selectThreshold() - use the tips in the function script template, and the bulleted list on page 6 of ex8.pdf, to compute each of the tp, fp, and fn values. Sample code for "fp" is given in the text box on the bottom of ex8.pdf - page 6.

Note: error in ex8\_cofi.m [(click this link)](https://www.coursera.org/learn/machine-learning/discussions/YD0v9TL_EeWj5iIACwIAYw)

Tip for estimateGaussian(): Compute the mean using "mean()". You can compute sigma2 using the equation in ex8.pdf, or you can use "var()" if you set the OPT parameter so it normalizes over the entire sample size.

Implementation Note: In order to compute tp, fp and fn, you may be able to use a vectorized implementation rather than loop over all the examples. This can be implemented by Octave/MATLAB’s equality test between a vector and a single number. If you have several binary values in an n-dimensional binary vector v ∈ {0,1}n, you can ﬁnd out how many values in this vector are 0 by using: sum(v == 0). You can also apply a logical and operator to such binary vectors. For instance, let cvPredictions be a binary vector of the size of your number of cross validation set, where the i-th element is 1 if your algorithm considers x(i) cv an anomaly, and 0 otherwise. You can then, for example, compute the number of false positives using: fp = sum((cvPredictions == 1) & (yval == 0)).

 tutorial for cofiCostFunc()

Tom Mosher

Mentor[Week 9](https://www.coursera.org/learn/machine-learning/discussions/weeks/9) · [4 years ago](https://www.coursera.org/learn/machine-learning/discussions/weeks/9/threads/92NKXCLBEeWM2iIAC0KUpw) · Edited

Vectorized tutorial for cost and gradients with regularization

Definitions:

R: a matrix of observations (binary values). Dimensions are (movies x users)

Y: a matrix of movie ratings: Dimensions are (movies x users)

X: a matrix of movie features (0 to 5): Dimensions are (movies x features)

Theta: a matrix of feature weights: Dimensions are (users x features)

- Compute the predicted movie ratings for all users using the product of X and Theta. A transposition may be needed.

Dimensions of the result should be (movies x users).

- Compute the movie rating error by subtracting Y from the predicted ratings.

- Compute the "error\_factor" my multiplying the movie rating error by the R matrix. The error factor will be 0 for movies that a user has not rated. Use the type of multiplication by R (vector or element-wise) so the size of the error factor matrix remains unchanged (movies x users).

(**Note**: there is a quirk in the submit grader's test case that requires you to use the R matrix to ignore movies that have had no ratings).

Calculate the cost:

- Using the formula on Page 9 of ex8.pdf, compute the unregularized cost as a scaled sum of the squares of all of the terms in error\_factor. The result should be a scalar.

- Test your code using ex8\_cofi.m and the additional test cases. You should get a passing grade for this portion from the submit script.

Calculate the gradients (ref: the formulas on Page 10 of ex8,pdf):

- The X gradient is the product of the error factor and the Theta matrix. The sum is computed automatically by the vector multiplication. Dimensions are (movies x features)

- The Theta gradient is the product of the error factor and the X matrix. A transposition may be needed. The sum is computed automatically by the vector multiplication. Dimensions are (users x features)

- Test your code, then submit this portion.

Calculate the regularized cost:

- Using the formula on the top of Page 13 of ex8.pdf, compute the regularization term as the scaled sum of the squares of all terms in Theta and X. The result should be a scalar. Note that for Recommender Systems there are no bias terms, so regularization should include all columns of X and Theta.

- Add the regularized and un-regularized cost terms.

- Test your code, then submit this portion.

Calculate the gradient regularization terms (ref: the formulas in the middle of Page 13 of ex8.pdf)

- The X gradient regularization is the X matrix scaled by lambda.

- The Theta gradient regularization is the Theta matrix scaled by lambda.

- Add the regularization terms to their unregularized values.

- Test your code, then submit this portion.