# CMSC 25400: Machine Learning
# Problem set 1

Rachel Hwang

January 24, 2014

Discussed with: Andrew Ding and James Porter.

1. An algorithm can be derived from the elimination algorithm discussed in class as follows: given any example from a training set, each feature is first negated. The elimination algorithm can then be run as usual. The negated resulting concept is then negated, then returned.

   (a) Given $x = (x_1, \ldots x_k)$, $x \in \{0, 1\}$, run elimination algorithm with input $x = (\neg x_1, \ldots \neg x_n)$,

   (b) Take resulting concept $f(x) = \neg x_1 \vee \cdots \vee \neg x_k$ and return the concept negated, $f(x) = \neg(\neg x_1 \vee \cdots \vee \neg x_k)$.

   Since De Morgan's Law states that $\neg(A \vee B) = (\neg A) \wedge (\neg B)$, given any monotone conjunction, we can first learn an equivalent monotone disjunction using the elimination algorithm in the procedure described above. By negating the results of the elimination algorithm run with input with negated attributes, we find $f(x) = \neg(\neg x_1 \vee \cdots \vee \neg x_k) = (\neg\neg x_1 \wedge \cdots \wedge \neg\neg x_k) = (x_1 \wedge \cdots \wedge x_k)$

2. Claim: $\forall A \exists A'$ with mistake bound $n$ where $A'$ is conservative and returns the same true concept as $A$. Given any algorithm $A$, we can design a conservative algorithm $A'$ in the following manner. Given examples $x_1, \ldots x_n$, let our *intermediate* concepts be $f_1, \ldots, f_n$ and the true concept be called $f_t$.

```
if f_t(x_i) = f_i-1(x_i) {
    f_i <- f_i-1
} else {
    behave as A' would
}
```

   Given this definition of A', one of two cases must be true: either there exists some sequence of inputs where A' makes more than n mistakes, or it does not.
   Now, by way of contradiction, assume that $A'$ has a mistake bound $b$ where $b > n$.

   Because $\{x | f_i(x) \neq f_{i-1}(x)\} \subset x$, and $A$'s mistake bound is $n$, the number of mistakes $A$ can make over that subset must be $\leq n$. However, consider that $A(\{x | f_i(x) \neq f_{i-1}(x)\}) = A'(x)$ by construction behave exactly the same. This gives us that $b \leq n$, which contradicts our assumption. Thus, $A'$ cannot have a mistake bound greater than $A$.

3. We assume that the distance $d$ is the Euclidean distance, and that $\vec{m^i} = \frac{1}{|C_i|}\sum_{\vec{x}\in C_i}\vec{x}$.

$$
\begin{aligned}
cost_{avg^2} &= \sum_{i=1}^{k}\sum_{\vec{x}\in C_i} d(\vec{x},\vec{m^i}) \\
&= \sum_{i=1}^{k}\sum_{\vec{x}\in C_i}\sum_{j=1}^{d}(x_j - m_j^i)^2 \\
&= \sum_{i=1}^{k}\sum_{\vec{x}\in C_i}\sum_{j=1}^{d} x_j^2 - 2x_j m_j^i + (m_j^i)^2 \\
&= \sum_{i=1}^{k}\sum_{j=1}^{d}\sum_{\vec{x}\in C_i} x_j^2 - 2x_j m_j^i + (m_j^i)^2 \\
&= \sum_{i=1}^{k}\sum_{j=1}^{d}\left(\sum_{\vec{x}\in C_i} x_j^2 - \sum_{\vec{x}\in C_i} 2x_j m_j^i + \sum_{\vec{x}\in C_i}(m_j^i)^2\right) \\
&= \sum_{i=1}^{k}\sum_{j=1}^{d}\left(\sum_{\vec{x}\in C_i}(x_j^2) - 2|C_i|(m_j^i)^2 + |C_i|(m_j^i)^2\right) \\
&= \sum_{i=1}^{k}\sum_{j=1}^{d}\left(\sum_{\vec{x}\in C_i} x_j^2 - \frac{1}{|C_i|}\sum_{\vec{x}\in C_i}\sum_{\vec{x'}\in C_i} x_j x_j'\right) \\
cost_{IC} &= \sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{\vec{x}\in C_i}\sum_{x'\in C_i} d(\vec{x},\vec{x'})^2 \\
&= \sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{\vec{x}\in C_i}\sum_{x'\in C_i}\sum_{j=1}^{d}(x_j - x_j')^2 \\
&= \sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{\vec{x}\in C_i}\sum_{x'\in C_i}\sum_{j=1}^{d} x_j^2 - 2x_j x_j' + (x_j')^2 \\
&= \sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{j=1}^{d}\sum_{\vec{x}\in C_i}\sum_{x'\in C_i} x_j^2 - 2x_j x_j' + (x_j')^2 \\
&= \sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{j=1}^{d}\left(\sum_{\vec{x}\in C_i}\sum_{x'\in C_i} x_j^2 - 2\sum_{\vec{x}\in C_i}\sum_{x'\in C_i} x_j x_j' + \sum_{\vec{x}\in C_i}\sum_{x'\in C_i}(x_j')^2\right) \\
&= \sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{j=1}^{d}\left(|C_i|\sum_{\vec{x}\in C_i} x_j^2 - 2\sum_{\vec{x}\in C_i}\sum_{x'\in C_i} x_j x_j' + |C_i|\sum_{x'\in C_i}(x_j')^2\right) \\
&= \sum_{i=1}^{k}\sum_{j=1}^{d}\left(\sum_{2\vec{x}\in C_i} x_j^2 - \frac{2}{|C_i|}\sum_{\vec{x}\in C_i}\sum_{\vec{x'}\in C_i} x_j x_j'\right)
\end{aligned}
$$

We conclude that $cost_{avg^2} = cost_{IC}$.

4. a. Python implementation can be found in hw1/hw1.py.

   b. See code.

   c. See images below. The compressed image has a higher contrast than the original image. Generally speaking, transitions between regions or two different colors look sharper, since there are far fewer intermediate colors between them. The compression is most obvious in areas where the original image was gradated, specifically, such as the blue part of the mandrill's face, the nostrils or the orange fur to the side. Overall, however, the image quality is quite good given the amount of storage space saved using the compression. The differences are not immediately noticable, and probably would scarcely be noticable at all if the original was not inspected with it side-by-side.

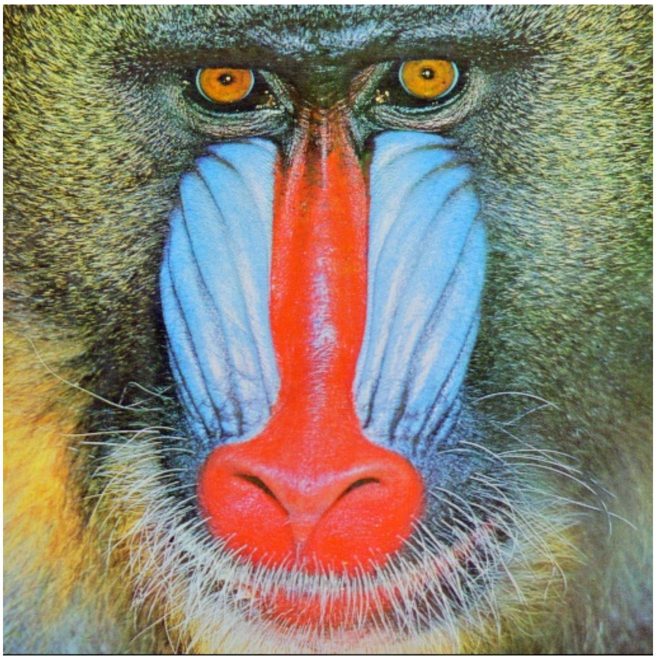

Figure 1: This is the original image.

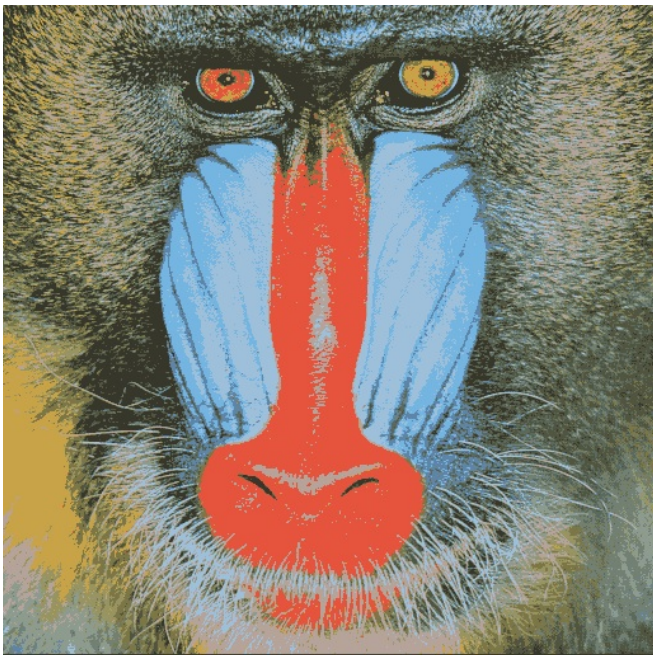

Figure 2: This is the compressed image.

d. This image compression technique, using the k-means algorithm with 16 clusters reduces the storage size by a factor of 6. Originally, each pixel required 8+8+8=24 bits. The compressed 16-color image requires only $log_2 16 = 4$ bits per pixel.