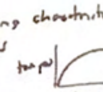
Matthew Tran
CS189
MT1
Lec 1-13

- Validation
  - training set error
  - test set error
  - outliers, overfitting
    - want generalize
  - hyperparameter - control fit
    - tune w/ validation set
- Supervised learning - classification, regression
- unsupervised learning - clustering, dimensionality reduction
- Classification - sample of observations w/ features / predictors
  - nearest neighbors
  - k-nearest neighbors
  - decision boundary
  - decision/predictor/discriminant function
    - $f(x) > 0$ if $\in C$, $f(x) < 0$ is not in C
  - linear classifier - line/plane/hyperplane
  - inner product == dot product ; linear!
  - Euclidean norm $\|x\| = \sqrt{x \cdot x}$ l_2
    - $\cos\theta = x \cdot y / \|x\|\|y\|$
  - $H = \{x : w \cdot x = -\alpha\}$ hyperplane $f(x) = w \cdot x + \alpha$
    - w is $\perp$ to all H vectors
    - $w \cdot x + \alpha$ == signed distance if $\|w\| = 1$
    - linearly separable == plane exists
- Centroid Method
  - μ_C of all points in C, μ_X not in C
  - $f(x) = \underbrace{(\mu_C - \mu_X)}_{\text{normal}} \cdot x - \underbrace{(\mu_C - \mu_X) \cdot \frac{\mu_C + \mu_X}{2}}_{\text{midpoint}}$
  - good for Gaussian
- Perceptron Algorithm
  - slow, but correct for lin sep
  - use gradient descent
  - 1 if $\in C$, -1 if not
  - only origin H
  - find w st $X_i \cdot w \geq 0$ if in C
    - aka $y_i X_i \cdot w \geq 0$ constraint
  - loss fnc $L(z, y_i) = \begin{cases} 0 & \text{if } y_i z \geq 0 \\ -y_i z & \text{else} \end{cases}$
  - risk fnc $R(w) = \frac{1}{n} \sum L(X_i \cdot w, y_i)$
  - optimization prob - find w min $R(w)$
  - x-space $w \cdot z = 0$, x
  - w-space w, $x \cdot z = 0$
  - $\nabla_w R(w) = -\sum_{i \in V} y_i X_i$
    - $w \leftarrow w - \epsilon \nabla R(w)$ gradient descent $O(nd)$
    - stochastic - pick one
  - add fictitious dimension to "lift" !!!
  - "online" alg - can train while running
  - Convergence Theorem - $O(r^2/\gamma^2)$ iter, r = max $\|X_i\|$, γ = margin
- Maximum Margin Classifier (SVM) (hard margin)
  - margin - distance H to pt closest
  - $y_i(w \cdot X_i + \alpha) \geq 1 \Rightarrow$ margin dist $= \frac{w}{\|w\|} \cdot X_i + \frac{\alpha}{\|w\|} \geq \frac{1}{\|w\|}$
  - Find w, α, minimize $\|w\|^2$
    - quadratic prog d+1 dim, n constraints
- Soft Margin SVM
  - hard SVM - fails if not lin sep, bad w/ outliers
  - add slack variables $\xi_i \geq 0$
    - $y_i(X_i \cdot w + \alpha) \geq 1 - \xi_i$
    - margin = $1/\|w\|$
  - add loss term
    min $\|w\|^2 + C \sum_{i=1}^{n} \xi_i$
  - small C - max $1/\|w\|$, underfit, less outlier, more flat
  - big C - imp slack 0, overfit, bad outlier, sinuous
  - nonlinear?
    - make nonlinear features!! hyperparam?
    - ex. add $\|x\|^2$
    - quadric - higher dim conic section
    - slow? kernelization
    - else select features instead?
- whiten data - mean = 0 subtract mean $N(0, I)$
  var = 1 divide by std

- Optimization Alg
  - unconstrained - obj fnc $f(w)$
    - smooth if $\nabla f(w)$ cont fnc
    - global/local min/max
    - Convex  never above line
    - smooth fcts
      - gradient descent (nonsmooth too)
      - newton's method
      - nonlinear conjugate gradient
    - nonsmooth - BFGS
    - line search - optimize in 1D
  - constrained & smooth
    - subject to $g(w) = 0$
  - LP - w st min $c \cdot w$
    subj to $Aw \leq b$ ($n \times d$)
    - feasible region - nonempty if linear
    - optimum - active constraints "support vector" in SVM
    - alg - simplex, interior point
  - QP $f(w) = w^T Q w + c^T w$
    - $Aw \leq b$
    - Q is PSD (to be convex)
    - alg simplex, SMO, coordinate descent
- Decision Theory == Risk min
  - X sample, Y class
  - $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$ ($P(Y)$ prior; $P(Y|X)$ posterior prob) careful w/ multivar $P(a|b,c) = \frac{P(a,b,c)}{P(b,c)}$
  - loss function can penalize false neg, pos diff (asym)
    - 0-1 loss fnc is symmetric
  - let r be decision rule, def risk all possibilities $P(Y|X)$
    $R(r) = E[L(r(X), Y)] = \sum_x (\ ) P(X=x)$
  - Bayes decision rule/classifier min $R(r)$ w/ Bayes risk
    - basically, pick class largest posterior prob
  - PDF $f(x)$ $\int_{-\infty}^{\infty} f(x)dx = 1$
    $E(X) = \int_{-\infty}^{\infty} x f(x) dx$
    $\sigma^2 = E((X-\mu)^2) = E(X^2) - \mu^2$
- Gaussian Discriminant Analysis
  - all class from Gaussian $X \sim N(\mu, \sigma^2)$: $f(x) = \frac{1}{(\sqrt{2\pi}\sigma)^d} e^{-\frac{\|x-\mu\|^2}{2\sigma^2}}$
    $X \sim N(\mu, \Sigma)$: $f(x) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$ (Σ covariance; Σ^{-1} precision matrix)
  - Bayes's max $Q_C(x) = \ln((\sqrt{2\pi})^d f_C(x) \pi_C)$ (elim, PDF, prior)
  - MLE
    - pick model, param that best fit p max $L(p)$ likelihood fn
    - $\frac{dL}{dp} = 0$, check $\frac{d^2L}{dp^2} < 0$ (max)
    - Gaussian $L(\ldots) = f(X_1)f(X_2)\ldots f(X_n)$ $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$
    - can also max log likelihood $\hat{\sigma}^2 = \frac{1}{dn}\sum_{i=1}^{n} \|X_i - \hat{\mu}\|^2$
  - Eigenvector,
    - $Av = \lambda v$, just scalar, $A^k v = \lambda^k v$
    - Spectral decomp real symm $A = V \Lambda V^T$ "square root" exist $\Sigma^{1/2}$
      - general $A = Q \Lambda Q^{-1}$
    - symm PD $x^T A x > 0$
      PSD $x^T A x \geq 0$
  - QDA
    $\hat{\Sigma}_C = \frac{1}{n_C} \sum (X_i - \hat{\mu}_C)(X_i - \hat{\mu}_C)^T$ outer product, conditional cov
    - max quadratic discriminant fnc
    $Q_C(x) = -\frac{1}{2}(x-\mu_C)^T \Sigma_C^{-1}(x-\mu_C) - \frac{1}{2}\ln|\Sigma_C| + \ln \pi_C$
  - LDA
    one $\hat{\Sigma} = \frac{1}{n} \sum\sum (X_i - \hat{\mu}_C)(X_i - \hat{\mu}_C)^T$ for all
    $Q_C(x) = \underbrace{\mu_C^T \Sigma^{-1} x}_{w \cdot x} - \underbrace{\frac{1}{2}\mu_C^T \Sigma^{-1} \mu_C + \ln \pi_C}_{\alpha}$
  - not perfect bc still estimating, data not perfect Gaussian
  - X sample mat $\begin{bmatrix} - X_1 - \\ \vdots \end{bmatrix}$ == design matrix
  sphering $W = \dot{X}\, \mathrm{Var}(R)^{-1/2}$ ($\dot{X}$: $-\mu_C$)

- Regression
  - regression fn h
    (1) $w \cdot x + \alpha$ - linear
    (2) polynomial
    (3) logistic $s(w \cdot x + \alpha)$ $s(\gamma) = \frac{1}{1+e^{-\gamma}}$
    $s'(\gamma) = s(\gamma)(1 - s(\gamma))$
  - loss fn L
    (A) $(z-y)^2$ - squared
    (B) $|z-y|$ - abs
    (C) $-y \ln z - (1-y)\ln(1-z)$ logistic loss/cross entropy
  - cost fn J
    (a) $\frac{1}{n}\sum_{i=1}^{n} L(h(X_i), y_i)$ - mean
    (b) $\max_{i=1}^{n} L(h(X_i), y_i)$ - max
    (c) $\sum_{i=1}^{n} \omega_i L(h(X_i), y_i)$ - weighted sum
    (d) " $+ \lambda \|w\|^2$ l_2 penalized/regularized
    (e) " $+ \lambda \|w\|_1$ l_1 "
  - least squares linear reg 1 + A + a
    - min $\|Xw - y\|^2 = RSS(w)$
      $w = (X^T X)^{-1} X^T y$
    - fail if underconstrained, sensitive outliers
  - logistic regression 3 + C + a
    - $\nabla_w J = -X^T(y - s(Xw))$ $\nabla^2_w J = X^T \Omega X$ ($s(1-s)$)
    - Newton's Method - GD but smarter step
      - $\nabla^2_w J(w)$ == Hessian
        $w \leftarrow w - (\nabla^2 J(w))^{-1} \nabla J(w)$ (f'/f'')
      - needs good start
  - LDA where well sep; log. reg - dec bound robust, natural fit labels
  - ROC curve (Receiver operating characteristics)
    - rate false pos vs true pos  true pos / false pos ↔ specificity, true neg / I false neg
  - empirical distribution - discrete RV w/ sample
  - empirical risk - $\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(X_i), y_i)$
  - Bias Variance Decomposition
    - bias - error from bad model
    - variance - error from noise
    $E(h(z)) - g(z)$ bias, $\mathrm{Var}(h(z))$ variance, $\mathrm{Var}(\epsilon)$ irreducible error
  - Ridge Regression (Tikhonov)
    - penalty less of optimal, encourage small $\|w\|$
      - unique soln PD
    - 1 + A + l_2 d
    - $X^T X + \lambda I$ is PD!
    - balances w/ λ
    - any desired soln, just whiten data
  - MAP (maximum a posteriori)
  - LASSO
    - 1 + A + e l_1 - more sparse

$\nabla_x (y \cdot z) = (\nabla_x y) z + (\nabla_x z) y$
$\nabla_x \alpha y = (\nabla_x \alpha) y^T + \alpha \nabla_x y$
$\nabla_x f(y) = (\nabla_x y)(\nabla_y f(y))$
$\nabla_x C y(x) = \nabla_x y(x) C^T$
$\nabla_x Ax = A^T$ $\nabla_x x = I$

Matthew Tran
CS189
Final

- Feature Subset Selection
  - all increase variance, not all ↓ bias
  - subset - less overfitting, inference (wisdom)
  - Best - try all $2^d - 1$ subsets, choose best, v. slow
  - Heuristic 1 - forward stepwise selection
    - null model 0 features init
    - add best feature until validation ↑ worse / decreases
    - $O(d^2)$ ↑ at each iter
  - Heuristic 2 - backward stepwise selection
    - remove those that improve validation
    - init all d, $O(d^2)$, but faster than
  - Add. Heuristic - remove small weight features
    - var of betas $\propto \sigma^2 (X^T X)^{-1}$
    - z scores $z_i = \frac{w_i}{\sigma \sqrt{v_i}}$] diag. pred
      ↑ remove if small
- Decision Trees
  - nonlinear classification / regression
  - nodes
    - internal - branch on usually 1 feature
    - leaf - class
  - features
    - rectangular cells
    - good for categorical, quantitative
    - interpretable result (inference)
    - arbitrary boundary
  - Training
    - classification - greedy heuristic
      - recursive
      - pick best feature + threshold
      - stop when all correct (same class)
      - min $J(S)$ or $J(S_l) + J(S_r)$
        - $J(S)$ measures entropy
        - $P(Y=C) = p_C$
        - surprise = $-\log_2 p_C$
        - entropy $H(S) = -\sum_C p_C \log_2 p_C$
          ↑ bits of info to encode class
        - weighted avg: $H_{after} = \frac{|S_l| H(S_l) + |S_r| H(S_r)}{|S_l| + |S_r|}$
        - maximize - info gain = $H(S) - H_{after}$ ← or min
      - $p(1-p)$ also works, concave good
  - Runtime $O(\log n)$ usually
    - classify - $O(\text{tree depth})$, for binary, ≤ d
    - train - ≤ $O(nd \text{ depth})$
  - variants ↘
  - Multivariate Tree
    - non-axis aligned splits ↙
      - use like SVM, logistic regression, GDA
  - Regression
    - piecewise constant
      - $J(S) = \frac{1}{|S|} \sum_{i \in S} (y_i - \mu_S)^2$
  - Stopping Early
    - limit depth = speed
    - limit size
    - reduce overfit
    - noise + overlapping distrib ⇒ better to posterior prob
  - Stopping Conditions
    - not enough entropy improvement
    - most points same class
    - few sample points
    - fancy cell edge
    - high depth, risky ↑ too many points
    - validation - best to make ↑, then prune
    - if multiple points
      - majority, probs, or average
  - Pruning
    - greedily remove splits that improve validation
    - don't redo everything, just see which pts go where
    - better bc bad split usually followed by good split
- Ensemble Learning
  - decision trees - high variance, can have low bias
  - average them!
    - regression - median or mean
    - classification - majority or average probs
    - low bias, high variance
      - underfitting == ↑ bias
      - overfitting == ↑ variance
  - Bagging
    - random subsample w/ replacement
      - some points have higher weight
    - until T learners
    - metalearner - ensemble learners
  - Random Forests
    - w/ bagging might split same
    - only split on m features!
      $\sqrt{d}$ for class, d/3 for regression
    - less correlation! meta learner

- Kernel trick
  - use p-dim w/o $O(d^p)$ features?
    - $\Phi(X)$ - X lifted to p dim
    - usually only dot product of $\Phi(X)$
  - Kernel Ridge Regression
    - $w = X^T a$ ← optimize a (n) instead of w (d^p)
    - center X and y and add $X_{i,d+1} = 1$
      $(X^T X + \lambda I) w = X^T y$
      $(X X^T + \lambda I) a = y$
    - a is dual
    - $h(z) = w^T z = a^T X z = \sum_{i=1}^{n} a_i X_i^T z$
    - $k(x, z) = x^T z$ is kernel fn
    - $K = X X^T$ is kernel matrix
      solve $(K + \lambda I) a = y$ for a
      $h(z) = \sum_{i=1}^{n} a_i k(X_i, z)$
  - Kernelization
    - polynomial kernel $k(x, z) = (x^T z + 1)^p$
      - then $k(x, z) = \Phi(x)^T \Phi(z)$ all monomials
  - Kernel Perceptron
    - orig alg
      - $w \leftarrow y_1 \Phi(X_1)$ // random starting
      - while some $y_i \Phi(X_i) \cdot w < 0$: // misclassified?
        $w \leftarrow w + \epsilon y_i \Phi(X_i)$
      - $h(z) \leftarrow w \cdot \Phi(z)$
    - dual $w = \Phi(X)^T a$
    - kernelized
      - $a \leftarrow$ random
      - $K_{ij} = k(X_i, X_j)$
      - while some $y_i (Ka)_i < 0$
        $a_i \leftarrow a_i + \epsilon y_i$
      - $h(z) \leftarrow \sum_{j=1}^{n} a_j k(X_j, z)$
  - Kernel Logistic Regression
    - start at a
    - $a_i \leftarrow a_i + \epsilon (y_i - s((Ka)_i))$ stochastic (s = logistic)
    - $a \leftarrow a + \epsilon (y - s(Ka))$ batch
      $h(z) = s\left(\sum_{j=1}^{n} a_j k(X_j, z)\right)$
  - Gaussian Kernel
    - infinite dimensional!
    - $k(x, z) = e^{-\frac{\|x - z\|^2}{2\sigma^2}}$
    - measure of "similarity"
    - σ is hyperparameter ← width of Gaussian
      ↑ σ → ↑ bias ↓ variance
- Neural Networks
  - need nonlinear fns → can do nonlin sep
  - training
    - SGD or BGD
    - pick loss fn $L(z, y)$ ← usually vector
    - pick cost $J(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(X_i), Y_i)$ ← for multi out
    - usually not convex ∴ local min
    - random weights - so all neurons diff
      - also not too big bc sigmoid ↑ near 0 ∇
    - backpropagation
      - gradient w.r.t. loss and update
  - variants
    - regression - linear output
    - classification - k ≥ 3, softmax ✓ $s(t)_i = \frac{e^{t_i}}{\sum_j e^{t_j}}$ diff Δ
    - sigmoid unit saturation - "stuck"
      - init edge to random w/ mean 0, $\sigma = 1/\sqrt{\text{fan in}}$
      - target to 0.15 or 0.85, not 0 (for class)
        - doesn't help hidden stuck
      - add $\epsilon \approx 0.1$ to s' - not real backprop
      - cross entropy loss instead of sigmoid
        $L(z, y) = -\sum_{i=1}^{k} y_i \ln(z_i)$
    - ℓ_2 regularization
    - ReLU, not sigmoid - arbitrary small but can be stuck, leaky
  - heuristics (more)
    - SGD
    - normalize data, center, scale
    - use tanh instead of sigmoid for hidden
    - diff learn rate per layer (larger for earlier)
    - emphasizing - weight rare or misclass
    - 2nd order - Newton's, more good in fx
    - RMSprop, Adam, AMSGrad
    - momentum - $w \leftarrow w + \Delta w$
      $\Delta w \leftarrow -\epsilon \nabla J(w) + \beta \Delta w$ (β < 1)
    - max w/ clock
    - ↓ overfit
      - ensemble
      - ℓ_2 reg
      - dropout - disable random neurons each iter
      - ↓ hidden

- CNN
  - deep learning - many layers
  - idea - overparameterized, via filters, mask, kernel
- Unsupervised Learning
  - no labels! find structure in data
  - PCA
    - find k directions that capture most variation
    - help in dim reduce
    - assume X centered, 0 mean
      $\tilde{X} = \frac{x \cdot w}{\|w\|^2} w$ ← proj$_w x$, x onto w
    - $X^T X$ square, sym, PSD, d×d
      - $0 \le \lambda_1 \ldots \lambda_d$ sorted
        $v_1 \ldots v_d$ unit eigen
    - MLE deriv!
    - deriv 2 max sample variance
    - deriv 3 - min projection
    - ex. eigenfaces, nearest neighbor
  - SVD
    - $X^T X$ slow $O(nd^2)$, inaccurate eigen
    - SVD $X = UDV^T$ ← $\begin{bmatrix} - v_1 - \\ \vdots \\ - v_d - \end{bmatrix}$ = $\sum_{i=1}^{d} \delta_i u_i v_i^T$
      n×d, U n×d, D $\begin{bmatrix} \delta_1 & & 0 \\ & \ddots & \\ 0 & & \delta_d \end{bmatrix}$ d×d
    - v_i is eigenvector of $X^T X$ w/ eigenval δ_i^2
      $X^T X = V D^2 V^T$
    - $O(ndk)$ or even $O(nd \log k)$
  - Clustering
    - k means aka Lloyd
      $\mu_i = \frac{1}{n_i} \sum_{y_j = i} X_j$
      $\min \sum_{i=1}^{k} \sum_{y_j = i} \|X_j - \mu_i\|^2$ ✓ equiv to min within dist $\sum_{i=1}^{k} \frac{1}{n_i} \sum \sum \|X_j - X_m\|^2$
    - $O(nk^n)$ - all partitions NP hard
    - k-means heuristic
      - update μ_i] alternate
      - update y_i
      - fast in practice
    - Forgy
    - Random
    - k means ++
    - run multiple times is better
    - k-medoids clustering
      - arbitrary distance fns, solution is sample
      - ex. ℓ_1, ℓ_∞, cosine
    - Hierarchical clustering
      - bottom up / agglomerative - fuse pairs
      - top down / divisive - all in 1, split
      - need dist for clusters
        - complete $O(n^3)$ naive
        - single $O(n^2)$ better
        - average
        - centroid
      - dendrogram - visual tree look
        - cut at # of clusters
    - Spectral Graph Clustering
      - weighted, undirected, no self edge
      - edge weights ∝ similarity
      - cut graph b/c min edge weight cut
      - good for image segmentation
      - look at info
    - High dim clustering
      - most in thin shell far origin
      - k-mean, NN bad for high d
      - usually 90°!
      - random projection
        - hard, but sim dist ccu
  - AdaBoost
    - multiple learners w/ weighted pts
    - diff weight each learner
    - increase weight of misclassified
    - more accurate → bigger vote
    - Theory ← exponential e-pl
      $risk = \frac{1}{n} \sum_{i=1}^{n} L(M(X_i), Y_i)$ $M(x) = \sum_{t}^{T} \beta_t G_t(X_i)$
    - no rate
    - overfitting
    - like subset selection
    - nonlin. no better
  - Nearest Neighbor Classification
    - $n \to \infty$ | NN error < $2B - B^2$
      2 NN $2B - 2B^2$
      $n, k \to \infty$, $k/n \to 0$: k NN → B optimal
    - alg exhaustive - naive
      voronoi -
      kd tree -