Задание №4 по курсу «Практикум на ЭВМ»

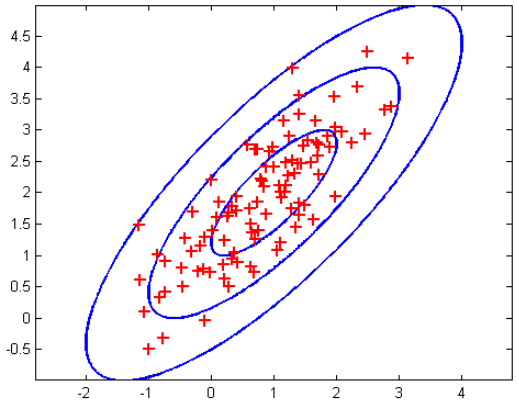
(обновлено 26 ноября)

Срок сдачи: 4 декабря (среда), 23:59

Максимальный балл: 5.0.

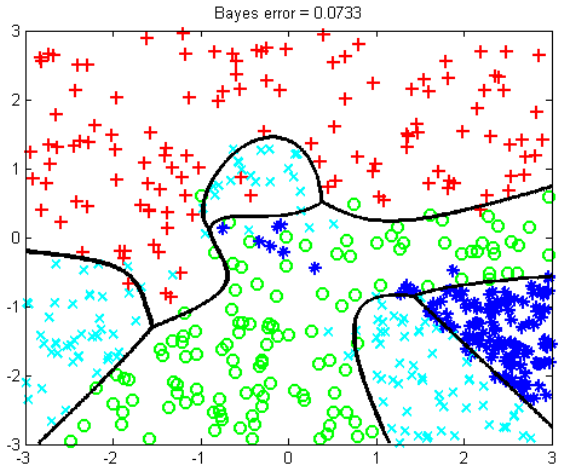
Формулировка задания

1. Для многомерного нормального распределения:
    a. Реализовать процедуру генерации выборки заданного объема на основе функции генерации из одномерного стандартного нормального распределения randn (функцией mvnrnd пользоваться нельзя);
    b. Реализовать процедуру отображения линий уровня плотности двумерного нормального распределения без использования функции contour и аналогичных ей;
    c. С помощью реализованных процедур отобразить линии уровня, соответствующие одному, двум и трем стандартным отклонениям, а также отобразить сгенерированную выборку.



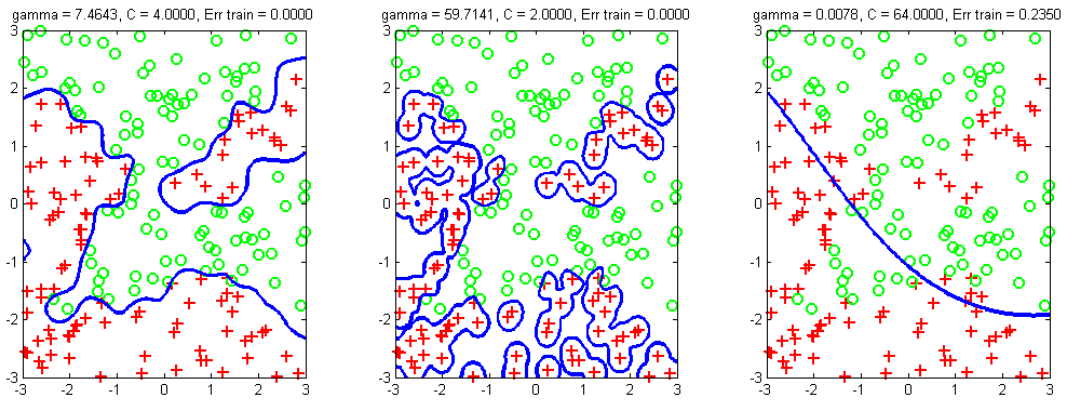
Пример рисунка с линиями уровня двумерного нормального распределения, соответствующие одному, двум и трем стандартным отклонениями, а также сгенерированная выборка.

2. Реализовать процедуру генерации модельных данных для задач классификации с нелинейными границами между классами; реализовать байесовский классификатор и процедуру оценки байесовского уровня ошибки (среднего риска байесовского классификатора) для выбранной модели генеральной совокупности.
3. Реализовать процедуру отображения разделяющей поверхности для заданного классификатора в двухмерном признаковом пространстве.



Пример модельных данных классификации с четырьмя классами; черной кривой показана разделяющая поверхность байесовского классификатора, уровень байесовской ошибки для этой задачи составляет 7,33%.

4. Для каждого из трех семейств классификаторов – метода k ближайших соседей, решающего дерева и метода опорных векторов (использовать готовые реализации методов, например, реализованные в MATLAB классы ClassificationKNN, ClassificationTree, а также библиотеку LIBSVM) – провести эксперимент по следующей схеме:
    a. Сгенерировать обучающие и тестовые двумерные модельные данные классификации;
    b. По обучающим данным провести процедуру кросс-валидации (скользящего контроля, бутстрапа) для настройки структурных параметров семейства алгоритмов; при этом построить график ошибки на кросс-валидации и на тестовых данных в зависимости от значения структурного параметра;
    c. Отобразить обучающую и тестовую выборку вместе с разделяющей поверхностью для трех классификаторов из семейства алгоритмов: 1) классификатора с параметрами, выбранными по кросс-валидации, 2) классификатора с высокой variance (соответствует значению структурного параметра, приводящего к модели максимальной сложности), 3) классификатора с высоким bias (соответствует значению структурного параметра, приводящего к модели минимальной сложности);
    d. Для каждого из трех классификаторов из предыдущего пункта построить кривые обучения (learning curves), т.е. графики ошибки на обучении и тесте в зависимости от объема обучающей выборки; на основе этих кривых сделать предположение о целесообразности увеличения объема обучающей выборки; проверить сделанное предположение экспериментально.



Пример графиков для пункта c: слева показана разделяющая поверхность оптимального классификатора, в середине – классификатора с высокой variance, справа – классификатора с высоким bias.

5. Реализовать процедуру вычисления TPR, FPR и AUC для заданного двухклассового классификатора и тестовой выборки (функциями roc, perfcurve и их аналогами пользоваться нельзя); построить на одном графике ROC-кривые для тестовой выборки для лучших классификаторов из каждого из трех семейств в эксперименте в п. 4.
6. Для каждого из трех семейств классификаторов – метода k ближайших соседей, решающего дерева и метода опорных векторов – провести эксперимент, аналогичный пункту 4, в котором выборки генерируются с несбалансированными классами (коэффициент несбалансированности может быть разным для обучающей и тестовой выборки), а вместо доли ошибок классификатора на выборке используется характеристика 1-AUC; при этом конкретный классификатор из ROC-кривой выбирается с помощью максимизации среднего гармонического чувствительности TPR и специфичности 1-FPR; сравнить полученные результаты со случаем использования характеристики доли ошибок на выборке.
7. Привести пример задачи классификации, в которой решающее дерево показывает лучший результат, чем метод опорных векторов и метод k ближайших соседей; аналогично привести примеры задач классификации, в которых два других семейства оказываются наилучшими; пояснить причины подобного поведения методов.
8. Написать отчет в среде LaTex с описанием всех проведенных исследований.

Отчет, а также все необходимые коды выслать преподавателю.

Требования к прототипам реализуемых функций

Генерация выборки из нормального распределения
X = gen_normal_sample(m, S, N)

m – мат.ожидание, вектор-столбец длины D;
S – матрица ковариации, матрица размера DxD;
N – объем генерируемой выборки, число;
X – сгенерированные данные, матрица размера NxD.

Рисование линий уровня двумерной нормальной плотности
plot_normal_contour(m, S, levels)

m – мат.ожидание, вектор-столбец длины 2;
S – матрица ковариации, матрица размера 2x2;
levels – линии уровня, измеряемые в количестве стандартных отклонений, вектор-столбец действительных чисел, по умолчанию = [1 2 3]’.

Создание модели генеральной совокупности
m = create_model(K, param1, param2, …)

K – число классов;
(param1, param2, …) – другие параметры модели (вид разделяющей поверхности, уровень шума, количество признаков и т.д.);
m - модель генеральной совокупности, структура.

Генерация выборки из модели генеральной совокупности
[X, t] = gen_sample(m, N)

m – модель генеральной совокупности, структура;
N – объем выборки, число;
X – сгенерированная выборка, матрица размера NxD;
t – метки классов для сгенерированной выборки, вектор-столбец длины N.

Рисование выборки
plot_sample(X, t)

X – выборка, матрица размера Nx2;
t – метки классов для выборки, вектор-столбец длины N.

Рисование разделяющей поверхности классификатора
plot_curve(m, borders)

m – обученная модель классификации, структура или объект класса;
borders – [minX, minY, maxX, maxY] – границы области рисования по первому и второму признаку.

Рисование разделяющей поверхности байесовского классификатора
plot_bayes_curve(m, borders)

m – модель генеральной совокупности, структура;
borders – [minX, minY, maxX, maxY] – границы области рисования по первому и второму признаку.

Оценка байесовского уровня ошибки
err = bayes_error(m, N)

m – модель генеральной совокупности, структура;
N – количество объектов выборки, используемых в методе Монте Карло, число, по умолчанию = 100000;
err – байесовский уровень ошибки, число.

Вычисление характеристик ROC-кривой
[TPR, FPR, AUC] = calculate_roc(t, outputs)

t – метки классов тестовой выборки, вектор-столбец длины N;
outputs – голоса за первый класс, вектор-столбец длины N;
TPR – true positive rate, вектор-столбец длины B, где B – количество различных порогов;
FPR – false positive rate, вектор-столбец длины B;
AUC – значение AUC, число.

Общий интерфейс к процедуре обучения алгоритма классификации
m = alg_train(alg_type, X, t, param_name1, param_value1, …)

alg_type – тип алгоритма классификации, строка, возможные значения ‘knn’, ‘dt’, ‘svm’;
X – обучающая выборка, матрица размера NxD;
T – метки классов для обучающей выборки, вектор-столбец длины N;
(param_name, param_value) – необязательный набор параметров классификатора, например:
‘NumNeighbors’ – количество соседей в методе k ближайших соседей;
‘MinLeaf’, ‘MinParent’ – минимальное число объектов в листе/узле дерева для решающего дерева;
‘gamma’, ‘C’ – параметр ядровой функции RBF и параметр регуляризации C в методе опорных векторов;
m – обученная модель классификации, структура или объект класса.

Общий интерфейс к процедуре прогнозирования алгоритма классификации
[t_predict, err, outputs] = alg_predict(m, X, t, use_auc)

m – обученная модель классификации, структура или объект класса;
X – тестовая выборка, матрица размера NxD;
t – метки классов для тестовой выборки, вектор-столбец длины N;
use_auc – флаг использования 1-AUC в качестве характеристики уровня ошибки, true или false;
t_predict – спрогнозированные метки классов для тестовой выборки, вектор-столбец длины N;
err – доля ошибок на тестовой выборке или 1-AUC, число;
outputs – оценки алгоритма за классы для каждого объекта выборки, матрица размера NxK.

Проведение эксперимента из п.4 или п.6 с рисованием всех необходимых графиков

knn_experiment(X_train, t_train, X_test, t_test, use_auc, partition_type)
dt_experiment(X_train, t_train, X_test, t_test, use_auc, partition_type)
svm_experiment(X_train, t_train, X_test, t_test, use_auc, partition_type)

X_train – обучающая выборка, матрица размера NxD;
t_train – метки классов для обучающей выборки, вектор-столбец длины N;
X_test – тестовая выборка, матрица размера N_testxD;
t_test – метки классов для тестовой выборки, вектор-столбец длины N_test;
use_auc – флаг использования характеристики 1-AUC, true или false;
partition_type – вид кросс-валидации, строка, возможные значения ’10-fold cv’, ‘5x2-fold cv’, ‘LOO’, ‘bootstrap’.