L 8

- Bias and variance
- Regularization
- Train/dev/test splits
- Model selection / cross-validation

Bias and Variance

Bias: original assumption is bad
High variance: a slightly different train set would create completely different model

Too many features → overfit
Too few features → not enough

Regularization prevent overfitting (with support vector machine)

$\min_\theta \frac{1}{2}\sum_{i=1}^{m}(y^{(i)}-\theta^T x^{(i)})^2 + \frac{\lambda}{2}\|\theta\|^2$ → for linear regression

(regularization term)

choose λ to prevent overfitting
↳ prevent high variability in general

$\arg\max_\theta \sum_{i=1}^{n} \log p(y^{(i)}|x^{(i)},\theta) - \lambda\|\theta\|^2$ → for classification

Why doesn't SVM overfit?
somehow $\min \|w\|^2 \leftrightarrow \lambda\|\theta\|^2$
have effect of regularization

Logistic regression with regularization better than naive bayes in text classification

Note: preprocess data to make x_i all on same scale
So $\lambda\|\theta\|$ is justified
by $\frac{x_i - \mu_i}{\sigma_i}$

S is training set $= \{x^{(i)}, y^{(i)}\}$

$p(\theta|S) = \frac{p(S|\theta)p(\theta)}{p(S)}$

$\arg\max_\theta p(\theta|S) = \arg\max_\theta p(S|\theta)p(\theta)$

$= \arg\max_\theta (\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta))\, p(\theta)$ ← (logistic regression model)

Assume: $p(\theta) = \theta \sim N(0, \tau^2 I)$

$p(\theta) = \frac{1}{\sqrt{2\pi}^n |\tau^2 I|} e^{-\frac{1}{2}\theta^T(\tau^2 I)^{-1}\theta}$

Frequentist $p(S|\theta)$ — MLE origin of regularization

Bayesian prior - $p(\theta)$ assumption

prior - distribution $\max_\theta p(\theta|S)$ — map → adding regularization

Train/dev/test set

eg 10,000 examples
order of polynomial?
or choose λ
or choose τ
or C

simple cross-validation set

1 Split S → S_{train}, S_{dev}, S_{test}
2 Train each model (optimize for degree of polynomial) on S_{train}
3 Measure error on S_{dev}, Pick best performance
4 Optional evaluate algorithm on S_{test}
↳ mainly for publication

Historical: S if S small
70% train, 30% test
60% train, 20% dev, 20% test

Modern perspective: $|S| = 10^6$ very large
train 98%
dev 2%
test 2%
→ way too much unless to measure performance increase very small
eg 90% vs 90.1%, then need more train data
but normally 99% train

Improve on the dev step, by optimize on dev set, can use feedback development
Do not incorporate test set performance into model

Suppose small set
$m = 100$
70 S_{train}, 30 S_{dev}
Seems S_{train} too small

k-fold cross-validation (only S small)
Split into k sets, eg $|100| = |5|$, $k = 5$, each set 20
For degree $d = 1 \ldots 5$ (degree of polynomial)
For $i = 1 \ldots k$
Train (at parameter) on $k-1$ pieces
Test on remaining 1 piece
Average of the error
Optional after choosing degree (retrain 100% of data)
Better than simple CV since has more data

What if even smaller?
$m = 20$, then choose $k = m$

mlyearning.org

Feature selection (most important)
find smaller subset of feature to avoid overfitting

Start with $f = \emptyset$
(1) Repeat add each feature i to f, and See which single feature add most important on dev set performance
(2) Add the feature to f

eg x_1 x_5
Forward selection ($\emptyset$) $h(x) = \theta_0$
1 $\begin{bmatrix} \theta+x_1 \\ \theta+x_2 \\ \theta+x_3 \\ \theta+x_4 \end{bmatrix}$ $f = \{x_2\}$
2 $\begin{bmatrix} x_1 & x_1 \\ x_2 & x_2 \\ x_3 & x_4 \\ x_4 & x_5 \end{bmatrix}$ → $f = \{x_2, x_4\}$
↓ repeat to complete good features

Also how backward selection which removes most irrelevant features