ECE M146 Introduction to Machine Learning

Prof. Lara Dolecek

ECE Department, UCLA

Today’s Lecture

Recap:

- Methods for classification; perceptron

New topic:

- Support Vector Machines (SVM)

Today's Lecture

Recap:

- Methods for classification; perceptron

New topic:

- Support Vector Machines (SVM)

Methods for linear classification we have already learnt about

- Perceptron and logistic regression: both operate on $w^T x$
- Recall perceptron: it stops as soon as it finds a separating hyperplane, but this might be very close to a point, likely causing very high misclassification rate on the unseen points

- Next: another method for binary classification

Today’s Lecture

Recap:

- Methods for classification; perceptron

New topic:

- Support Vector Machines (SVM)

Margin

- Margin is the smallest distance from any training point to the decision boundary.
- Picture:

- Goal in SVM is to **maximize the margin**.
- Note that when the margin is maximized, there will be at least one point on each side (otherwise, can shift the margin more)

Set-up

- First, geometry:

- Again, we are after the vector w perpendicular to the decision boundary.
- What is new is the new optimization rule to choose the best w.

Set-up

- Consider point x. Decompose it as follows:

- Note that the first term lies on the decision surface and the second term is in the direction of w.
- This decomposition is of course possible. What is interesting is what happens next:

Set-up

- Consider:

- Here r is the distance of point x to the decision boundary $\mathcal{D}$. It can be positive or negative.
- We also introduce the unsigned distance:

Optimization problem

- We optimize the vector w and the intercept term b (we didn't use dimensionality enlargement here) to maximize the unsigned distance:

- Smallest distance to the decision boundary over all N points:

- Thus, the optimization problem is:

Convenient rescaling

- It is convenient to rescale parameters w and b as follows:

- Note that the unsigned distance is unchanged:

Convenient rescaling

- Consider a point x_n^* closest to the boundary:

- Key insight: All point then satisfy:

Active and inactive points

- Points that satisfy the preceding inequality with equality are called **active points**, and all the rest are called **inactive points**.

- Note that for the optimized decision boundary, there will be at least one active point on each side (i.e., in each class)

Reformulation of the optimization problem

- We so far have:
- Which we can also write as:
- Equivalent to:

Reformulation of the optimization problem

- The last expression we had

- Is an instance of a quadratic program with linear constraints.
- Example of a constrained optimization problem.

Lagrangian

- Next, we are going to convert a constrained optimization problem into an unconstrained optimization problem (a very general approach).
- We need a **Lagrangian**

- Terms a_n are called **Lagrange multipliers.**

New formulation

- Repeat Lagrangian

- Interpretation:

Min-max

- Min-max of the Lagrangian:

- Why like this ?
- Inner maximization works against us.
- When the inequality condition at some x_n is violated, we can just set a_n to $+\infty$. Then the $\min(\infty)$ is still ∞.

Min-max

- Min-max of the Lagrangian:

- Why like this ?
- Inner maximization works against us.
- When the inequality condition is not violated, i.e., $1-t_n(w^Tx_n+b)$ is at most 0, the best a_n can do is to be 0.

Set derivatives to 0.

- Set the derivative with respect to b of $\mathcal{L}$ to 0:

- Set the derivative with respect to w of $\mathcal{L}$ to 0:

Math, continued

- Substitute this w back into $\mathcal{L}$.

- Next, simplify this expression.

More math

- Go term by term.
- 1st term:

More math

- Go term by term.
- 2nd term:

New formulation of the Lagrangian

- New expression:

- We want to maximize this with respect to a_n's.
- Note that the new form does not involve w!
- Ok, but it seems that the problem is more computationally difficult than before (N constraints for a_n's vs. d+1 constraints for b and for w of dimension d)

Advantage of the new formulation

- Consider the evaluation y(x) on a new point x:

- So we only need the non-zero a_n's!
- Recall active points ☺

Recall active points

- Lagrange coefficient a_n is non-zero only if $t_n(w^T x_n+b) - 1 = 0$ i.e., only if the point x_n is at the margin distance from the decision boundary.
- Picture:
- Consequence: Most points will indeed have a_n zero, except for a few points that are on the margin distance from the decision boundary.
  - Subtle point: Only margin distance points can have a_n non-zero, but not all of them need to as long as there is at least one on each side with a_n non-zero.

Support vectors

- Points x_n that have a_n non-zero are called support vectors. Hence the name of the method.

- Now, recall the evaluation y(x) at test time:

- Generally, very fast!

Transformation from primal to dual

- What we did in effect was the following:

- Note that the solution for the dual is a lower bound for the solution of the primal.
- Picture:

- When so-called KKT conditions are satisfied, the bound is tight.

KKT conditions

- At the saddle point solution, the following holds:
- Condition 1:
- Condition 2:
- Condition 3: (complementary slackness)

Back to our set-up

- Recall our vector w.

- Lagrange multipliers a_n's can be computed using standard methods such as sequential minimization optimization (SMO); forthcoming.

- Let's suppose that we have computed a_n's.

- What's left is to evaluate b.

Evaluation of the parameter b

- Directly:

- Numerically more stable:

Computing a_n's

- SMO is a standard technique.
- Consider again the expression we want to maximize w.r.t. a_n's:

- Initialize a_n's to some values so that constraints A and B hold.
- But setting all a_n's to zero does not work!

SMO on a small example

- Consider the following example with two data points, as follows:

SMO in general

- Pick two distinct points x_j and x_k with a_j and a_k
- Update from the current value by solving the quadratic.
- Re-optimize as needed.