ECE551, Fall 2016 - Homework Set #1

Rev. 0 (Aug 26th), Due on Thu. Sep. 8 2016 in class

1. **Geometry of orthogonal transformations in Euclidean spaces**
   Consider the vector space $V := \mathbb{R}^N$ with the Euclidean inner product and norm

$$\langle x, y \rangle := y^T x = \sum_{i=0}^{N-1} x_i y_i, \qquad \|x\|^2 := x^T x = \langle x, x \rangle, \tag{1}$$

   and let $U \in \mathbb{R}^{N \times N}$ be an orthogonal matrix. Show that lengths and angles in V remain invariant under right multiplication by U, that is

   (i) $\|Ux\| = \|x\|$ for any $x \in \mathbb{R}^N$.

   (ii) $\langle Ux, Uy \rangle = \langle x, y \rangle$ for any $x, y \in \mathbb{R}^N$.

   Now assume that $U \in \mathbb{R}^{M \times N}$ is rectangular.

   (iii) If $M > N$ and $U^T U = I$, do (i) and (ii) still hold ?

   (iv) Show that if $M < N$ then neither of the first two results hold.

2. **Some basic properties of inner product spaces**
   Prove the following:

   (a) The Cauchy-Schwarz inequality given in (2.29).

   (b) The triangle inequality given in Definition 2.9.

   (c) Parallelogram law given in (2.28).

3. **Least-squares approximation with orthonormal bases**
   Let $\mathcal{B} = \{\varphi_0, \ldots, \varphi_{N-1}\} \subset \mathbb{R}^N$ be an orthonormal basis of $\mathbb{R}^N$ with the standard inner product and norm defined in (1). Let $\hat{\mathcal{B}} \subset \mathcal{B}$ be a subset spanning the subspace $\hat{V} := \mathrm{span}(\hat{\mathcal{B}}) \subset \mathbb{R}^N$. For every $x \in \mathbb{R}^N$, define $\hat{x}$ as

$$\hat{x} := \sum_{\varphi \in \hat{\mathcal{B}}} \langle x, \varphi \rangle \, \varphi \in \hat{V}. \tag{2}$$

   (a) Show that $\hat{x}$ is the closest vector in $\hat{V}$ to x, namely, the Euclidean distance $\|x - \hat{x}\|$ attains its lowest possible value. **Hint:** all the vectors in $\hat{V}$ can be parametrized by coordinates on $\hat{\mathcal{B}}$:

$$f(\vec{\alpha}) := \sum_{\varphi_i \in \hat{B}} \alpha_i \varphi_i.$$

   Use the *Pythagorean Theorem* to show that $\|x - f(\vec{\alpha})\| \geq \|x - \hat{x}\|$ for all $\vec{\alpha}$.

   (b) Does this result hold for inner products other than the standard Euclidean one?

4. **Signal Sets and Spaces**

We usually index signals by scalars, e.g $u[0]$, v_2 or $f(\pi)$, or sometimes by tuples, like $v[320, 240]$ (as often seen in image processing). In this problem, we recall that functions can be indexed by general sets like $I = \{0, \text{red}, 13, \text{apple}\}$.

Definition 1 (Signal) *Let S and I be two sets, where S is the set of* values*, and I is the set of* indices *(or sometimes* labels*). A* signal *is merely a function $v : I \to S$.*

Depending on context, we may denote the *value* (or valuation) at index $i \in I$ by $v[i]$ (for discrete indices), v_i (for enumerations ,usually in $\mathbb{R}^N$), or $v(t)$ (for continuous t).

Definition 2 (Set of Signals) *The set of all* signals *(functions) from I to S is*

$$S^I := \left\{ v \;\middle|\; v : I \to S \right\}, \tag{3}$$

(a) Whenever $S = \mathbb{R}$, show that $\mathbb{R}^I$ is a vector space[1] over $\mathbb{R}$ with the operations

$$(u + v)[t] := u[t] + v[t], \qquad\qquad (\alpha u)[t] := \alpha u[t] \tag{4}$$

for every $t \in I,\ u, v \in \mathbb{R}^I$ and $\alpha \in \mathbb{R}$ (refer the definition in this link). What is the zero vector in the space $\mathbb{R}^I$?

(b) For the following scenarios, describe the signal sets by choosing S and I appropriately, and determine which are *linear spaces*:

   i. Complex-valued sequences indexed by the integers.

   ii. 8-bit RGB color (three channels) digital photos of dimension $W \times H$.

   iii. 32-bit floating point buffers containing 1 second of *stereo* audio at $48KHz$.

(c) If $I_1 \subset I_2$, can we claim that $\mathbb{R}^{I_1}$ is a subspace of $\mathbb{R}^{I_2}$? Explain.

(d) Assume a finite I with N elements *enumerated* as $I = \{i_0, \ldots i_{N-1}\}$. Show that the mapping $T : \mathbb{R}^I \to \mathbb{R}^N$ defined in (5) below is linear and invertible[2]:

$$(Tu)_k := v[i_k] \qquad\qquad k = 0, \ldots, N-1 \tag{5}$$

(e) For a finite I, show that the form defined in (6) is an inner product in $\mathbb{R}^I$:

$$\langle u, v \rangle_I := \sum_{i \in I} v[i]u[i] \tag{6}$$

(f) Find a basis $\{e_\tau\}_{\tau \in I} \subset \mathbb{R}^I$ such that

$$u[t] = \langle u, e_t \rangle_I\,, \qquad\qquad t \in I \tag{7}$$

for all $t \in I$. We call this basis the *standard basis* or *reproducing kernel.*

[1] This holds whenever S is a field $\Bbbk$. Even if $S \subset \Bbbk$, linear operations are still well defined on $S^I \subset \Bbbk^I$.

[2] that means essentially that $\mathbb{R}^I$ and $\mathbb{R}^N$ are linearly *isomorphic*

5. **Computer Exercise**

- Familiarize yourself with Python basics: functions, data types (integer, strings), containers (tuples, lists, dictionaries, sets), classes, modules (`import`), list/set comprehensions, `for/while` loop, and conditional execution (`if`). All of those topics are covered in the Python2 or Python3 tutorials.
- Download the python module index_vector.py from the course website, which implements arbitrary-indexed vector type, as discussed in problem 4

(a) Define the stereo 2D integer square grid index set as:

$$I^{(N)} = \left\{ (c, i, j) \;\middle|\; 0 \leq i, j \leq N-1, \quad c \in \{L, R\} \right\}.$$

Write a function `create_grid_index(N)` that creates $I^{(N)}$ as a `set` of `tuples`:

```
>>> I = create_grid_index(2)
>>> print(I)
        {('R',0, 0), ('L',0, 0), ('R',0, 1), ('L',0, 1),
         ('R',1, 0), ('L',1, 0), ('R',1, 1), ('L',1, 1)}
```

You may use *set comprehension* or `for` loops. **Note**: the order in which the indices appear does not matter (it's a set).

(b) Initialize a vector index by some index set as follows:

```
>>> from index_vector import *
>>> v= Vector(I), u = Vector(I)
>>> print(2*v+u) # (left) scaling and addition already implemented
```

Create two vectors, `u` and `v`. Fill `u` with random values, and fill `v` with the constant 1. Then print the sum `u+v`, and the scaling $10 *$ `u`.

(c) Write two functions that implement the inner product $\langle u, v \rangle_I$ defined in (6) and the norm $\|u\|_I := \sqrt{\langle u, u \rangle_I}$ respectively. Print the norms and the inner product of of `u,v` from the last part. Be sure to assert that both `u,v` have the same index (e.g `if u.index!=v.index: print some error`).

(d) Write a function `standard_vec(I,t)` that creates the vector e_t that satisfies (7). Verify your result, that is, show that

```
>>> et = standard_vec(I,t) # For some arbitrarily chosen index
>>> print(u[t]-inner_product(u,et)) # Should give 0
```

(bonus) How would you represent and implement linear mappings in $\mathbb{R}^I$? Explain and show your code.

While abstract indexing is powerful in theoretic analysis, in practice we index signals mostly on grids of integers, namely $I \subset \mathbb{Z}^n$. The `NumPy` library (see tutorial here), which we will use in the following part, is an excellent framework in such cases.

(e) Let $p \in \mathbb{R}^N$ and $q \in \mathbb{R}^L$ be two real Euclidean vectors. The *linear convolution* of p by q, is the vector $p * q \in \mathbb{R}^{N+L-1}$ defined by

$$(p * q)[m] = \sum_{i=0}^{L-1} p[m-i]q[i] \qquad\qquad 0 \leq m \leq N+L-1 \tag{8}$$

where $p[m-i] := 0$ whenever $m-i < 0$ or $m-i \geq N$. This defines a linear operation in both entries, and can be written as a matrix multiplication:

$$p * q = T_p q = T_q p$$

where T_p, T_q are rectangular matrices of appropriate dimensions. Write a script that constructs the convolution matrix T_p as Numpy array, given a vector p, and the length L of the vector q. Your code should look roughly like

```python
def conv_matrix(p,L):
        Tp = numpy.zeros((len(p)+L-1,L))
        # ... fill in your code to generate Tp
        return Tp
```

Print the convolution matrix T_p of $p = (1, -2, 1)$ convolved with a vector of length $L = 6$. Validate your result against the Numpy `numpy.convolve()` function, namely show that `np.dot(Tp,q) - np.convolve(p,q)` vanish, for example, if $q = (1, 2, 3, 3, 2, 1)$.