$. Likewise, neural networks utilize complex yet adaptive stochastic gradient descent training with complexity $\mathcal{O}(K \cdot |\mathcal{D}| \cdot |\mathbf{W}|) \ll \mathcal{O}(|\mathcal{D}|^{2.376})$ (in this particular case, because $|\mathbf{W}| \ll |\mathcal{D}|$) but provides minimal control over learning cycles $K \in \mathbb{Z}$ or optimality/accuracy $\alpha \in [0, 1]$ compared to the Kernel classifier that retrieves an optimal solution for the model. Hence, the Kernel classifier is the optimal choice for the separable dataset $\mathcal{D}_{seeds}$.

Classification Algorithm

To implement the multi-class Kernel classifier, we train/compute the Kernel matrix $\mathbf{K}$ with the kernel function $k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\phi}(\mathbf{x}_n)^T\boldsymbol{\phi}(\mathbf{x}_m) = e^{-\|\mathbf{x}_n - \mathbf{x}_m\|^2/\tau\sigma_n\sigma_m}$ and classification training matrix $\mathbf{T}$ that contains $|\mathcal{D}_{train}| = 90$ data-points consisting of $|\mathcal{C}_{i,train}| = 30$ points per class $\mathcal{C}_i$ for $i \in \{1, 2, 3\}$. Subsequently, we simulate the Kernel classifier $\mathbf{y}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T}$ on the test dataset $\mathcal{D}_{test} = \mathcal{D} - \mathcal{D}_{train}$ with $|\mathcal{D}_{test}| = 120$ data-points consisting of $|\mathcal{C}_{i,test}| = 40$ points per class $\mathcal{C}_i$ for $i \in \{1, 2, 3\}$. Optionally, we normalize the attributes/components $(A, P, c, l, w, a_s, l_g)$ of the dataset $\mathcal{D}$ to fixed variance $\sigma_{attr} =$

$\sigma_{\{A,P,c,l,w,a_s,l_g\}} = 0.0048$ such that the Gaussian radial basis kernel function weights the measure/distance of the attributes in classification without bias. Accuracy/performance α of Kernel classification is computed by:

$$\alpha(\mathcal{D}) = \frac{\left|\left\{\mathbf{x} \in \mathcal{D} \mid \mathbf{x} \in \mathcal{C}_i \ \wedge \ i = \underset{k}{\mathbf{argmax}}\ y_k(\mathbf{x})\right\}\right|}{|\mathcal{D}|} \in [0,1]$$

Multi-Class Kernel Classifier Algorithm

(a) Define the training classification matrix $\mathbf{T} = \mathcal{C}_{train} = \begin{bmatrix} | \\ \{\mathbf{t}_{n,train}^T\}_{n=1}^{|\mathcal{D}_{train}|} \\ | \end{bmatrix} \in \{0,1\}^{|\mathcal{D}_{train}| \times 3}$.

(b) *Optional*: Normalize the dataset $\mathcal{D}$ to σ_{attr}.

(c) Compute k-th nearest neighbors $\mathbf{x} \sim_{\mathbf{knn}} \{\mathbf{x}_k\}$ and distances $\sigma_k = \|\mathbf{x} - \mathbf{x}_k\|$ for all $\mathbf{x} \in \mathcal{D}$.

(d) Compute the Kernel matrix $\mathbf{K}$.

$$\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi} = \left[\mathbf{K}_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \boldsymbol{\phi}(\mathbf{x}_n)^T\boldsymbol{\phi}(\mathbf{x}_m) = e^{-\|\mathbf{x}_n - \mathbf{x}_m\|^2/\tau\sigma_n\sigma_m}\right] \in \mathbb{R}^{|\mathcal{D}_{train}|^2}$$

(e) Compute the matrix $(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T} \in \mathbb{R}^{|\mathcal{D}_{train}| \times 3}$.

(f) Compute the kernel vector $\mathbf{k}(\mathbf{x})$.

$$\forall\, \mathbf{x} \in \mathcal{D}_{test} \implies \mathbf{k}(\mathbf{x}) = \left[\left\{k(\mathbf{x}_{n,train}, \mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}_{n,train})^T\boldsymbol{\phi}(\mathbf{x})\right\}_{n=1}^{|\mathcal{D}_{train}|}\right] \in \mathbb{R}^{|\mathcal{D}_{train}|}$$

(g) Classify $\mathbf{x} \in \mathcal{D}_{test}$ and compute $\alpha(\mathcal{D}_{test})$ via Kernel classifier $\mathbf{y}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T}$ and classification algorithm $\mathbf{x} \in \mathcal{C}_i \iff i = \underset{k}{\mathbf{argmax}}\ y_k(\mathbf{x})$.

Result and Analysis

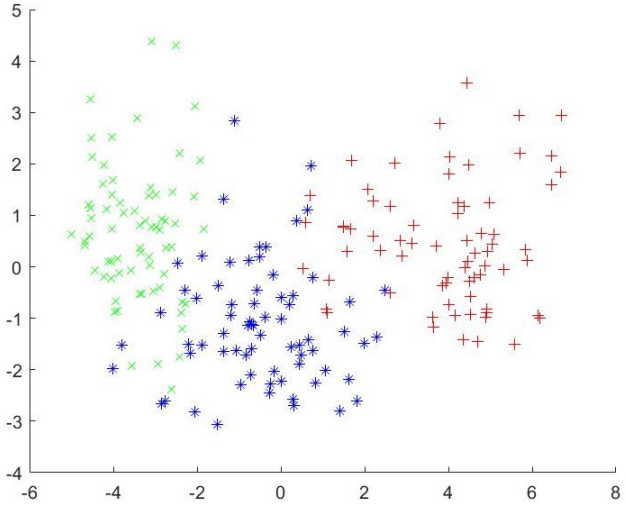

Fig. 1 – PCA Projection/True Classification of $\mathcal{D}$

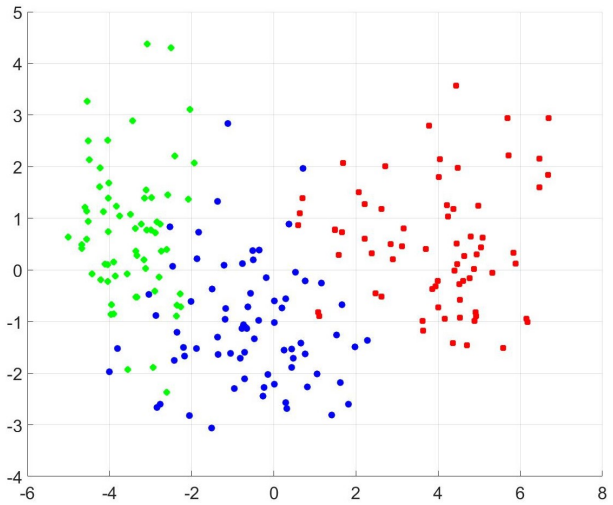

Fig. 2 – Kernel Classification of $\mathcal{D}$

Executing the Kernel classification algorithm on $\mathcal{D}_{seeds}$ induces an optimal classification accuracy of $\alpha = 91.67\%$ with parameters $\tau = 8$, $\lambda = 1$, and $k = 2$. The classification accuracies for various combinations

of parameters are reported:

Table 3

$\boldsymbol{\lambda}$	1	1	10	10	0.5	1	100	1	1
$\boldsymbol{\tau}$	1	8	10	10	8	0.2	8	20	20
$\mathbf{k}$	2	2	2	10	2	20	3	2	20
$\boldsymbol{\alpha}(\mathcal{D})$	90%	91.67%	90.83%	89.17%	90.83%	88.33%	90%	83.33%	87.5%

It follows that the Kernel classifier is sensitive to changes in normalizer τ and the k-th nearest neighbor $\mathbf{x}_k \leftrightarrow \sigma_k$ in the kernel/correlation function $k(\mathbf{x}_n, \mathbf{x}_m, \tau) = e^{-\|\mathbf{x}_n - \mathbf{x}_m\|^2/\tau\sigma_n\sigma_m}$, which is expected because the kernel function defines the classification mechanism of the Kernel classifier. In accordance to hypothesis, the Kernel classifier is an effective model to classify or cluster the dataset $\mathcal{D}_{seeds}$. However, it appears that the classifier $\mathbf{y}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{T}$ is limited to a certain accuracy ($\sim 90\%$) before the Kernel method cannot be further optimized in the context of cluster analysis, perhaps because the Kernel classification technique is unintelligent to particular outliers in the Kama or Canadian wheat species regardless of parameter tuning. Instead, pure k-means and gradient clustering techniques produce superior classification accuracies that average to $\sim 94\%$. [2] In any case, we conclude that Kernel classification is a viable and relatively accurate method to classify or cluster approximately separable multi-class datasets.

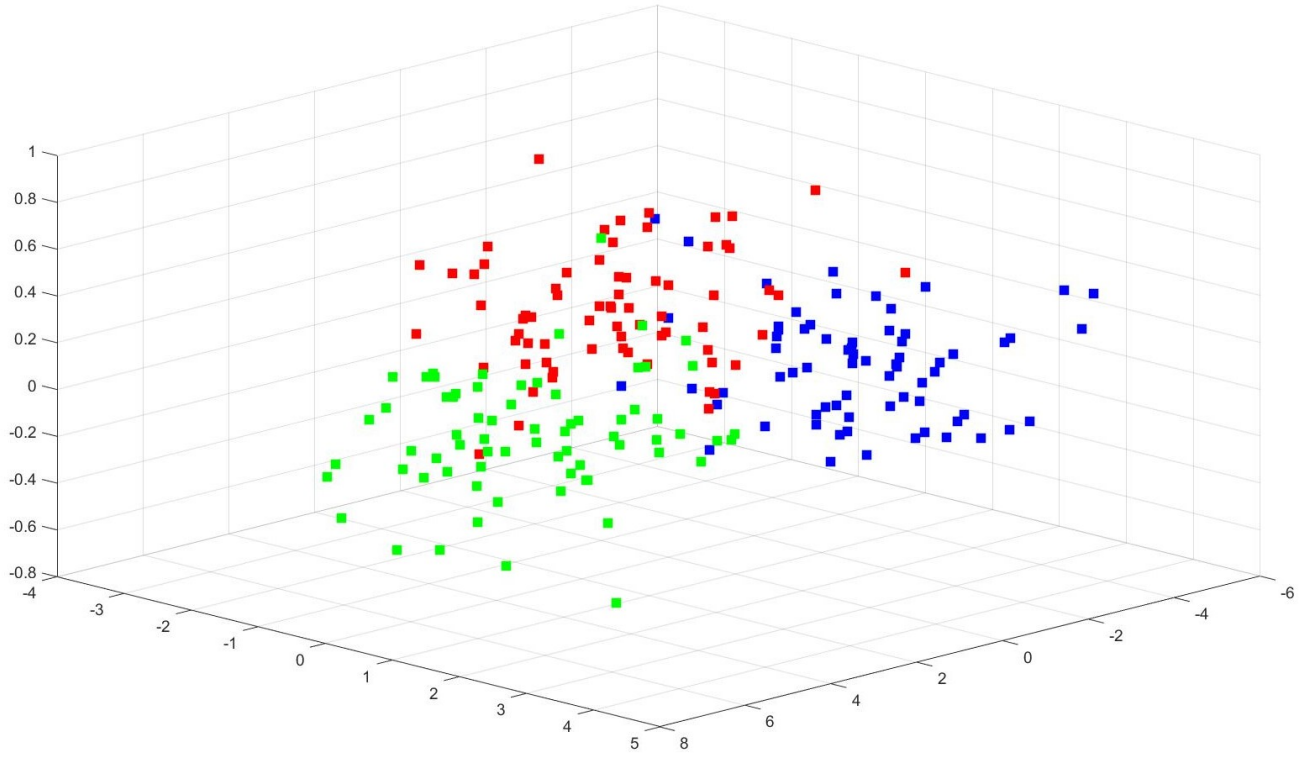

Fig. 4 – 3-D PCA Projection and Classification of $\mathcal{D}$

Extra: Limitations of Kernel Classifier

Attempting to apply the Kernel classifier to an inseparable/disconnected dataset produces fascinating results. **Fig. 5** demonstrates a dense dataset $\mathcal{D}_{wine} \subset \mathbb{R}^{11}$ that cannot be clustered or classified with the Kernel classifier. In particular, there exists a unknown wine-quality function $q(\mathbf{x}) \in \{0, 1, \ldots, 10\}^{\mathbb{R}^{11}}$ that measures the quality of wine from 11 separate attributes, various combinations of which can simulate high-quality wine. To learn the multi-variate function/classifier $q(\mathbf{x})$, a multi-layer neural network or advanced regression model beyond the scope of the Kernel method is necessary.

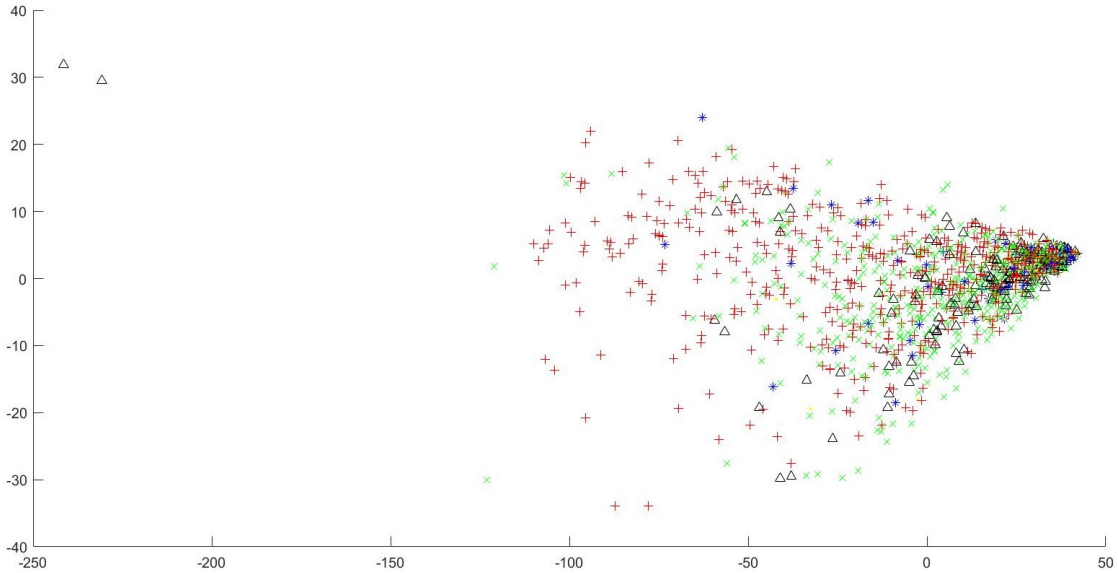

Fig. 5 – PCA Projection of Inseparable Wine Quality Dataset $\mathcal{D}_{wine}$

References

[1] Charytanowicz, Niewczas, Kulczycki, et al. "*seeds* Data Set." UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, September 29, 2012.
Reference Link: <`http://archive.ics.uci.edu/ml/datasets/seeds`>

[2] M. Charytanowicz, J. Niewczas, P. Kulczycki, P.A. Kowalski, S. Lukasik, S. Zak, 'A Complete Gradient Clustering Algorithm for Features Analysis of X-ray Images', in: Information Technologies in Biomedicine, Ewa Pietka, Jacek Kawa (eds.), Springer-Verlag, Berlin-Heidelberg, 2010, pp. 15-24.