# 10-701 Cheat Sheet

## Non-Parametric

MaxLikelihood learning window will give you delta functions, which is a kind of over fitting. Use Leave-one-out cross validation for model selection. Idea: Use some of the data to estimate density; Use other part to evaluate how well it works. Pick the parameter that works best.
$\log p(x_i|X\setminus\{x_i\}) = \log \frac{1}{n-1}\sum_{j\neq i} k(x_i,x_j)$, the sum over all points is $\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{n}{n-1}p(x_i) - \frac{1}{n-1}k(x_i,x_i)\right]$ where $p(x) = \frac{1}{n}\sum_{i=1}^{n} k(x_i,x)$.
**why must we not check too many parameters?** that you can overfit more; for a given dataset, a few particular parameter values might happen to do well in k-fold CV by sheer chance, where if you had a new dataset they might not do so well. Checking a reasonable number of parameter values makes you less likely to hit those "lucky" spots helps mitigate this risk.

**Silverman's Rule for kernel size** Use average distance from k nearest neighbors $r_i = \frac{r}{k}\sum_{x\in \mathrm{NN}(x_i,k)} \|x_i - x\|$.

**Watson Nadaraya** 1. estimate $p(x|y=1)$ and $p(x|y=-1)$; 2. compute by Bayes rule
$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{\frac{1}{m_y}\sum_{y_i=y} k(x_i,x)\cdot \frac{m_y}{m}}{\frac{1}{m}\sum_i k(x_i,x)}$. 3. Decision boundary
$p(y=1|x) - p(y=-1|x) = \frac{\sum_j y_j k(x_j,x)}{\sum_i k(x_i,x)} = \sum_j y_j \frac{k(x_j,x)}{\sum_i k(x_i,x)}$
Actually, we assume that p(x—y) is equal to $1/m_y * \sum_y k(x_i,x)$. Using this definition, we can see $p(x,-1)+p(x,1) = p(x|-1)p(-1) + p(x|1)p(1) = p(x)$.
This can be incorporated into the regression framework in chap 6 of PRML. Where we define $f(x-x_n, t\neq t_n) = 0$, and $f(x-x_n, t=t_n) = f(x-x_n)$. Using this definition, we can derive all the probabilities on this slide. (see my handwritten notes on chap 6 of PRML).
Regression case is the same equation.
**kNN** Let optimal error rate be $p$. Given unlimited **iid** data, 1NN's error rate is $\leq 2p(1-p)$.

## Matrix Cookbook

$\frac{\partial \boldsymbol{x}^T\boldsymbol{a}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{a}^T\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{a}$
$\frac{\partial \boldsymbol{a}^T\boldsymbol{X}\boldsymbol{b}}{\partial \boldsymbol{X}} = \boldsymbol{a}\boldsymbol{b}^T$, $\frac{\partial \boldsymbol{a}^T\boldsymbol{X}^T\boldsymbol{b}}{\partial \boldsymbol{X}} = \boldsymbol{b}\boldsymbol{a}^T$, $\frac{\partial \boldsymbol{a}^T(\boldsymbol{X}^T|\boldsymbol{X})\boldsymbol{a}}{\partial \boldsymbol{X}} = \boldsymbol{a}\boldsymbol{a}^T$
$W\in \boldsymbol{S}$, $\frac{\partial}{\partial \boldsymbol{s}}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s}) = -2\boldsymbol{A}^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})$,
$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}-\boldsymbol{s})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s}) = 2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s})$,
$\frac{\partial}{\partial \boldsymbol{s}}(\boldsymbol{x}-\boldsymbol{s})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s}) = -2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s})$,
$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s}) = 2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})$,
$\frac{\partial}{\partial \boldsymbol{A}}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s}) = -2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})\boldsymbol{s}^T$,

## Naive Bayes

Conditionally independent: $P(x_1,x_2,\ldots|C) = \prod_i P(x_i|C)$.
One way to avoid divide by zero: add $(1,1,\ldots,1)$ and $(0,0,\ldots,0)$ to both classes.
**Learns** $P(x_i|y)$ for *Discrete* $x_i$ – $P(x_i|y) = \frac{\#D(X_i=x_i,Y=y)}{\#D(Y=y)}$
For smoothing, use $P(x_i|y) = \frac{\#D(X_i=x_i,Y=y)+k}{\#D(Y=y)+n_i k}$, where $n_i$ is the number of different possible values for $X_i$ (In practice problem set, Jing Xiang used $k=1$?) *Continuous* $x_i$ – Can use any PDF, but usually use Gaussian
$P(x_i|y) = \mathcal{N}(\mu_{X_i|y}, \sigma^2_{X_i|y})$, where $\mu_{X_i|y}$ and $\sigma_{X_i|y}$ are, respectively, the average and variance of $X_i$ for all data points where $Y=y$. The Gaussian distribution already provides smoothing.