CSCI567 Machine Learning (Fall 2017)

Prof. Fei Sha

U of Southern California

Lecture on Sept. 28, 2017

Outline

Outline

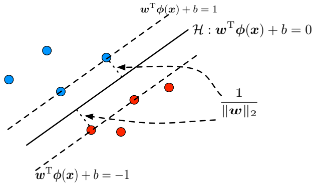
Administrative stuff

- Homework 2 Released
- Homework 1 Past Due: If you have not applied for Extension, your turn-in is considered late.

Outline

Support Vector Machines

Interpretation: maximize the margin



- For separable data

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad \forall \ n$$

- For non-separable data

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad \forall \ n$$
$$\xi_n \geq 0, \quad \forall \ n$$

where C is our tradeoff (hyper)parameter.

Support Vector Machines

Interpretation: minimize loss

- Minimize loss on all data

$$\min_{\boldsymbol{w},b} \sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

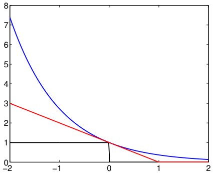
- equivalently

$$\min_{\boldsymbol{w},b,\{\xi_n\}} \quad C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$\text{s.t.} \quad 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \le \xi_n, \quad \forall\ n$$

$$\xi_n \ge 0, \quad \forall\ n$$

where all ξ_n are called *slack variables*.



$$\ell^{\mathrm{HINGE}}(f(\boldsymbol{x}), y) = \max(0, 1 - yf(\boldsymbol{x}))$$

Primal and dual

Primal

$$
\begin{aligned}
\min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2} \|\boldsymbol{w}\|_2^2 \\
\text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \ \forall\ n \\
& \xi_n \geq 0, \quad \forall\ n
\end{aligned}
$$

Dual

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n) \\
\text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad \forall\ n \\
& \sum_n \alpha_n y_n = 0
\end{aligned}
$$

Why we seek dual formulation

- We can kernelize the method by using kernel function in place of inner products
- We can discover interesting structures in solution: *support vectors*

Geometric interpretation of support vectors

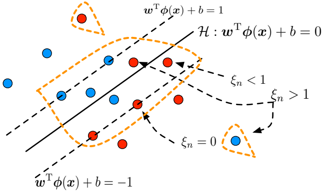
Some α_n will become zero

$$\min_{\boldsymbol{\alpha}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad 0 \le \alpha_n \le C, \quad \forall\ n$$

$$\sum_n \alpha_n y_n = 0$$

Nonzero α_n is called support vector

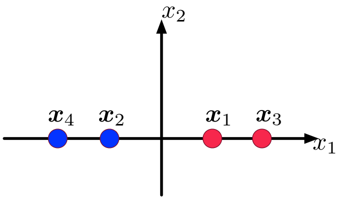


Support vectors are those being circled with the orange line. Removing them will change the solution.

Outline

The following toy problem

idx	x_1	x_2	y
$\boldsymbol{x}_1$	1	0	1
$\boldsymbol{x}_2$	-1	0	-1
$\boldsymbol{x}_3$	2	0	1
$\boldsymbol{x}_4$	-2	0	-1



Let us use linear kernel to solve the problem

$$k(\boldsymbol{x}_m, \boldsymbol{x}_n) = \boldsymbol{x}_m^{\mathrm{T}} \boldsymbol{x}_n$$

in other words, $\phi(\boldsymbol{x}) = \boldsymbol{x}$.

Guess the solution

- Decision boundary by SVM

$$x_1 = 0$$

ie, the vertical axis

- Support vectors: $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$

What is the dual formulation?

idx	x_1	x_2	y
$\boldsymbol{x}_1$	1	0	1
$\boldsymbol{x}_2$	-1	0	-1
$\boldsymbol{x}_3$	2	0	1
$\boldsymbol{x}_4$	-2	0	-1

Kernel matrix $\boldsymbol{x}_m^{\mathbf{T}}\boldsymbol{x}_n$

$$\boldsymbol{K} = \begin{pmatrix} 1 & -1 & 2 & -2 \\ -1 & 1 & -2 & 2 \\ 2 & -2 & 4 & -4 \\ -2 & 2 & -4 & 4 \end{pmatrix}$$

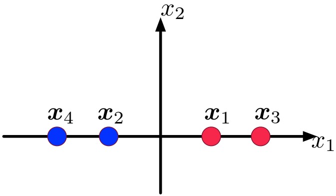
Dual formulation, by setting $C = +\infty$

$$\max_{\boldsymbol{\alpha}} \quad \sum_{n=1}^{4} \alpha_n - \frac{1}{2} \sum_{m=1,n=1}^{4} y_m y_n \alpha_m \alpha_n K_{mn}$$

$$\begin{aligned} \text{s.t.} \quad & 0 \leq \alpha_1 \leq +\infty \\ & 0 \leq \alpha_2 \leq +\infty \\ & 0 \leq \alpha_3 \leq +\infty \\ & 0 \leq \alpha_4 \leq +\infty \\ & \alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 y_3 + \alpha_4 y_4 = 0 \end{aligned}$$

Simplify a bit

$$\begin{aligned}\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{m=1,n=1}^{4} y_m y_n \alpha_m \alpha_n K_{mn} - \sum_{n=1}^{4}\alpha_n \\ \text{s.t.} \quad & 0 \leq \alpha_1 \\ & 0 \leq \alpha_2 \\ & 0 \leq \alpha_3 \\ & 0 \leq \alpha_4 \\ & \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0\end{aligned}$$



Intuition (due to symmetry

$\alpha_1 = \alpha_2$ and $\alpha_3 = \alpha_4$

Note that the linear equality in the constraint is automatically satisfied now.

Putting the value of the kernel matrix in

$$\begin{aligned} \min_{\alpha_1,\alpha_3} \quad & 2(\alpha_1^2 + 4\alpha_1\alpha_3 + 4\alpha_3^2 - \alpha_1 - \alpha_3) \\ \text{s.t.} \quad & 0 \leq \alpha_1 \\ & 0 \leq \alpha_3 \end{aligned}$$

The objective function is (after removing the prefactor of 2)

$$\left(\alpha_1 + 2\alpha_3 - \frac{1}{2}\right)^2 - \frac{1}{4} + \alpha_3 \geq \alpha_3 - \frac{1}{4}$$

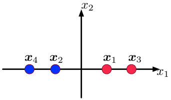
How to solve α_1 and α_3?

Since α_3 is always nonnegative, thus, to minimize the objective function, we have to set

$$\alpha_3 = 0$$

and set

$$\alpha_1 = \frac{1}{2}$$



We have shown now

$$\alpha_1 = \alpha_2 = 1/2, \quad \alpha_3 = \alpha_4 = 0$$

- Namely, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are support vectors
- $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ are removable without changing solution - obviously from the graph!
- $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ contribute equally – intuitively true too!

$$\boldsymbol{w} = \sum_n \alpha_n y_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = (1\ 0)^T$$

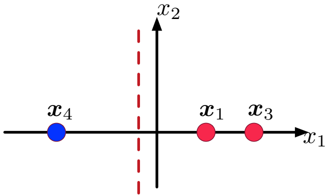
Thus, the decision boundary $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 0$ is

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = x_1 = 0$$

(I will leave out as an exercise to show $b = 0$).

Importance of support vectors

If we remove them, say x_2



and obviously the optimal decision boundary changes (to the dashed line)

Demo of SVM

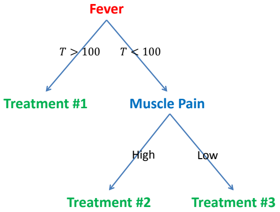
- Binary classification problem
- Nonlinear kernel

$$k(\boldsymbol{x}_m, \boldsymbol{x}_n) = e^{-\|\boldsymbol{x}_m - \boldsymbol{x}_n\|_2^2 / 2\sigma^2}$$

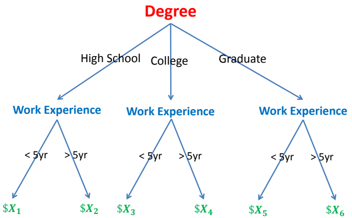
Outline

Many decisions are tree structures

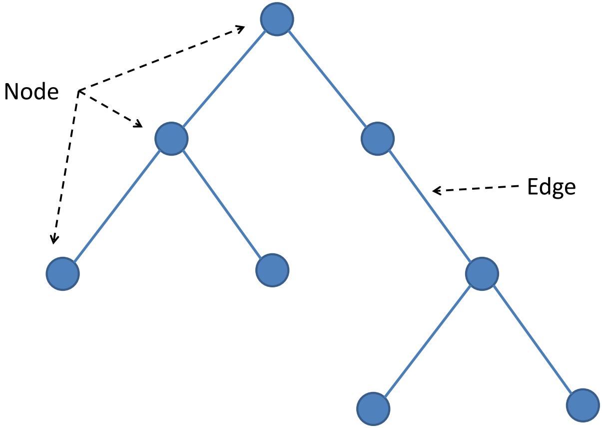
Medical treatment



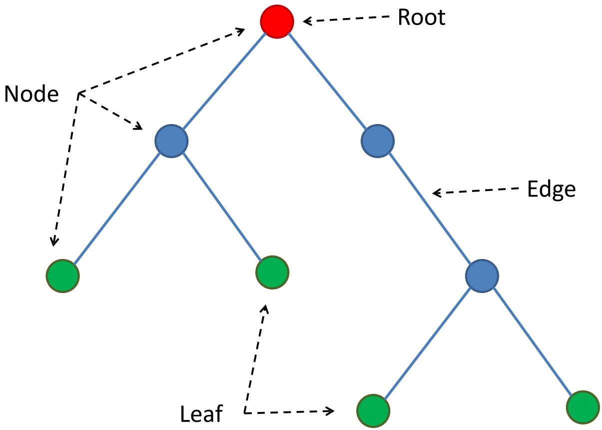
Salary in a company



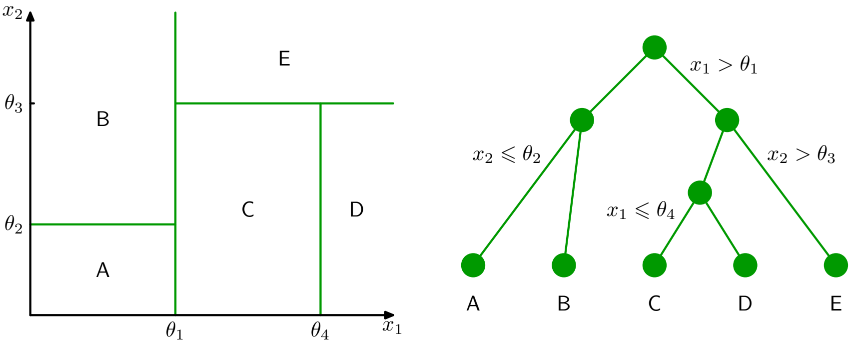
What is a Tree?



Special Names for Nodes in a Tree



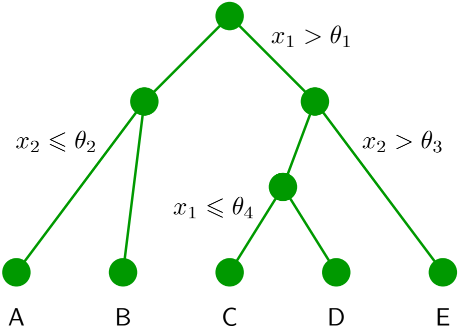
A tree partitions the feature space



Learning a tree model

Three things to learn:

1. The structure of the tree.
2. The threshold values (θ_i).
3. The values for the leafs ($A, B, \ldots$).



A tree model for deciding where to eat

Choosing a restaurant

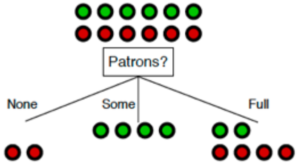
(Example from Russell & Norvig, AIMA)

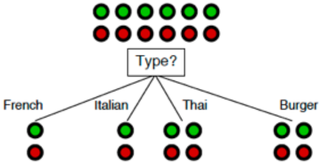
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

Classification of examples is positive (T) or negative (F)

First decision: at the root of the tree

Which attribute to split?





Patrons? is a better choice—gives **information** about the classification

Idea: use information gain to choose which attribute to split

How to measure information gain?

Idea:

Gaining information reduces uncertainty

the base can be 2 , though it is not essential (if the base is 2, the unit of the entropy is called "bit")

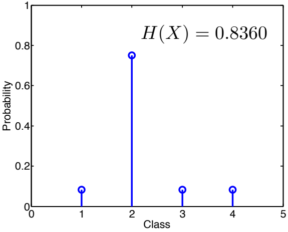
Use to entropy to measure uncertainty

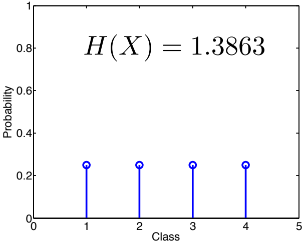
If a random variable X has K different values, a_1, a_2, ...a_K, it is entropy is given by

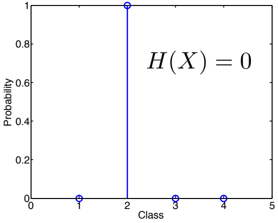
$$H[X] = -\sum_{k=1}^{K} P(X = a_k) \log P(X = a_k)$$

Examples of computing entropy

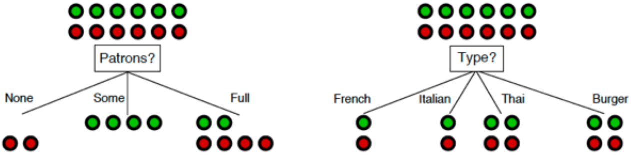
Entropy







Which attribute to split?



Patrons? is a better choice—gives **information** about the classification

Patron vs. Type?

By choosing Patron, we end up with a partition (3 branches) with smaller entropy, ie, smaller uncertainty (0.45 bit)

By choosing Type, we end up with uncertainty of 1 bit.

Thus, we choose Patron over Type.

Uncertainty if we go with "Patron"

For "None" branch

$$-\left(\frac{0}{0+2}\log\frac{0}{0+2}+\frac{2}{0+2}\log\frac{2}{0+2}\right)=0$$

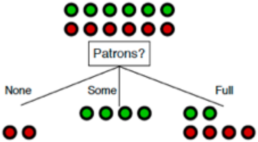
For "Some" branch

$$-\left(\frac{4}{4+0}\log\frac{4}{4+0}+\frac{4}{4+0}\log\frac{4}{4+0}\right)=0$$

For "Full" branch

$$-\left(\frac{2}{2+4}\log\frac{2}{2+4}+\frac{4}{2+4}\log\frac{4}{2+4}\right)\approx 0.9$$

For choosing "Patrons"



weighted average of each branch: this quantity is called conditional entropy

$$\frac{2}{12}*0+\frac{4}{12}*0+\frac{6}{12}*0.9=0.45$$

Conditional entropy

Definition. Given two random variables X and Y

$$H[Y|X] = \sum_k P(X = a_k) H[Y|X = a_k]$$

In our example

X: the attribute to be split

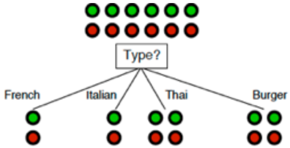
Y: Wait or not

When H[Y] is fixed, we need only to compare conditional entropy

Relation to information gain

$$\mathrm{GAIN} = H[Y] - H[Y|X]$$

Conditional entropy for Type



For "French" branch

$$-\left(\frac{1}{1+1}\log\frac{1}{1+1}+\frac{1}{1+1}\log\frac{1}{1+1}\right)=1$$

For "Italian" branch

$$-\left(\frac{1}{1+1}\log\frac{1}{1+1}+\frac{1}{1+1}\log\frac{1}{1+1}\right)=1$$

For "Thai" and "Burger" branches

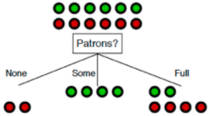
$$-\left(\frac{2}{2+2}\log\frac{2}{2+2}+\frac{2}{2+2}\log\frac{2}{2+2}\right)=1$$

For choosing "Type"

weighted average of each branch:

$$\frac{2}{12}*1+\frac{2}{12}*1+\frac{4}{12}*1+\frac{4}{12}*1=1$$

next split?

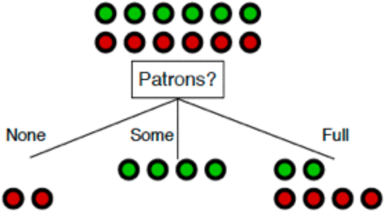


We will look only at the 6 instances with Patrons == Full

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

Classification of examples is positive (T) or negative (F)

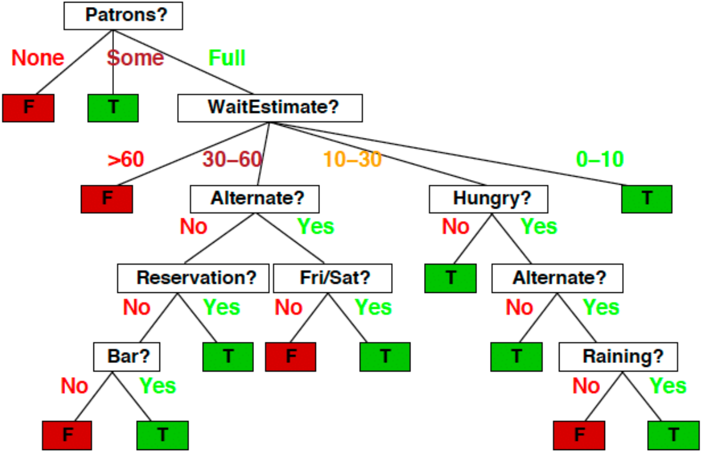
Do we split on “Non” or “Some”?



No, we do not

The decision is deterministic, as seen from the training data

Greedily we build the tree and get this



How deep should we continue to split?

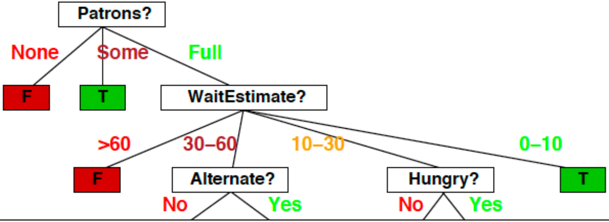
We should be very careful about this

Eventually, we can get all training examples right. But is that what we want?

The maximum depth of the tree is a hyperparameter and should not be tuned by training data — this is to prevent overfitting (we will discuss later)

Control the size of the tree

We would prune to have a smaller one



If we stop here, not all training sample would be classified correctly.

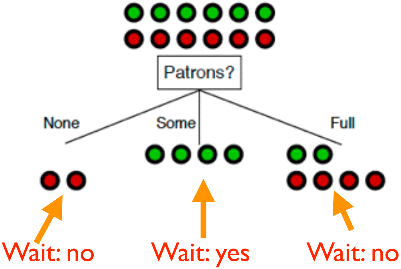
More importantly, how do we classify a new instance?

We label the leaves of this smaller tree with the majority of training samples' labels

Example

Example

We stop after the root (first node)



Splitting and Stopping Criteria

For every leaf m, define the node impurity $Q(m)$ as:

Misclassification error	$\frac{1}{N_m}\sum_{i\in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk}$.
Gini Index	$\sum_{k\neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$.
Cross-entropy	$-\sum_{k=1}^{K} \hat{p}_{mk}\log \hat{p}_{mk}$.

The **Misclassification Error** is less sensitive to changes in class probability:

$\Rightarrow$ Use Gini Index or Cross-entropy for growing T_0,

$\Rightarrow$ Use Misclassification Error for pruning T_0 and finding T.

Summary of learning trees

Other ideas in learning trees

- There are other ways of splitting attributes, such as Gini index.
- There are other fast ways of learning tree models.
- There are approaches of learning an ensemble of tree models (more on this later)

Advantages of using trees

- The models are transparent: easily interpretable by human (as long as the tree is not too big)
- It is parametric thus compact: unlike NNC, we do not have to carry our training instances around