CCA

In PCA we looked at only the input X to carry out dimensionality reduction and is unsupervised in that sense. CCA adds invariance and incorporates the labels y into the analysis.

In [0]:
```
import numpy as np
from sklearn.cross_decomposition import CCA
```

In [0]:
```
# some dummy data for showing the process of running CCA
X = np.array([[0., 0., 1.], [1.,0.,0.], [2.,2.,2.], [3.,5.,4.]])
y = np.array([[0.1, -0.2], [0.9, 1.1], [6.2, 5.9], [11.9, 12.3]])
print(X)
print(y)
```

```
[[0. 0. 1.]
 [1. 0. 0.]
 [2. 2. 2.]
 [3. 5. 4.]]
[[ 0.1 -0.2]
 [ 0.9  1.1]
 [ 6.2  5.9]
 [11.9 12.3]]
```

In [0]:
```
cca = CCA(n_components=1)
cca.fit(X, y)
```

Out[0]:
```
CCA(copy=True, max_iter=500, n_components=1, scale=True, tol=1e-06)
```

In [0]:
```
# more useful than simply the fit, the actual transformation from our computed C
CA
X_c, y_c = cca.fit_transform(X, y)
print(X_c)
print(y_c)
```

```
[[-1.3373174 ]
 [-1.10847164]
 [ 0.40763151]
 [ 2.03815753]]
[[-0.85511537]
 [-0.70878547]
 [ 0.26065014]
 [ 1.3032507 ]]
```

To demonstrate why CCA is more powerful, let's examine scaling the data.

In [0]:
```python
X_c, y_c = cca.fit_transform(2*X, 2*y)
print(X_c)
print(y_c)
```

```
[[-1.3373174 ]
 [-1.10847164]
 [ 0.40763151]
 [ 2.03815753]]
[[-0.85511537]
 [-0.70878547]
 [ 0.26065014]
 [ 1.3032507 ]]
```

As we see, scaling both by 2 doesn't change the transform at all. Going further:

In [0]:
```python
X_c, y_c = cca.fit_transform(2*X, 3*y)
print(X_c)
print(y_c)
```

```
[[-1.3373174 ]
 [-1.10847164]
 [ 0.40763151]
 [ 2.03815753]]
[[-0.85511537]
 [-0.70878547]
 [ 0.26065014]
 [ 1.3032507 ]]
```

Even scaling separately, the transformation remains the same. This shows a very important porperty of CCA - it's invariant to affine transformations (scaling or addition).

Proof:

In [0]:
```python
from IPython.display import Image
Image(filename="/content/data/screenshot.PNG")
```

Out[0]:

$$\begin{aligned}
\rho(aX+b, cY+d) &= \frac{\mathrm{Cov}(aX+b, cY+d)}{\sqrt{\mathrm{Var}(aX+b)\,\mathrm{Var}(cY+d)}} \\
&= \frac{\mathrm{Cov}(aX, cY)}{\sqrt{\mathrm{Var}(aX)\,\mathrm{Var}(cY)}} \\
&= \frac{a \cdot c \cdot \mathrm{Cov}(X, Y)}{\sqrt{a^2\,\mathrm{Var}(X) \cdot c^2\,\mathrm{Var}(Y)}} \\
&= \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}} \\
&= \rho(X, Y)
\end{aligned}$$

Proof Source: EECS 189 Course Notes