SVM: Support Vector Machine (classification)

Linear

- Constructing a projection
  - Datapoint x
  - Projection vector (Normal to plane) w
  - projection of x onto w
    $w^\top \cdot x = \|w\|\|x\| \cos(\theta)$
  - => the sign allows to separate points on either side of the plane
    $w^\top \cdot x > 0 \rightarrow$ on the left handside of plane
    $w^\top \cdot x < 0 \rightarrow$ on the right handside of plane
- Linear classifiers
  - Class label $y = \{-1;\ 1\}$
  - Separating hyperplane is comprised of normal and intercept
    w : plane normal
    b : intercept
  - **Decision unction** $y = f(x; w, b) = \text{sgn}(w^\top x + b)$
- **Classifier margin**
  - difinition: the width of the boundary
  - best measure of SVM: the maximum margin
  - **Support vectors**
    - datapoints closest to the boundary
    - define the margin
- Computing the Distance to the Separating Hyperplane
  - The margins on either side of the hyperplane satisfy $< w, x > +b = \pm 1$
    - positive $\{x: \ < w, x > +b = +1\}$
    - negative $\{x: \ < w, x > +b = -1\}$
  - Distance to place = projection $(x' - x)$
    - distance projection with unitary vector $\frac{\langle w, (x' - x)\rangle}{\|w\|} \frac{w}{\|w\|}$
    - datapoint porjection on the plane $\langle w, x'\rangle + b = 0 \Rightarrow \langle w, x'\rangle = -b$
    - distance to place $\frac{|\langle w, x\rangle + b|}{\|w\|}$
- Computing the margin
  - Distance of each points on either side of the margin => margin between two classes
    $\left.\begin{aligned} \left\|\mathrm{x}^1 - \mathrm{x}'\right\| &= \frac{\left|\langle w, x^1\rangle + b\right|}{\|w\|} = \frac{1}{\|w\|} \\ \left\|\mathrm{x}^2 - \mathrm{x}'\right\| &= \frac{\left|\langle w, x^2\rangle + b\right|}{\|w\|} = \frac{1}{\|w\|} \end{aligned}\right\} \Rightarrow \left\|\mathrm{x}^1 - \mathrm{x}^2\right\| = \frac{2}{\|w\|}$
- **Objective function (Primal problem)**
  - minimize the convex form $\frac{\|w\|^2}{2}$ ← Maximize => Minimize
  - constraints $y^j\left(\langle w, x^i\rangle + b\right) \geq 1,\ \mathrm{i} = 1, 2, \ldots, \mathrm{M}$
  - MIsc.
    - (N+1) parameters $w \in \mathbb{R}^N$, $b \in \mathbb{R}$ — dimension of data
    - M contraints — number of data
- Solving the constrained optimization
  - Lagrange method
    - introduction M Lagrange multipliers alpha
    - MaxMin problem — maximizing over alpha and minimizing over w and b
    - Requesting the gradient
  - Solution
    - Normal vector $\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \Leftrightarrow w = \sum_{i=1}^{M} \alpha_i y^i x^i$
      - global solution but not identity
      - different linear combinations lead to different solutions
    - Intercept $\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^{M} \alpha_i y^i = 0$
- **Dual optimization problem**
  - Taking the definition of w and plugging it back to the Lagrangian
  - Solved through Sequential Minimal Optimization algorithm (SMO)
  - Objective function $\max_{\alpha} W(\alpha_i) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j x_i^T x_j$
  - Constraints subject to: $\alpha_i \geq 0$ and $\sum_{i=1}^{M} a_i y_i = 0$
- **The KKT Conditions**
  - Ensure that the primal and dual optimization problems have the same optimal solutions
    - Support vectors $\alpha_i > 0$
    - Ignored points $\alpha_i = 0$
  - Decision function in terms of the support vectors $f(x) = \text{sgn}\left(\sum_{i=1}^{M} \alpha_i y^i < x, x^i > +b\right)$
  - Use KKT to compute b $\left(y^i\left(\left\langle \sum_{i=1}^{M} \alpha_i y^i x^i, x^i \right\rangle + b\right) - 1\right) = 0$
- Non-separable datasets
  - Introduce slack variables $\xi_i \geq 0$
    - correct classification $\{0\}$
    - correct classification inside margin $(0, 1)$
    - missclassification $[1, \infty)$
  - Relax the constraints $y_i(< w, x > +b) \geq 1 - \xi_i$
  - Modify the optimization function $\min_{w, \xi}\left(\frac{1}{2}\|w\|^2 + \frac{C}{M} \sum_{j=1}^{M} \xi_j\right)$ — Tradeoff: The higher C is, the better fitting the model is => lead to over-fitting possibly — $\frac{C}{M}$

Introduce slack variables

Integrate the slack variables into objective

Non-linear

- Using polar coordinates => Data become linearly separable — e.g. two groups of points are randomly distributed with different radii
- **Idea: Making the problem linear**
  - data X
  - feature space H
  - nonlinear map ϕ
  - Map original input space (data X) into high-dimension feature space through the nonlinear map
- Finding the transformation
  - The dimension of mapped feature space may be greater than original one
  - Determining \phi is difficult and even impossible
  - **Sufficient to compute distance in feature space**
- Kernel method
  - Kernel function $k: \ X \times X \rightarrow \mathbb{R}$, $k(x^i, x^j) \rightarrow \langle \phi(x^i), \phi(x^j)\rangle$
  - Different kernels
    - RBF (Gaussian) kernel $k(x, x^i) = \exp(-\frac{\|x - x^i\|}{2\sigma^2})$
    - Polynomial kernel $k(x, x^i) = \left(\langle x, x^i\rangle + c\right)^p$
    - Hyperparameters that need to be determined by hand

Misc.

- Key components
  - Hyperplane
  - Support vectors
  - Margin
    - Region of equal distance on both side
    - Distance to hyperplane — Color gradient
- Effect of hyperparameters on performance
  - Higher C => Higher accuracy (possible over-fitting)
  - C that determines the costs associated to incorrectly classifying datapoints
- Multi-class classification
  - Construct a set of K binary classifiers
    - Sufficient to use K-1 classifier for K classes
    - Computing the K-th classifier may provide tighter bounds
- Summary
  - **No confidence in prediction**
    - Does not entail a notion of confidence! (no a notion of likelihood)
    - Predict by default sign of b lead to large amount of false positives
      - Doing crossvalidation would not prevent the effect as it is close to training datapoints.
    - Improvements
      - Verify the sign of b for specfic class
      - Run crossvalidation by generating the datapoints that never seen
      - Compare the distribution of train/ test
  - **Pros**
    - Compared to KNN — Build a model & boundary is a specific function
    - Compared to GMM — Guarantee to find global optimum
  - **Cons**
    - Computing costs at testing O(M), M is number of datapoints
    - Require to choose two more parameters (Compared to 1 for KNN and GMM)
      - Weight of slack variable C
      - Kernel width σ