CV_M10_L01
Support Vector Machines

CS/DSA 5970: Machine Learning Practice
Support Vector Machines

A different take on classification/regression
- Classification: Vapnick et al. (1963, 1992)
- Regression: Smola and Schölkopf (1998)

- Linear models
- But: can have non-linear transformations (and we can do these very efficiently)
- We don’t explicitly represent the model parameters. Instead, the function is captured using a subset of the training samples
Support Vector Machines

Classification problem:
• Given: positive and negative samples (training set)
• Want to find a hyperplane that divides the set of points as “best” as possible
• Here: best means that we want to place the hyperplane so that it separates the points while being as far away from the points as possible
Quadratic Programming

CS/DSA 5970: Machine Learning Practice
Quadratic Programming

Standard class of problems and corresponding algorithm:

- Minimize an objective function (quadratic, linear or mixture)
- Subject to a set of inequalities (\(\geq\))

- Algorithm engages in a search:
  - Which equalities to satisfy (\(=\)) and which to allow to be unequal (\(>\))
  - Values for each of the parameters
Quadratic Programming

- QP solvers are available in just about any serious mathematical tool kit, including numpy
- To use, one needs to transform your problem into a standard form
Soft-Boundary Classification

CS/DSA 5970: Machine Learning Practice
Soft-Boundary Classification

- Many classification problems do not have a perfect linear solution
- We would like to explicitly acknowledge this, but still have a sense of maximum margin
Soft-Boundary Classification

Approach:

• Allow the algorithm to choose which samples to not have on the correct side of the margin

• Now have two objectives:
  – Minimize squared weights (which maximizes the boundary)
  – Minimize the misclassification error
  – Hyper-parameter: what is the balance between the two?
Non-Linear Preprocessing for SVMs

CS/DSA 5970: Machine Learning Practice
Non-Linear Preprocessing for SVMs

- Support Vector Machines are inherently linear methods.
- As with our earlier linear regression methods, we can apply non-linear transformations to the input features.
- This gives us an effective way of expressing curved decision boundaries in the original feature space.
Non-Linear Preprocessing for SVMs

If we choose our non-linear transformations carefully:

• Can create very large dimensional feature sets, which gives us the ability to be very expressive about decision boundaries

• Can also be computed very efficiently!
• CV_M4_L04c
Kernel Functions

• We can use a $K()$ without having to explicitly articulate what the corresponding $\Phi()$
  – For the Gaussian kernel: $\Phi()$ is an infinite dimensional vector
• If we have two existing kernel functions ($K_1()$ and $K_2()$), then we can create new kernel functions:
  – $K_1() + K_2()$
  – $K_1() \times K_2()$
Kernel Function Implications

• Allow us to express a decision surface in a high-dimensional space without explicitly touching that space
  – Don’t have to represent the feature vectors
  – Don’t have to represent W

• For a single query, we have to touch all of the support vectors in the training set
  – The alphas are zero for the non-support vectors, so we can leave them out of the sum
  – The set of non-zero alphas can still be very large (an issue with large training sets)
Example: SVMs for Classification

CS/DSA 5970: Machine Learning Practice
Example: SVMs for Classification

Scikit-learn provides several implementations of SVMs
• Some variation in parameters and naming
• Our focus: SVC
  – Based on the libsvm implementation
• Live demo
IPAD_M10_L05b

• Gaussian kernel revisit
Support Vector Regression

CS/DSA 5970: Machine Learning Practice
Support Vector Regression

• Basis: linear model
• Cost function: trade-off between explaining the training data and making the coefficients small
• Can transform into a dual problem where:
  – Queries are addressed using a weighted sum over the training set
  – The same *kernel trick* can be used to transform high-dimensional problem into a low-dimensional one
• IPAD_M10_L06
Example: Support Vector Regression

CS/DSA 5970: Machine Learning Practice
Example: Support Vector Regression

Scikit-learn:

• LinearSVR
• SVR
• Live demo