

The objective of this lab is for you to program in Matlab several types of support vector machines (SVMs) for binary classification, apply them to some datasets and observe their behavior. The TA will first demonstrate the results of the algorithms on a toy dataset, and then you will program them, replicate those results, and further explore the datasets with the algorithms. You can use the textbook, lecture notes and your own notes.

I Datasets

Construct your own toy datasets in 2D to visualize the result easily and be able to get the algorithm right, such as Gaussian classes with no or some overlap, or classes with curved shapes as in the 2moons dataset.

II Implementing and using SVMs

Most types of SVMs define a constrained optimization problem where the objective function is convex quadratic and the constraints (equalities or inequalities) are linear. Such problems are called convex quadratic programs (QPs). They have a unique solution, which can be found by solving either the original, primal QP, or the dual QP.

To solve a QP, you will use Matlab's Optimization Toolbox, specifically the following two functions:

- **quadprog**: this solves any kind of QP. All you have to do is put the QP for the SVM (primal or dual) in the form required by **quadprog** (see `help quadprog`). As output arguments, it returns everything you need to construct the SVM discriminant function $g(\mathbf{x})$ (the optimal solution and/or its Lagrange multipliers).
- **optimoptions**: this is not strictly necessary, but you can use it to select which QP solver to use and various other options or parameters (what to display, the maximum number of iterations, whether you want to provide an initialization, etc.). I suggest you use the active-set algorithm, because (with small problems) it reliably identifies the support vectors:

```
options = optimoptions('quadprog','Algorithm','active-set',MaxIter',1000);  
... = quadprog(...,options);
```

Then, implement the following types of SVM:

Linear SVM It has two types:

- *For data that is linearly separable: the optimal separating hyperplane.* You can solve either the primal and get (\mathbf{w}, w_0) , and hence the discriminant $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$, or the dual and get the Lagrange multiplier for each data point, α_n , and from there obtain the support vectors (SVs), which have $\alpha_n > 0$, and construct (\mathbf{w}, w_0) .
- *For data that is not linearly separable: the soft margin hyperplane.* Now the QP uses a slack variable $\xi_n \geq 0$ to account for possible constraint violations (points correctly classified but within the margin, or points misclassified), and a hyperparameter $C > 0$ that controls the tradeoff between minimizing the total violations $\sum_{n=1}^N \xi_n$ and maximizing the margin $\frac{1}{\|\mathbf{w}\|}$ (i.e., minimizing $\frac{1}{2}\|\mathbf{w}\|^2$). Again, you can solve either the primal or the dual.

Nonlinear (kernel) SVM You must choose what kernel $K(\mathbf{x}, \mathbf{y})$ to use (polynomial, Gaussian, etc.). In addition to C , the kernel may have its own hyperparameters (e.g. the degree q for the polynomial kernel or the width σ for the Gaussian kernel). With nonlinear SVMs you can only solve the dual QP, obtain the Lagrange multipliers $\alpha_1, \dots, \alpha_N \geq 0$, and from there construct the nonlinear discriminant

$$g(\mathbf{x}) = \sum_{n=1}^N \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}) = \sum_{n \in \text{SVs}}^N \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}).$$

In either case (linear or nonlinear), the discriminant function $g(\mathbf{x}) \in \mathbb{R}$ classifies an instance into the +1 or -1 class according to the sign of $g(\mathbf{x})$.

The hyperparameters C and (for kernel SVMs) q or σ are set by cross-validation over a suitable range of values; for example, try $C \in \{10^{-2}, 10^{-1}, 10^0, 10^1\}$ and $q \in \{1, 2, 3, 5, 10\}$. Plot the resulting SVM classifier for different hyperparameter values and observe how it looks like.

Implementation and exploration: toy problem Start with the simplest case: the optimal separating hyperplane. Find the SVM optimal parameters given the training set of instances $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^2$ and their class labels $\{y_n\}_{n=1}^N \subset \{-1, +1\}$, by running `quadprog` with appropriate arguments in the following two ways:

1. Solve the primal QP and obtain the SVM parameters (\mathbf{w}, w_0) directly. `quadprog` will also return the Lagrange multipliers, which you can use to find the SVs.
2. Solve the dual QP and obtain the Lagrange multipliers $\alpha_1, \dots, \alpha_N \geq 0$. From there, find the SVs (which have $\alpha_n > 0$), and construct (\mathbf{w}, w_0) .

To visualize the results, create the following plots:

- Plot the dataset in 2D, i.e., each training point \mathbf{x}_n colored according to its class label $y_n \in \{-1, +1\}$.
- To plot the SVM discriminant function $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$, use `contour`. The discrimination boundary corresponds to the contour $g(\mathbf{x}) = 0$. (You can also plot the line $\mathbf{w}^T \mathbf{x} + w_0 = 0$ directly, but the other contours are informative with nonlinear SVMs.)
- Plot the margin, indicated by the SVs on each side of the separating hyperplane.

Consider the following questions:

- Verify that solving the primal QP and the dual QP give the same result (same (\mathbf{w}, w_0) , Lagrange multipliers, SVs, margin).
- Verify that the margin (i.e., the distance from the hyperplane to its closest instance) is $\frac{1}{\|\mathbf{w}\|}$.
- When training the optimal separating hyperplane, what happens if the data is not linearly separable? *Hint*: look at the error code returned by `quadprog`.

Once you understand the case of the optimal separating hyperplane well, proceed with the soft margin hyperplane:

- What is the effect on the discrimination boundary and the SVs of varying the value of $C \in (0, \infty)$?
- If the data is linearly separating, does it give the same result as the optimal separating hyperplane case? Why?

If you feel adventurous, you can then implement:

- The nonlinear SVMs. What is the effect on the discrimination boundary and the SVs of varying the value of C ? How about the value of the kernel hyperparameter (q, σ) ? Determine the best hyperparameter values (C, q) or (C, σ) by using cross-validation.
- Multiclass SVMs (using either the one-vs-one or the one-vs-all approaches).
- Support vector regression.

III What you have to submit

We provide you with a script `lab09.m` which sets up the problem (toy dataset) and plots the figures mentioned earlier. You have to code the training for the linear SVM (both cases: the optimal separating hyperplane and the soft margin hyperplane) and explore its behavior.

Follow these instructions strictly. Email the TA the following packed into a **single** file (`lab09.tar.gz` or `lab09.zip`) and with email subject [CSE176] `lab09`:

- Matlab code for the function `linsvmtrain.m`. It trains a linear SVM (taking input vectors in D dimensions). Use the template provided. Read it carefully to understand what the function should do, and the function `linsvm.m` (which we provide). The function should work when called from the script `lab09.m` listed above. Note: you are not allowed to use any functions from the Matlab Toolboxes (in particular, the Statistics and Machine Learning Toolbox, or the Neural Network Toolbox). You can only use basic Matlab functions.
- A brief report (2 pages) in PDF format describing your experience with the algorithms. The more extensive and insightful your exploration, the higher the grade. Be concise. Don't include code or figures, we can recreate them by running your functions. Indicate the part that each member of the group did.