

Machine Learning and Computational Statistics

Homework 0: L^AT_EX With Program Listings

Due: Monday, February 29, 2016, at 6pm (Submit via NYU Classes)

Instructions: Your answers to the questions below, including plots and mathematical work, should be submitted as a single PDF file. It's preferred that you write your answers using software that typesets mathematics (e.g. L^AT_EX, L^AT_EX, or MathJax via iPython), though if you need to you may scan handwritten work. You may find the `minted` package convenient for including source code in your L^AT_EX document. If you are using L^AT_EX, then the `listings` package tends to work better.

1 Listings Package

While the `minted` package works nicely with plain L^AT_EX, with L^AT_EX the `listings` package tends to work better.

2 Including Python Code from Python File

Here we're extracting lines 4 through 13 from the file `code.py`.

```
def dotProduct(d1, d2):
    """
    @param dict d1: a feature vector represented by a mapping from a feature (string) to
    a weight (float).
    @param dict d2: same as d1
    @return float: the dot product between d1 and d2
    """
    if len(d1) < len(d2):
        return dotProduct(d2, d1)
    else:
        return sum(d1.get(f, 0) * v for f, v in d2.items())
```

3 Python Code Inline

```
def increment(d1, scale, d2):
    """
    Implements d1 += scale * d2 for sparse vectors.
    @param dict d1: the feature vector which is mutated.
    @param float scale
    @param dict d2: a feature vector.
```

```
NOTE: This function does not return anything, but rather
increments d1 in place. We do this because it is much faster to
change elements of d1 in place than to build a new dictionary and
return it.
"""
for f, v in d2.items():
    d1[f] = d1.get(f, 0) + v * scale
```