CS-E4830 Kernel Methods in Machine Learning 2017: Tips for vectorizing in Matlab

Matlab is optimized for operations involving matrices and vectors. Vectorized code that uses vector and matrix operations often runs much faster than the corresponding code containing loops.

Simple example:

Here is one way to compute the sine of 1001 values ranging from 0 to 10:

```
i = 0;
for t = 0:.01:10
i = i + 1;
y(i) = sin(t);
end
```

A vectorized version of the same code is

```
t = 0:.01:10;
y = sin(t);
```

The second example executes usually faster than the first.

Logical indexing

A logical expression applied to a vector or matrix returns a vector or matrix containing 0 and 1. Logical expressions can be used to select specific entries from a matrix.

Examples:

- `>> Z = X(X>0); % Z is a vector containing all the positive entries of X`
- `>> X(X<0) = 0; % Replace all negative entries of X with 0`
- `>> a = find(X); % a is a vector containing the indices of the nonzero elements of X`
- `>> a = find(X==12); % a is a vector containing the indices of the elements of X equal to 12`

Commonly used functions for vectorizing

- `sum`: sum of array elements
- `repmat`: repeat copies of array
- `find`: find indices and values of nonzero elements

- `ones, zeros`: array of ones or zeros (useful for initializing vectors or matrices)