Statistical Learning - Classification

Assignment 2

RBF Networks, Risk Estimation, Kernels, and SVMs

Due: Monday, November 9, 2026, 11:59 p.m. **100 marks**
Time zone: America/Toronto (Eastern Time).

Written work. Show the main steps of each derivation and explain your conclusions. For experiments, include the requested tables and figures and refer to them in your explanations. Report classification error as the number of incorrect predictions divided by the number of observations. Label plot axes, curves, and contours. Questions 2 and 4 require written answers only.

Submission. This is an individual assignment. Upload written answers, tables, and figures to the matching questions in Crowdmark. Submit your code and results to the **Assignment 2 Dropbox** in Learn as one `WatIAM_A2.zip`, replacing `WatIAM` with your Waterloo username. Include the main scripts `Q1.py`, `Q3.py`, `Q5.py`, and `Q6.py` (or their `.R` or `.m` equivalents), all required helper and input files, and the requested results CSVs. Include a short README listing run commands, dependencies, random seeds, and solver settings. After extracting the ZIP, it must be possible to run each main script without first running another question. Submit full code in the ZIP.

Programming. Use Python, R, or MATLAB. You may use libraries for array operations, QR/SVD, linear solves, random-number generation, CSV input/output, and plotting. Implement the requested RBF fits, risk calculations, kernel evaluations, and prediction rules yourself. In Questions 3 and 6, you may use a generic constrained-optimization or quadratic-programming solver, but you must construct its objective and constraints. Do not use ready-made RBF, SVM, or model-selection routines. Solver guidance follows Question 3. Required results filenames and column names appear at the end.

Data and notation. Extract the accompanying data ZIP and read its README for file layouts and loading examples. Each observation is a column: $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$. Uppercase letters denote matrices and bold lowercase letters denote vectors. All vector norms are Euclidean. Classification labels are -1 and $+1$; a score of zero predicts $+1$. Use test data only to evaluate the prescribed models. Do not use them to fit models, select models, or adjust solver settings.

Generative AI survey (required). AI use is permitted. After completing the assignment, open Learn and choose **Submit** $\rightarrow$ **Surveys** $\rightarrow$ **Assignment 2: Generative AI Use Survey**. Everyone must complete it, including students who used no generative AI (answer **No** to all five questions). Responses do not affect your assignment grade. The survey is not anonymous, so completion can be checked.

Individual Verification of Understanding. You may be asked to complete a brief individual quiz or oral discussion about your assignment, including explaining your reasoning, answering related questions, or explaining and making small modifications to your code. You must demonstrate your own understanding of all submitted work, including work developed with permitted generative AI assistance. These checks must be completed without generative AI or assistance from others. If you cannot demonstrate adequate understanding, your overall assignment mark may be reduced.

Late Submission Policy. Submissions received up to 24 hours after the deadline receive a 10% penalty on the earned mark. Submissions more than 24 hours late receive a grade of zero.

Question 1: Implementing RBF regression and selecting model size

20 marks

Use $n = 80$ equally spaced inputs and the following true regression function:

$$x_i = -1 + \frac{2(i-1)}{n-1}, \quad i = 1, \ldots, n, \qquad f(x) = \sin(2\pi x) + \tfrac{1}{2}\cos(5\pi x).$$

Generate noisy responses from

$$y_i = f(x_i) + \epsilon_i, \qquad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2), \qquad \sigma = 0.3.$$

Choose and report a fixed random seed for generating the noise. Then fit all four models, with $m = 5, 9, 17, 33$ hidden units, to the **same generated data**. For each model, define m basis functions by

$$\mu_j = -1 + \frac{2(j-1)}{m-1}, \quad \phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{h^2}\right), \quad h = 0.12, \quad \Phi_{ji} = \phi_j(x_i).$$

Here $j = 1, \ldots, m$ and $i = 1, \ldots, n$, so $\Phi \in \mathbb{R}^{m \times n}$. The fitted function and the least-squares problem used to estimate its weights are

$$\widehat{f}_m(x) = \sum_{j=1}^{m} \widehat{w}_j \phi_j(x), \qquad \min_{\boldsymbol{w} \in \mathbb{R}^m} \|\boldsymbol{y} - \Phi^T \boldsymbol{w}\|_2^2.$$

Use a QR- or SVD-based least-squares routine, without an intercept or regularization. Do not explicitly invert $\Phi\Phi^T$. All four design matrices have full row rank. For each m, keep its centers and width fixed across repetitions. With five narrow basis functions, separated bumps in the fitted curve are expected.

Error measures. Let $\widehat{\boldsymbol{f}}_m = (\widehat{f}_m(x_1), \ldots, \widehat{f}_m(x_n))^T$ and $\boldsymbol{f} = (f(x_1), \ldots, f(x_n))^T$. For each fitted model, calculate

$$e_m = \frac{\|\boldsymbol{y} - \widehat{\boldsymbol{f}}_m\|_2^2}{n}, \qquad s_m = e_m - \sigma^2 + \frac{2\sigma^2 m}{n}, \qquad q_m = \frac{\|\widehat{\boldsymbol{f}}_m - \boldsymbol{f}\|_2^2}{n}.$$

Here e_m measures error against the noisy responses, while q_m measures error against the true function values. For each fixed m, s_m is Stein's unbiased risk estimator (SURE) for the expected value of q_m. Use f to generate data and evaluate fits in every part. Use SURE to select the reported model; the choices based on e_m and q_m in part (b) are comparisons only. Retain negative SURE values as calculated.

(a) **[3]** Implement the design matrix and least-squares fit. State the dimensions of Φ^T, $\boldsymbol{w}$, $\boldsymbol{y}$, and $\widehat{\boldsymbol{f}}_m$. For each m, report the smallest singular value of Φ divided by its largest singular value. Use these ratios to identify which model's estimated weights are most sensitive to small relative perturbations. Explain briefly.

(b) **[5]** Use your first generated response vector for all four fits. Report a four-row table containing m, the singular-value ratio, and e_m, s_m, q_m. Select $\widehat{m}$ as the value of m with the smallest SURE, choosing the smaller m if there is a tie. For comparison, also report which m minimizes e_m and which minimizes q_m.

Plot the true function and all four fitted curves on a fine grid over $[-1, 1]$, and add the observed data points. Label each curve and identify the SURE-selected model. Submit one figure; use four aligned panels if a single panel is crowded.

Question 1: repeated experiments and interpretation

(c) **[6]** Run $b = 200$ repetitions in total. Use the response vector from part (b) as repetition 1, then generate 199 new independent noise vectors and their corresponding responses. Keep the inputs fixed. Set the random seed once and continue drawing noise; do not reset it inside the repetition loop. In each repetition, fit all four models to the **same response vector** and select m by SURE, using the tie rule in part (b).

For each m, average e_m, s_m, q_m over all 200 repetitions. Report these averages and the number of times SURE selected that m. Let $s_{t,m}$ and $q_{t,m}$ be the values in repetition t, and define $d_{t,m} = s_{t,m} - q_{t,m}$. Also report

$$\overline{d}_m = \frac{1}{b}\sum_{t=1}^{b} d_{t,m}, \qquad \mathrm{mcse}_m = \frac{1}{\sqrt{b}}\sqrt{\frac{\sum_{t=1}^{b}(d_{t,m} - \overline{d}_m)^2}{b-1}}.$$

The Monte Carlo standard error measures simulation uncertainty in $\overline{d}_m$. Put the results in one four-row table. Make one figure with the average training error, average SURE, and average true-function error plotted against m.

(d) **[3]** Use your tables and figures to answer the following briefly:

- Compare the model sizes that minimize average training error and average true-function error. Explain the difference, if any, in terms of approximation ability and sensitivity to noise.
- Compare each mean difference $\overline{d}_m$ with its Monte Carlo standard error. Are the observed differences small relative to simulation uncertainty? Explain what you observe; exact agreement is not expected.
- Why can you calculate q_m in this simulation but generally not with real data? Why can changing the noise change the SURE-selected model? Interpret your selection counts, even if the same model is selected in every repetition.

(e) **[3]** Consider any one of the four models. Add one hidden unit with exactly the same center and width as an existing unit. Show that this does not change the set of response vectors the network can fit, or the unique least-squares fitted vector. Explain why the weights are no longer unique and why the enlarged design matrix loses full row rank.

Let H be the orthogonal projection onto the original space of fitted response vectors, so that $\widehat{\boldsymbol{f}} = H\boldsymbol{y}$ and $\mathrm{tr}(H) = m$. Explain why H and $\mathrm{tr}(H)$ stay the same after duplication. Should the SURE correction use m or $m+1$? An argument that combines the two coefficients of the identical units is sufficient. No additional computation is required.

Deliverables for Question 1
Crowdmark Q1: The dimensions and singular-value comparison, two results tables, model choices, two figures, your seed, and the explanations and duplicate-unit argument.
Learn: `Q1.py`, `Q1.R`, or `Q1.m`; `Q1_first.csv`; and `Q1_simulation.csv`. Use the column names in the results-file specifications. No input file is needed.

Question 2: Shrinkage, risk, and the SURE correction

12 marks

Let $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon}$, where $\boldsymbol{f} \in \mathbb{R}^n$ is fixed and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$ with known $\sigma > 0$. Let $H \in \mathbb{R}^{n \times n}$ be a fixed orthogonal projection, independent of $\boldsymbol{y}$, with

$$H^T = H, \qquad H^2 = H, \qquad \operatorname{tr}(H) = m, \qquad 1 \le m \le n.$$

Consider shrinking the fitted vector toward zero:

$$\widehat{\boldsymbol{f}}_c = cH\boldsymbol{y}, \qquad 0 \le c \le 1.$$

In parts (a)–(c), treat c as fixed before observing $\boldsymbol{y}$. Every expectation is over the noise $\boldsymbol{\epsilon}$, with $\boldsymbol{f}$ and H fixed. Use *total* squared error throughout this question; do not divide by n.

You may use the SURE identity

$$s(\boldsymbol{y}) = \|\boldsymbol{y} - \widehat{\boldsymbol{f}}\|_2^2 - n\sigma^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial \widehat{f}_i}{\partial y_i}, \qquad \mathbb{E}[s(\boldsymbol{y})] = \mathbb{E}\|\widehat{\boldsymbol{f}} - \boldsymbol{f}\|_2^2,$$

This identity applies to sufficiently regular estimators. You may use it without proof.

(a) **[4]** Derive the risk $r(c) = \mathbb{E}\|cH\boldsymbol{y} - \boldsymbol{f}\|_2^2$. Separate it into three terms: squared approximation bias, squared bias introduced by shrinkage, and variance. Express the result using $\|(I_n - H)\boldsymbol{f}\|_2^2$, $\|H\boldsymbol{f}\|_2^2$, c, m, and σ^2.

Explain where you use orthogonality and where you use the zero mean of the noise. For the cross term between bias and noise, distinguish being zero for every noise realization from having zero expectation. Discuss $0 < c < 1$ and $c = 1$ separately.

(b) **[3]** Substitute $\widehat{\boldsymbol{f}}_c = cH\boldsymbol{y}$ into the supplied SURE identity to obtain $s_c(\boldsymbol{y})$. Calculate $\operatorname{tr}(cH)$ and $\operatorname{tr}((cH)^T(cH))$. Identify which trace appears in the SURE correction and which appears in the variance term. Explain why the two traces need not be equal.

(c) **[3]** Let $a = \|H\boldsymbol{f}\|_2^2$ and $b = \|(I_n - H)\boldsymbol{f}\|_2^2$. For this part, use

$$r(c) = b + (1 - c)^2 a + c^2 \sigma^2 m.$$

Find the value $c_{\text{oracle}} \in [0, 1]$ that minimizes $r(c)$. This is called an oracle choice because it may use the unknown true function. State its value when $a = 0$ and describe its behavior when a is large relative to $\sigma^2 m$. Explain why the observed responses alone generally do not determine this value.

(d) **[2]** Suppose $\widehat{c}(\boldsymbol{y})$ is chosen by minimizing $s_c(\boldsymbol{y})$ over $[0, 1]$ on the observed data, and the minimum value $s_{\widehat{c}}(\boldsymbol{y})$ is used as a risk estimate. Does unbiasedness of s_c for every fixed c guarantee that this minimum is an unbiased estimate of

$$\mathbb{E}\|\widehat{c}(\boldsymbol{y})H\boldsymbol{y} - \boldsymbol{f}\|_2^2?$$

Explain the difference between fixing c before observing the data and choosing it from those data. Give a conceptual argument only; no calculation of $\widehat{c}$ or selection correction is required.

Deliverables for Question 2
Crowdmark Q2: The derivations and explanations for (a)–(d). No simulation is required.

Question 3: An SVM intercept penalty and the choice of origin

22 marks

Use `Q3_train.csv` (60 observations) to fit the models and `Q3_test.csv` (200 observations) to evaluate them. The training data are strictly linearly separable. In each file, the first row contains observation IDs and the first column contains row names. Read rows `x1` and `x2` as $X \in \mathbb{R}^{2\times n}$, and row `y` as the label vector. Do not standardize the features.

Let $\boldsymbol{z}_i$ denote the coordinates used to fit a model; these are specified in the table below. Use the score $g(\boldsymbol{z}) = \boldsymbol{\beta}^T\boldsymbol{z} + \beta_0$ and the optimization problem

$$\min_{\boldsymbol{\beta},\beta_0} \ \tfrac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \tfrac{\rho}{2}\beta_0^2 \quad \text{subject to } y_i(\boldsymbol{\beta}^T\boldsymbol{z}_i + \beta_0) \geq 1.$$

For $\rho = 0$, this is the usual hard-margin SVM. A positive ρ also penalizes the intercept.

(a) **[5]** Derive the Lagrange dual for $\rho > 0$. Introduce multipliers $\alpha_i \geq 0$ for the constraints $1 - y_i g(\boldsymbol{z}_i) \leq 0$. Write the Lagrangian, minimize it over $\boldsymbol{\beta}$ and β_0, and substitute those minimizing values. State the dual objective and all constraints. Explain why the dual for $\rho > 0$ has no constraint $\boldsymbol{y}^T\boldsymbol{\alpha} = 0$. Also give the formulas that recover $\boldsymbol{\beta}$ and β_0 from $\boldsymbol{\alpha}$. Show these steps; quoting the computational formulas below is not sufficient.

Models to fit. Fit the following three models to the same training observations and labels. For the translated model, subtract $\boldsymbol{a}$ from both training and test inputs. This changes the coordinate origin; the labels stay the same.

Model name	Fitting coordinates	Intercept penalty
`standard`	$\boldsymbol{z} = \boldsymbol{x}$	$\rho = 0$
`penalized`	$\boldsymbol{z} = \boldsymbol{x}$	$\rho = 50$
`translated`	$\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{a}$, $\boldsymbol{a} = (-2, 2)^T$	$\rho = 50$

Computational formulas. Define $G_{ij} = y_i y_j \boldsymbol{z}_i^T\boldsymbol{z}_j$. For `standard`, maximize

$$\ell_d = \mathbf{1}^T\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^T G\boldsymbol{\alpha}, \qquad \boldsymbol{\alpha} \geq \mathbf{0}, \quad \boldsymbol{y}^T\boldsymbol{\alpha} = 0.$$

For either model with $\rho = 50$, maximize

$$\ell_d = \mathbf{1}^T\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^T G\boldsymbol{\alpha} - \frac{(\boldsymbol{y}^T\boldsymbol{\alpha})^2}{2\rho}, \qquad \boldsymbol{\alpha} \geq \mathbf{0},$$

with **no equality constraint**. There is no upper bound on any multiplier. If your solver minimizes an objective, supply $-\ell_d$. The gradient of $-\ell_d$ is $G\boldsymbol{\alpha} - \mathbf{1}$ for `standard`, and $G\boldsymbol{\alpha} - \mathbf{1} + \boldsymbol{y}(\boldsymbol{y}^T\boldsymbol{\alpha})/\rho$ for the penalized models.

After solving for $\boldsymbol{\alpha}$, set $\boldsymbol{\beta} = \sum_i \alpha_i y_i \boldsymbol{z}_i$. In all three models, count observation i as a numerical support vector when $\alpha_i > 10^{-6}$, and let $\mathcal{S}$ be the set of these indices. For `standard`, recover $\beta_0 = |\mathcal{S}|^{-1}\sum_{i\in\mathcal{S}}(y_i - \boldsymbol{\beta}^T\boldsymbol{z}_i)$. For a penalized model, use $\beta_0 = \boldsymbol{y}^T\boldsymbol{\alpha}/\rho$. Do not insert $\rho = 0$ into a formula containing $1/\rho$.

Question 3: fitting, numerical checks, and interpretation

Reporting in original coordinates. The translated model's score at an original input is

$$g(\boldsymbol{x}-\boldsymbol{a})=\boldsymbol{\beta}^T\boldsymbol{x}+\widetilde{\beta}_0, \qquad \widetilde{\beta}_0=\beta_0-\boldsymbol{\beta}^T\boldsymbol{a}.$$

For the other two models, $\widetilde{\beta}_0=\beta_0$. Make all plots using the original coordinates $\boldsymbol{x}$ and intercept $\widetilde{\beta}_0$. When evaluating the optimization objective, use the fitted intercept β_0.

(b) **[6]** Fit all three models using a permitted generic solver. Write fitting and prediction functions that accept data matrices with any compatible number of observations. For each model, report $\boldsymbol{\beta}$, β_0, $\widetilde{\beta}_0$, the one-sided geometric margin $1/\|\boldsymbol{\beta}\|_2$, the support-vector count, and the training and test misclassification fractions. A zero score predicts $+1$.

(c) **[4]** Check the numerical solution for each model. In its fitting coordinates, calculate the primal objective ℓ_p and the two numerical checks below:

$$\ell_p=\tfrac{1}{2}\|\boldsymbol{\beta}\|_2^2+\tfrac{\rho}{2}\beta_0^2, \qquad v_{\text{gap}}=\frac{|\ell_p-\ell_d|}{\max\{1,|\ell_p|,|\ell_d|\}},$$

$$v_{\text{feas}}=\max\{0,-\min_i \alpha_i,\max_i[1-y_i g(\boldsymbol{z}_i)]\}.$$

Here ℓ_d is the dual objective evaluated at the computed multipliers. For `standard`, also include $|\boldsymbol{y}^T\boldsymbol{\alpha}|$ among the quantities in the maximum defining v_{feas}. The target is $v_{\text{feas}}\le 10^{-5}$ and $v_{\text{gap}}\le 10^{-5}$. Report both values and the solver's stopping status. If a check fails, investigate and explain the failure; do not change the optimization problem to make the check pass. Explain why zero training error alone does not establish optimality.

(d) **[3]** Make one figure with three panels, one per model. Use the original coordinates and the same limits, $[-2.8, 2.8]$ on each axis. Show the training observations, outline the support vectors for that model, and draw and label the score contours at 0, -1, and $+1$.

(e) **[4]** Show algebraically that translating all inputs in the ordinary hard-margin SVM ($\rho=0$) preserves the decision boundary when expressed in the original coordinates. No additional fit is required. Explain why penalizing the intercept removes this invariance. Use your table and plots to compare all three models, discussing the effects of both the intercept penalty and the translation.

Does a smaller input-space margin necessarily mean that a penalized fit failed? Do equal test-error fractions mean that two classifiers are identical? Explain each answer briefly.

Deliverables for Question 3
Crowdmark Q3: The dual derivation, one three-row results table (which may be split for readability), one three-panel figure, solver details, and the explanations. **Learn:** The Q3 script, the two input files, and `Q3_results.csv`.

Solver guidance for Questions 3 and 6

- **Python:** `scipy.optimize.minimize` with `method="SLSQP"`, analytic derivatives, zero initial multipliers, `ftol=1e-11`, and `maxiter=5000` is a tested starting configuration.
- **R:** `nloptr::slsqp` supports bounds and equality constraints through `heq` and `heqjac`. Use the diagnostics to set accuracy. `quadprog::solve.QP` requires a positive-definite quadratic term and is unsuitable for the singular matrices in Question 3.
- **MATLAB:** `quadprog` with `Algorithm="interior-point-convex"` is a suitable generic option; it requires Optimization Toolbox.

Supply objective and constraint derivatives where the solver supports them. Its solution vector gives $\boldsymbol{\alpha}$. Check the reported feasibility residual and primal–dual gap even when the solver reports success.

Question 4: Information loss and separation with quadratic features

12 marks

Consider the feature map and its associated kernel

$$\boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2)^T, \qquad k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{z}) = (\boldsymbol{x}^T \boldsymbol{z})^2.$$

The score is $g(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \beta_0$. You may use the displayed kernel identity without proof. Answer all parts analytically; no coding or numerical optimization is required.

(a) **[3]** Show that $g(\boldsymbol{x}) = g(-\boldsymbol{x})$ for every choice of $\boldsymbol{\beta}$ and β_0. What information about the input is lost in the feature map? If a nonzero input $\boldsymbol{u}$ and its negative $-\boldsymbol{u}$ have opposite labels, can this model classify both correctly? Justify your answer.

(b) **[5]** Consider these four labeled observations:

Label	Inputs
$+1$	$(1,1)^T,\ (-1,-1)^T$
-1	$(1,-1)^T,\ (-1,1)^T.$

Find a minimum-norm hard-margin separator in the feature space:

$$\min_{\boldsymbol{\beta},\beta_0} \tfrac{1}{2}\|\boldsymbol{\beta}\|_2^2 \quad \text{subject to } y_i g(\boldsymbol{x}_i) \geq 1.$$

Give $\boldsymbol{\beta}$, β_0, and the one-sided margin $1/\|\boldsymbol{\beta}\|_2$ in feature space. To prove optimality, find a lower bound on $\|\boldsymbol{\beta}\|_2$ for every feasible separator, then show that your separator attains it. In the original input plane, describe the boundary $g(\boldsymbol{x}) = 0$ and the predicted label in each open quadrant (excluding the axes).

Hint: The four observations give only two distinct feature vectors. Compare their two margin inequalities and use $\|\boldsymbol{\beta}\|_2 \geq |\beta_3|$.

(c) **[2]** Prove that no affine score $\boldsymbol{w}^T\boldsymbol{x} + w_0$ can be strictly positive on both positive-label observations in part (b) and strictly negative on both negative-label observations. Work in the original input space. A short argument using the mean of each class is sufficient.

(d) **[2]** Now extend the feature vector to

$$\boldsymbol{\psi}(\boldsymbol{x}) = (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2, x_1, x_2)^T.$$

Use the score $g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\psi}(\boldsymbol{x}) + w_0$. For $(1,0)^T$ with label $+1$ and $(-1,0)^T$ with label -1, give a five-dimensional vector $\boldsymbol{w}$ and an intercept w_0 such that $y_i g(\boldsymbol{x}_i) \geq 1$ for both observations. Explain which added information allows the model to distinguish this pair. Any feasible coefficients are acceptable; minimum norm is not required.

Deliverables for Question 4
Crowdmark Q4: The arguments, coefficient vectors, the margin from part (b), and the boundary description. No figure is required.

Question 5: Implementing kernel prediction and comparing boundaries

14 marks

Four trained models are supplied. Implement their prediction rules, evaluate their errors, and plot their decision regions. No model fitting is required.

The observations are in `Q5_train.csv` (40 training points) and `Q5_test.csv` (200 test points). The model parameters are in `Q5_models.csv` and `Q5_model_info.csv`. In each observation file, the first row contains IDs and the first column contains row names. Read rows `x1` and `x2` as $X \in \mathbb{R}^{2\times n}$, and row `y` as the labels. Use the original coordinates.

The four supplied hard-margin models use

$$k_{\text{quad}}(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^T \boldsymbol{z})^2, \qquad k_h(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|_2^2}{h^2}\right), \quad h \in \{0.2, 0.7, 1.4\}.$$

The Gaussian convention has **no factor of 2** in the denominator. The training set is separable in each of these feature spaces.

`Q5_models.csv` contains one row of multipliers for each model. Match its coefficient columns to the training observation IDs. Each entry is α_i; multiply it by y_i exactly once in the score below. `Q5_model_info.csv` has one row per model and columns for the model name, kernel, width, and intercept β_0. The README lists the row names. Use the supplied coefficients and intercepts unchanged.

For a new input $\boldsymbol{z}$, calculate the score

$$g(\boldsymbol{z}) = \beta_0 + \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{z}).$$

The sum runs over the training observations. Predict $+1$ for $g \geq 0$ and -1 otherwise. These scores are not probabilities. Count observation i as a numerical support vector when $\alpha_i > 10^{-6}$.

(a) **[4]** Implement both kernels and a prediction rule that accepts a matrix of new observations. For each model, form the training Gram matrix K, with entries $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and check its symmetry. State the dimensions of K and of the cross-kernel matrix with entries $k(\boldsymbol{x}_i, \boldsymbol{z}_j)$, where $\boldsymbol{x}_i$ is training observation i and $\boldsymbol{z}_j$ is test observation j. Keep the coefficients aligned with the training IDs.

(b) **[3]** Make one four-row table with training error, test error, support-vector count, and symmetry error $\max_{i,j} |K_{ij} - K_{ji}|$ for each model. State whether every model classifies all training observations correctly.

(c) **[4]** Make one figure with four panels, one per model. Use the same limits, $[-2.2, 2.2]$ on each axis, and a grid with at least 80 points per axis. Color the regions by predicted class, show the training observations, and outline the support vectors. Draw and label the contours $g = 0$, $g = -1$, and $g = +1$ wherever they occur. Use consistent class colors and label each model.

(d) **[3]** Use your table and plots to explain how Gaussian width affects the decision boundary and the number of support vectors. Explain how models can agree on every training label but predict differently on test observations. Explain why the $g = -1$ and $g = +1$ contours need not be equally far from $g = 0$ when Euclidean distance is measured in the original input plane. Evaluate all four models; do not use test labels to select a model or adjust its width.

Deliverables for Question 5
Crowdmark Q5: Matrix dimensions, one four-row results table, one four-panel figure, and explanations. **Learn:** The Q5 script, four supplied input files, and `Q5_results.csv`, using the specified columns.

Question 6: Implementing soft margins with altered labels

20 marks

Use `Q6_train.csv`, `Q6_test.csv`, `Q6_hard_grid.csv`, and `Q6_hard_summary.csv`.

The training file contains 40 two-dimensional observations. Four labels, at IDs `train004`, `train009`, `train024`, and `train029`, have been reversed. **Use these labels as supplied; do not reverse them again.** The test file contains 200 observations whose labels have not been altered. In both files, observations are columns; rows `x1` and `x2` contain features and row `y` contains labels. Use the original coordinates. In the formulas below, y_i always means the label supplied in `Q6_train.csv`.

Use the Gaussian kernel

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|_2^2}{0.7^2}\right).$$

Write $g(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \beta_0$, where $\boldsymbol{\phi}$ is the kernel's feature map and $\boldsymbol{\beta}$ is its coefficient vector. For computation, use the kernel formulas below. The soft-margin problem is

$$\min_{\boldsymbol{\beta}, \beta_0, \boldsymbol{\zeta}} \ \tfrac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \gamma \sum_i \zeta_i, \qquad y_i g(\boldsymbol{x}_i) \geq 1 - \zeta_i, \quad \zeta_i \geq 0.$$

The slack ζ_i relaxes the unit-margin requirement $y_i g(\boldsymbol{x}_i) \geq 1$. The penalty is γ times the **sum** of the slacks. Fit three models, with $\gamma = 0.2, 1, 10$, to the **same supplied training data**. For each value, solve the dual problem

$$\max_{\boldsymbol{\alpha}} \ \ell_d = \sum_i \alpha_i - \tfrac{1}{2}\boldsymbol{\alpha}^T G \boldsymbol{\alpha}, \quad G_{ij} = y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad 0 \leq \alpha_i \leq \gamma, \quad \boldsymbol{y}^T \boldsymbol{\alpha} = 0.$$

If your solver minimizes, supply $-\ell_d$ and its gradient $G\boldsymbol{\alpha} - \mathbf{1}$. After solving for $\boldsymbol{\alpha}$, recover the intercept from the *free support vectors*, whose multipliers are strictly between the bounds:

$$\mathcal{F} = \{i : 10^{-6} < \alpha_i < \gamma - 10^{-6}\}, \qquad \beta_0 = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} \left[y_i - \sum_j \alpha_j y_j k(\boldsymbol{x}_j, \boldsymbol{x}_i) \right].$$

Each prescribed fit has free support vectors. If your computed set $\mathcal{F}$ is empty, check your implementation and solver accuracy. To predict at a new input $\boldsymbol{z}$, use all computed coefficients in

$$g(\boldsymbol{z}) = \beta_0 + \sum_i \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{z}).$$

Predict $+1$ when $g \geq 0$ and -1 otherwise. The support-vector count includes every $\alpha_i > 10^{-6}$; the free-support count includes only the indices in $\mathcal{F}$.

(a) **[5]** Consider two penalties, $0 < \gamma_1 < \gamma_2$, with the same training observations, labels, and kernel. Let $(\boldsymbol{\beta}^{(j)}, \beta_0^{(j)}, \boldsymbol{\zeta}^{(j)})$ be a global optimum for penalty γ_j, for $j = 1, 2$. Define the total slack and half the squared coefficient norm by

$$u_j = \sum_i \zeta_i^{(j)}, \qquad v_j = \tfrac{1}{2}\|\boldsymbol{\beta}^{(j)}\|_2^2.$$

Compare the two solutions under the objective with penalty γ_1, then under the objective with penalty γ_2. Use the resulting two optimality inequalities to prove

$$u_2 \leq u_1, \qquad v_2 \geq v_1.$$

Do these conclusions imply that training or test misclassification must decrease as γ increases? Explain what total slack measures and how it differs from a misclassification count. Your proof must allow nonunique optima. You do not need to construct a counterexample.

Question 6: checks, comparison, and interpretation

(b) **[2]** Consider a training set that is separable in the feature space of a fixed kernel. Let $(\boldsymbol{\beta}^{\text{hard}}, \beta_0^{\text{hard}})$ solve the hard-margin primal problem, and let $\boldsymbol{\alpha}^{\text{hard}}$ solve its dual. Prove that, whenever

$$\gamma \geq \max_i \alpha_i^{\text{hard}},$$

the hard-margin solution with zero slacks, $(\boldsymbol{\beta}^{\text{hard}}, \beta_0^{\text{hard}}, \mathbf{0})$, also solves the soft-margin primal problem. The hard-margin dual has the same objective and equality constraint as the soft-margin dual, but no upper bound on its multipliers. You may use strong duality for the hard-margin problem: its optimal primal and dual objective values are equal.

Numerical checks. For each soft-margin fit, set $\zeta_i = \max\{1 - y_i g(\boldsymbol{x}_i), 0\}$ and $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Calculate the primal objective

$$\ell_p = \tfrac{1}{2}(\boldsymbol{\alpha} \odot \boldsymbol{y})^T K (\boldsymbol{\alpha} \odot \boldsymbol{y}) + \gamma \sum_i \zeta_i,$$

where $\odot$ means entrywise multiplication. Evaluate ℓ_d at the same computed multipliers, then calculate

$$v_{\text{feas}} = \max\{0, -\min_i \alpha_i, \max_i \alpha_i - \gamma, |\boldsymbol{y}^T \boldsymbol{\alpha}|\}, \qquad v_{\text{gap}} = \frac{|\ell_p - \ell_d|}{\max\{1, |\ell_p|, |\ell_d|\}}.$$

The target is $v_{\text{feas}} \leq 10^{-5}$ and $v_{\text{gap}} \leq 10^{-5}$. The chosen slacks automatically satisfy the primal inequalities. Report both checks and investigate any failure. Do not clip the computed multipliers or add a diagonal penalty to K to make the checks pass.

Supplied hard-margin comparison. A hard-margin model has already been fitted to the same altered training labels with Gaussian width 0.7. Use its supplied results; no hard-margin fit is required. Copy its training error, test error, and support-vector count from `Q6_hard_summary.csv`. Use the coordinates and scores in `Q6_hard_grid.csv` to plot its boundary. The README explains how to reshape the grid for plotting.

(c) **[6]** Implement and fit the three soft-margin models. Make one row of results for each γ. Report training error using the supplied altered labels, test error using the supplied test labels, support-vector count, free-support count, intercept, and total slack $\sum_i \zeta_i$. Include both numerical checks and report the solver's stopping status.

(d) **[3]** Add a fourth row for the supplied hard-margin model, using `hard` as its model name. Copy its errors and support count. Leave the fields identified as unavailable in the results-file specification blank.

Make one figure with four panels, one per model, over $[-2.2, 2.2]^2$. Use the supplied grid coordinates for every panel. Evaluate your three fitted models at those coordinates; use the supplied scores for the hard-margin panel. In every panel, show the training observations with their supplied labels, circle the four reversed-label observations, and draw the boundary $g = 0$.

(e) **[4]** Use your table and plots to explain how increasing γ affects the boundary, training error, test error, and total slack. Explain why fitting every altered training label can worsen predictions on the test set.

The intercept formula averages over free support vectors. Explain why averaging over all indices with $\alpha_i > 0$ would generally be incorrect. You may use $(\gamma - \alpha_i)\zeta_i = 0$ and $\alpha_i[1 - \zeta_i - y_i g(\boldsymbol{x}_i)] = 0$ without deriving these optimality conditions.

Deliverables for Question 6
Crowdmark Q6: The two proofs, one results table, one four-panel figure, solver settings and status, and explanations. **Learn:** The Q6 script, four supplied input files, and `Q6_results.csv`, using the specified columns.

Results-file specifications

Use the filenames below and put the columns in the order shown. Each results CSV must have one header row and one row per model, with no extra row-index column. Report classification errors as fractions and counts as integers. Save floating-point values with at least eight significant digits so the numerical checks can be assessed. Leave a field blank only where the specification says to do so. Put solver settings and stopping messages in your README and written answers.

`Q1_first.csv`: four rows, in increasing order of m.

Column	Meaning
`m`	Number of RBF units: 5, 9, 17, 33
`singular_ratio`	Smallest divided by largest singular value of Φ
`empirical_error`	e_m for the first response vector
`sure`	s_m for the first response vector
`latent_error`	q_m for the first response vector

`Q1_simulation.csv`: four rows, in increasing order of m.

Column	Meaning
`m`	Number of RBF units
`mean_empirical_error`	Average e_m over all 200 repetitions
`mean_sure`	Average s_m over all 200 repetitions
`mean_latent_error`	Average q_m over all 200 repetitions
`mean_difference`	$\overline{d}_m$, the average of $s_{t,m} - q_{t,m}$
`mcse_difference`	Monte Carlo standard error of $\overline{d}_m$
`selection_count`	Number of repetitions selecting m by SURE

The four selection counts must sum to 200. These files describe the four original full-row-rank designs, not the duplicate-unit construction in Q1(e).

`Q3_results.csv`: three rows, ordered `standard`, `penalized`, `translated`.

Column	Meaning
`model`	`standard`, `penalized`, or `translated`
`beta1`	First component of $\boldsymbol{\beta}$
`beta2`	Second component of $\boldsymbol{\beta}$
`beta0_fit`	Intercept β_0 in the fitting coordinates
`beta0_original`	Intercept $\widetilde{\beta}_0$ in the original coordinates
`margin`	One-sided geometric margin, $1/\|\boldsymbol{\beta}\|_2$
`support_count`	Number of multipliers greater than 10^{-6}
`train_error`	Training misclassification fraction
`test_error`	Test misclassification fraction
`v_feas`	Combined feasibility residual in Q3(c)
`v_gap`	Absolute relative primal–dual gap in Q3(c)

Results-file specifications (continued)

`Q5_results.csv`: four rows in the order shown below.

Column	Meaning
`model`	`quadratic`, `gaussian_02`, `gaussian_07`, or `gaussian_14`
`train_error`	Training misclassification fraction
`test_error`	Test misclassification fraction
`support_count`	Number of multipliers greater than 10^{-6}
`symmetry_error`	$\max_{i,j} \lvert K_{ij} - K_{ji} \rvert$ for the training Gram matrix

`Q6_results.csv`: three soft-margin rows in increasing γ, followed by the hard-margin reference row.

Column	Meaning
`model`	`soft` for the three fitted models; `hard` for the reference
`gamma`	Slack penalty: 0.2, 1, or 10; blank for the reference
`train_error`	Error fraction against the supplied altered training labels
`test_error`	Error fraction against the unchanged test labels
`support_count`	Number of multipliers greater than 10^{-6}
`free_count`	Number of free support vectors; blank for the reference
`beta0`	Intercept; blank for the reference
`slack_sum`	Total slack $\sum_i \zeta_i$; blank for the reference
`v_feas`	Dual feasibility residual; blank for the reference
`v_gap`	Absolute relative primal–dual gap; blank for the reference

Copy the reference row's errors and support count from `Q6_hard_summary.csv`. The kernel width is 0.7 for all four rows, so no width column is needed.