Non Linear Sparse Recovery Algorithm

Pallavi Das, Mansi Jain and Angshul Majumdar

Indraprastha Institute of Information Technology, Delhi

ABSTRACT

Compressed sensing addresses the problem of recovering a sparse solution to a system of linear under-determined equations. In this work we are interested in deriving algorithms when the system is non-linear. Our algorithm is based on gradient descent approach followed by subsequent soft thresholding. We have tested our algorithm for both l_2-norm and l_1-norm cost functions (data fidelity) with linear and exponential systems.

***Index Terms*—** Non linear compressed sensing, non linear sparse recovery, algorithms.

1. INTRODUCTION

Compressed Sensing (CS) studies the problem of solving an under-determined linear inverse problem when the solution is known to be sparse. Since its inception in 2006-07, CS has penetrated almost all application domains of signal processing - seismic imaging, astrophysics, medical imaging, biomedical signal processing, bio-informatics, radar imaging, to name a few.

As electrical engineers and computer scientists, we always linearize our system. Such linearity assumptions serve our purpose on most cases. But there are several applications where one cannot make such simplifying assumptions. For such scenarios, one needs to solve a non-linear inverse problem.

Literature on non-linear sparse recovery is itself sparse. There are two studies [1, 2] of theoretical nature that explores the conditions under which such recovery is possible. As engineers, we are more interested in solving the problems. There has not been any concerted effort in developing such non-linear sparse recovery algorithms. To the best of our knowledge, there is only a single work [3] (back in 2008) that proposed a greedy algorithm to solve the problem. However, greedy algorithms are almost never used in practice for standard compressed sensing problems. Both in theory and in practice, optimization based sparse recovery approaches yield better results. Therefore, there is a need to develop algorithms for non-linear sparse recovery that are based on an optimization framework. In [1], the theoretical guarantees of non-linear CS are derived for optimization based algorithms.

In the following section we motivate the readers regarding the practical need of such non-linear CS algorithms. We discuss about problems in Magnetic Resonance Imaging (MRI), non-linear sparse classification and briefly on radar imaging. The baseline for our proposed algorithm will be discussed in section 3. The actual derivation will be in section 4. The experimental results will be shown in section 5. Finally in section 6, the conclusions of the work and future directions of research are discussed.

2. NEED FOR NON-LINEAR CS

2.1. Quantitative Magnetic Resonance Imaging

Anyone familiar with MRI [4] knows that the intensity at a voxel / pixel 'i' is given by:

$$x(i) = \rho_0 (1 - e^{-T_R/T_1(i)}) e^{-T_E/T_2(i)} \tag{1}$$

where ρ_0 is the proton density (spin density), T_R is the repetition time and T_E the echo time of the applied magnetization. The T_1 and the T_2 are the tissue parameters.

In standard MRI, one is not concerned about the tissue parameters; the only problem is to recover the intensity values. For MRI, the data is acquired in the Fourier domain (K-space); the data acquisition model is expressed as:

$$y = Fx + \eta, \eta \sim \tag{2}$$

where x is the intensity image, F is the Fourier transform, η is the noise and y is the acquired K-space data.

When the K-space is fully measured, recovering the intensity image is trivial - one only needs to apply the inverse Fourier transform (followed by denoising). But, sampling the full K-space is time consuming. In recent years, the practice is to partially sample the K-space and recover the image using CS techniques [5, 6] that exploit the sparsity of the image in the transform domain. In such a scenario the data acquisition is expressed as:

$$y = RFx + \eta \tag{3}$$

The corresponding recovery is via solving the following problem:

$$\|y - RFx\|_2^2 + \lambda \|\Psi x\|_1 \tag{4}$$

where Ψ is the sparsifying transform.

Such techniques are good for recovering intensity images; unfortunately such images are not quantitative. Referring to (1), it is easy to notice that the intensity changes depending on the scan parameters T_E and T_R. Quantitative imaging modalities are agnostic to scan parameters - T_1 / T_2 maps are such quantitative images. Unfortunately, such maps cannot be acquired directly. They need to computed from the multiple intensity images via non-linear curve fitting. There are CS based techniques to accelerate such multi-echo scans [7, 8].

But, assume an alternate scenario where we want to compute the T_2 map. In such a case, the repetition time (T_R) is kept small, so that (1) can be approximated as:

$$x(i) = \rho_0 e^{-T_E/T_2(i)} \quad (5)$$

The corresponding data acquisition could be expressed as:

$$y = RF\rho e^{-T_E/T_2} + \eta \quad (6)$$

If we had non-linear sparse recovery techniques at our disposal, we could directly compute the T_2 maps from one or two images. This is a massive reduction in data acquisition times compared to multi-echo scans where at least 32 images are needed for estimating the same map.

2.2. Non-linear Sparse Classification

In the seminal paper on sparse classification [9] it is assumed that training samples of any class form a linear basis for representing test samples of the same class; expressed mathematically,

$$v_{k,test} = \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + ... + \alpha_{k,n_k}v_{k,n_k} + \varepsilon \quad (7)$$

where $v_{k,i}$ are the training samples and ε is the approximation error.

Equation (7) expresses the assumption in terms of the training samples of a **single** class. Alternately, it can be expressed in terms of **all** the training samples so that

$$v_{k,test} = V\alpha + \varepsilon \quad (8)$$

where $V = [v_{1,1} | ... | v_{n,1} | ... | v_{k,1} | ... | v_{k,n_k} | ... v_{C,1} | ... | v_{C,n_C}]$

and $\alpha = [\alpha_{1,1} ... \alpha_{1,n_1} ... \alpha_{k,1} ... \alpha_{k,n_k} ... \alpha_{C,1} ... \alpha_{C,n_C}]'$.

In a classification problem, the training samples and their class labels are provided. The task is to assign the given test sample with the correct class label. This requires finding the coefficients $\alpha_{k,i}$ in equation (1). By assumption, α is sparse since it has non-zero values only corresponding to the correct class. In [9] the solution is framed as a sparse optimization problem.

$$\|v_{k,test} - V\alpha\|_2^2 + \lambda\|\alpha\|_1 \quad (9)$$

It was pointed out in [10, 11] that there is no reason that the linearity assumption (8) should hold. In general, one can assume a non-linear form:

$$v_{k,test} = f(V\alpha) + \varepsilon \quad (10)$$

But the sparsity assumption still holds. Therefore, one required solving a non-linear sparse recovery problem:

$$\|v_{k,test} - f(V\alpha)\|_2^2 + \lambda\|\alpha\|_1 \quad (11)$$

In [10, 11], naive algorithms were proposed to solve such sparse recovery problems. The results showed improvement over the linear approach.

2.3. Through Wall Radar Imaging

First we will consider the simple problem of radar imaging of a human being in free space. Such a problem can be expressed as a linear model:

$$y = Fx + \eta \quad (12)$$

Here x is the target, y is the received data (at the radar) and F is the Fourier transform.

In order to get a well resolved image, the aperture of the radar should be large and to cover a large field-of-view the number of antennas on the radar should be large as well. Increasing the number of receiver antennas on the radar is cumbersome. Therefore in a recent study [12], it was shown that CS can be used to decrease the number of antenna elements without compromising the quality of the image.

However, free space imaging is a simplifying assumption. In all practical situations, the target is behind an obstruction (like a wall). In such a situation, the multipath effects coupled with refraction and diffraction leads to a non-linear imaging problem. The forward problem (image formation) cannot be expressed as a linear problem (12). One needs to express it in a non-linear form:

$$y = f(x) + \eta \quad (13)$$

In such a scenario, the recovery should be posed as a non-linear CS problem:

$$\|y - f(x)\|_2^2 + \lambda\|\Psi x\|_1 \quad (14)$$

where C is a constant.

In this section, we have discussed several motivating problems for non-linear CS. There are more, but given the limitations in space we refrain from discussing these topics.

3. BRIEF REVIEW OF ITERATIVE SOFT THRESHOLDING

We take the simple problem of recovering a sparse solution to a linear inverse problem:

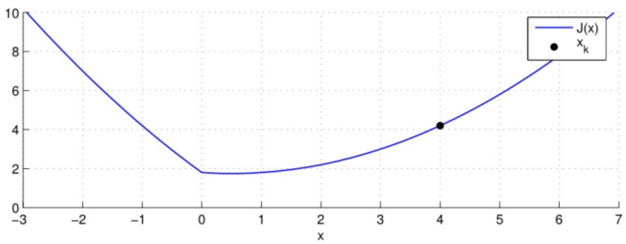
$$y = Ax + \eta \quad (15)$$

One of the best known solutions for this problem is via l_1-minimization.

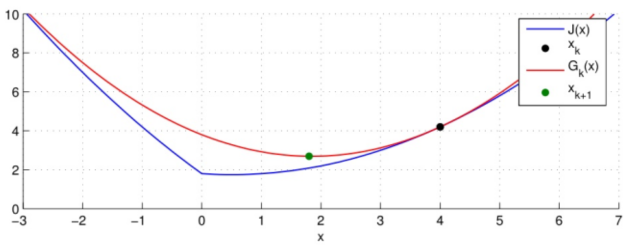
$$\|y - Ax\|_2^2 + \lambda\|x\|_1 \quad (16)$$

We know that, l_1-minimization is a quadratic programming problem that needs to be solved iteratively. Unfortunately, gradient descent cannot be directly applied owing to the non-differentiability of the l_1-penalty. In such a scenario, one of the most favoured methods to solve this problem is via the Majorization Minimization approach.

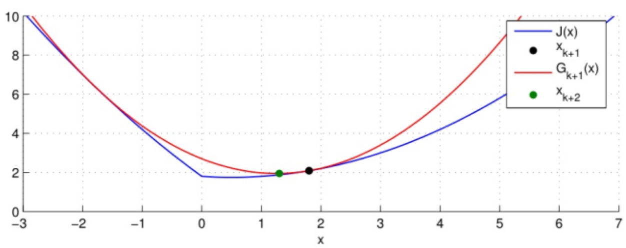
3.1. Majorization-Minimization



(a)



(b)



(c)

Fig. 1. Majorization Minmization [13]

Fig. 1 shows the geometrical interpretation behind the Majorization-Minimization (MM) approach. The figure depicts the solution path for a simple scalar problem but essentially captures the MM idea.

Let, J(x) is the function to be minimized. Start with an initial point (at k=0) x_k (Fig. 1a). A smooth function $G_k(x)$ is constructed through x_k which has a higher value than J(x) for all values of x apart from x_k, at which the values are the same. This is the Majorization step. The function $G_k(x)$ is constructed such that it is smooth and easy to minimize. At each step, minimize $G_k(x)$ to obtain the next iterate x_{k+1} (Fig 1b). A new $G_{k+1}(x)$ is constructed through x_{k+1} which is now minimized to obtain the next iterate x_{k+2}(Fig. 1c). As can be seen, the solution at every iteration gets closer to the actual solution.

3.2. Landweber Iteration

Let us consider the minimization of the following optimization problem,

$$J(x) = \|y - Ax\|_2^2$$

For this minimization, $G_k(x)$ is chosen to be,

$$G_k(x) = \|y - Ax\|_2^2 + (x - x_k)^T (aI - A^T A)(x - x_k) \quad (17)$$

where a is the maximum eigenvalue of the matrix $A^T A$. This majorizes the original cost function.

$$G_k(x) = \|y - Ax\|_2^2 + (x - x_k)^T (aI - A^T A)(x - x_k)$$
$$= y^T y - 2y^T Ax + x^T A^T Px + (x - x_k)^T (aI - A^T A)(x - x_k)$$
$$= y^T y + x_k^T (aI - A^T A)x_k - 2(y^T A + x_k^T (aI - A^T A))x + ax^T x$$
$$= a(-2b^T x - x^T x) + c$$

where $b = x_k + \frac{1}{a} A^T (y - Ax_k)$, $c = y^T y + x_k^T (aI - A^T A)x_k$

Using the identity $\| b - x \|_2^2 = b^T b - 2b^T x + x^T x$, one can write,

$$G_k(x) = a \| b - x \|_2^2 - ab^T b + c$$
$$= a \| b - x \|_2^2 + K$$

where K consists of terms independent of x.

Therefore, minimizing (17) is the same as minimizing the following,

$$G_k^{'}(x) = \| b - x \|_2^2 \quad (18)$$

where $b = x_k + \frac{1}{a} A^T (y - Ax_k)$.

This update is known as the Landweber iteration.

3.3. Iterative Soft Thresholding

After combining the Landweber iterations (18) with the original problem (16), we get the following problem in each iteration,

$$\min_x \|b - x\|_2^2 + \frac{\lambda}{a} \|x\|_1 \quad (19)$$

The above function (19) is actually de-coupled, i.e.

$$\|b - x\|_2^2 + \frac{\lambda}{a} \|x\|_1 = \sum_i (b(i) - x(i))^2 + \frac{\lambda}{a} |x(i)| \quad (20)$$

Therefore, (20) can be minimized term by term,

$$\frac{\partial}{\partial x(i)} \left[\|b - x\|_2^2 + \frac{\lambda}{a} \|x\|_1 \right] = 2b(i) - 2x(i) + \frac{\lambda}{a} signum(x(i))$$

Setting the partial derivatives to zero and solving gives,

$$x(i) = signum(b(i)) \max(0, | b(i) | - \frac{\lambda}{2a})$$

Written compactly in matrix vector form:

$$x = signum(b) \max(0, | b | - \frac{\lambda}{2a})$$

This leads to a simple two-step iterative solution for the l_1-minimization problem (16).

Initialize: $\alpha_0 = 0$

Repeat until convergence

Step 1. $b = x_k + \frac{1}{a} A^T (y - Ax_k)$

Step 2. $x = signum(b)\max(0, |b| - \frac{\lambda}{2a})$
End

4. PROPOSED ALGORITHM

The iterative soft thresholding algorithm has two steps in every iteration. The first step is the Landweber iteration, followed by soft thresholding. A closer look at the Landweber iteration reveals that it is a gradient descent step. $A^T(y-Ax_k)$ is the negative of the gradient - and hence the descent direction; and $1/a$ is the step-size.

Intuitively one can understand the IST algorithm. It is fundamentally a gradient descent method. Therefore, it is natural that there is a gradient descent step - Landweber iteration. But, it is known that the solution we are seeking for is sparse. To enforce sparsity we have the additional soft thresholding step.

Compressed Sensing solves a linear inverse problem:

$$y = Ax + \eta$$

Since the noise is Gaussian, the cost function to minimize is the Euclidean norm. But, the additional assumption is that the solution is sparse. Hence the l_2-norm data fidelity term is regularized by an l_1-norm penalty on the solution.

$$\min_x \|y - Ax\|_2^2 + \lambda \|x\|_1$$

The IST algorithm as per our intuitive understanding can also be explained in two steps. The cost function we are interested in minimizing is the l_2-norm data fidelity. The Landweber iteration, does just that. The l_1-norm promotes a sparse solution. In the algorithm, we promote the sparsity by thresholding the gradient update step. The Landweber iteration can be alternately expressed as follows:

Initialize: $\alpha_0 = 0$
Repeat until convergence
Step 1. $b = x_k + \sigma \nabla \|y - Ax\|\big|_{x=x_k}, \sigma = 1/a$
Step 2. $x = signum(b)\max(0, |b| - \tau), \tau = \lambda / 2a$
End

The IST algorithm, when expressed in this form, looks more attuned to the well known gradient descent algorithms we know.

In non-linear sparse recovery the inverse problem is as follows:

$$y = f(x) + \eta \quad (20)$$

The sparse recovery problem in the non-linear scenario can be expressed as:

$$\min_x \|y - f(x)\|_2^2 + \lambda \|x\|_1 \quad (21)$$

Following the same intuition as before, we can have an algorithm that minimizes the l_2-norm data fidelity term followed by a sparsity promoting step. We will have a non-linear IST algorithm.

Initialize: $\alpha_0 = 0$
Repeat until convergence
Step 1. $b = x_k + \sigma \nabla \|y - f(x)\|_2^2\big|_{x=x_k}$
Step 2. $x = signum(b)\max(0, |b| - \tau)$
End

This algorithm can only yield a solution when certain conditions are met. Previously the Moreau's proximal operator and the Lipschitz conditions have been used to derive accelerated IST algorithms like FISTA [14]. The general idea is to define a proximal operator that solves the composite non-smooth problem given by

$$F(x) = h(x) + g(x) \quad (22)$$

where h(x) is smooth function with a Lipschitz continuous gradient L:

$$\|\nabla h(x) - \nabla h(z)\|_2 \le L\|x - z\|_2, \forall x, z \in \mathbb{R} \quad (23)$$

and $g(x)$ is a non-smooth convex function.

For our case, $f(x) = \|y - f(x)\|_2^2$ and $g(x) = \lambda\|x\|_1$.

When the condition (23) is satisfied, the step size for the descent algorithm is determined by the upper bound on L, i.e. $\sigma \le \frac{1}{L}$.

Signal processing engineers are aware that the bounds and conditions set by CS are rather pessimistic. Although in theory one would require h(x) to be continuously differentiable, we will show that our proposed algorithm also works for non-differentiable functions such as the l_1-norm data fidelity.

5. EXPERIMENTAL EVALUATION

The only algorithm that can be used as a baseline for non-linear sparse recovery is [3], which is an approximate greedy technique. We conduct our experiments with two different functions - linear ($y=Ax$) and exponential ($y=exp(Ax)$), each with two different data fidelity terms - Euclidean (l_2-norm) and Manhattan (l_1-norm). The matrix A is i.i.d Gaussian. The experiments were conducted for different sampling ratios' sampling ratio is defined as the ratio between the number of columns and number of rows in A. The reconstruction error is measured in terms of Normalized Mean Squared Error (NMSE) defined as:

$$NMSE = \frac{\|original - reconstructed\|_2}{\|original\|_2}.$$

For each sampling ratio, the matrix was generated a hundred times, and the average NMSE is reported.

In Tables 1 to 4, the results linear function with l_2-norm data fidelity, linear function with l_1-norm data fidelity, exponential function with l_2-norm data fidelity and exponential function with l_1-norm data fidelity are reported respectively. The results are compared against the non-linear greedy algorithm [3].

For all the experiments the size of the matrix A was fixed at 60 X 100. The number of non-zero entries in the signal is varied. The variation of reconstruction error is reported.

Table 1. Error on Linear function with l_2-norm data fidelity

# Non-zero values	σ, τ	NMSE proposed	NMSE Greedy algo [3]
5	0.01, 5	.0073	0.0761
10	0.01, 5	.0115	0.0854
15	0.01, 5	.0138	0.0985
20	0.01, 5	.0441	0.1569

Table 2. Error on Linear function with l_1-norm data fidelity

# Non-zero values	σ, τ	NMSE proposed	NMSE Greedy algo [3]
5	0.01, 5	.0293	0.1058
10	0.01, 5	.0348	0.1394
15	0.01, 5	.0261	0.1661
20	0.01, 5	.0680	0.1973

Table 3. Error on Exponential function with l_2-norm data fidelity

# Non-zero values	σ, τ	NMSE proposed	NMSE Greedy algo [3]
5	0.02, 0.03	0.0050	0.0585
10	0.02, 0.03	0.0120	0.1124
15	0.02, 0.03	0.0292	0.1576
20	0.02, 0.03	0.0251	0.2123

Table 4. Error on Exponential function with l_2-norm data fidelity

# Non-zero values	σ, τ	NMSE proposed	NMSE Greedy algo [3]
5	0.001, 0.05	0.0234	0.1058
10	0.001, 0.05	0.0386	0.1644
15	0.001, 0.05	0.0580	0.1502
20	0.001, 0.05	0.0942	0.2813

We observe that our proposed method always yields better results than the greedy algorithm we compared against. This is not surprising, as it is well known that optimization based methods (with proper choice of parameters) yield better results than greedy algorithms.

We carried out a simple t-test to determine if our method yields significantly different result compared to the non-linear greedy algorithm. We found that our method is significantly superior than the other at 99% confidence.

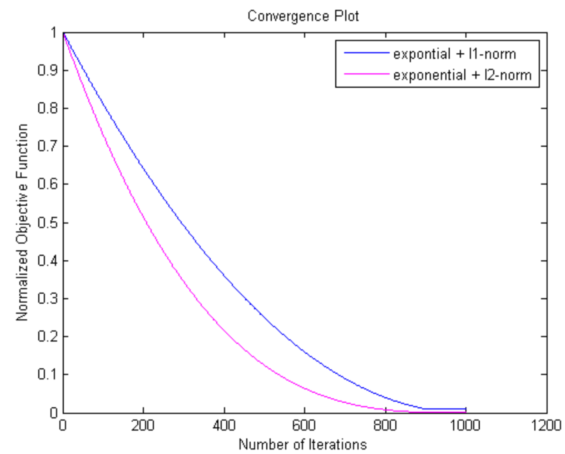


Fig. 1. Convergence of plot of proposed algorithm for exponential functions

In Fig. 1, we show the convergence plot of our proposed algorithm for the exponential function with l_2-norm and l_1-norm data fidelity terms. The convergence is very slow. This is because the step size used in our approach (following Lipschitz condition) is too small (pessimistic).

6. CONCLUSION

Our paper is one of the first algorithmic works on non-linear sparse recovery. Prior studies [1, 2] concentrated on the theory of non-linear compressed sensing. We have conclusively shown that our proposed method is better than the only existing algorithm [3] on non-linear sparse recovery.

However we must admit that our work is a proof-of-concept. There are two issues with our current approach. The first one is the convergence. The step size is based on the Lipschitz condition. This is a pessimistic bound. We must find ways to increase the step size in order to accelerate convergence.

The other issue is with the way we are computing the gradient in each iteration. In order to keep our method general, we are using a numerical gradient. This is slow. Better results can be obtained if the gradient is computed analytically. These issues need to be addressed, before such non-linear sparse recovery algorithms can be used for large scale problems.

REFERENCES

[1] T. Blumensath; "Compressed Sensing with Nonlinear Observations", Information Theory, IEEE Transactions on , vol.59, no.6, pp.3466,3474, June 2013.

[2] Beck, A; Eldar, Y.C.; Shechtman, Y., "Nonlinear compressed sensing with application to phase retrieval," Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE , vol., no., pp.617,617, 3-5 Dec. 2013.

[3] T. Blumensath and M. E. Davies; "Gradient Pursuit for Non-Linear Sparse Signal Modelling", European Signal Processing Conference (EUSIPCO), Lausanne, Switzerland, April 2008.
[4] E. M. Haacke, R. R. Brown, M. R Thomson and R. Venkatesh, Magnetic Resonance Imaging: Physical Principals and Sequence Design, John Wiley and Sons, 1999.
[5] M. Lustig, D.L Donoho and J.M Pauly, "Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging", Magnetic Resonance in Medicine, Vol. 58(6):1182-1195, 2007.
[6] J. Trzasko, and A. Manduca, "Highly Undersampled Magnetic Resonance Image Reconstruction via Homotopic l0-Minimization", IEEE Trans. on Medical Imaging, Vol. 28 (1), pp. 106-121, 2009M.
[7] A. Majumdar and R. K. Ward, "Accelerating Multi-Echo T2 Weighted MR Imaging: Analysis Prior Group Sparse Optimization", Journal of Magnetic Resonance, Vol. 210 (1), pp. 90-97, 2011.
[8] A. Majumdar and R. K. Ward, "Joint Reconstruction of Multi-echo MR Images Using Correlated Sparsity", Magnetic Resonance Imaging, Vol. 29 (7), pp. 899-906, 2011.
[9] J Wright, AY Yang, A Ganesh, SS Sastry, Y Ma, "Robust face recognition via sparse representation", Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (2), 210-227, 2009.
[10] A. Majumdar and R. K. Ward, "Non-linear Sparse and Group-sparse Classifier", IEEE Canadian Conference on Electrical and Computer Engineering.
[11] Majumdar, A; Ward, R.K.; Aboulnasr, T., "Generalized Non-linear Sparse Classifier," Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European , vol., no., pp.1,5, 9-13 Sept. 2013.
[12] A. Majumdar and S. S. Ram. "Two-Dimensional Array Processing with Compressed Sensing", RadarCon 2014
[13] http://cnx.org/content/m32168/latest/ : Last Accessed September 3, 2014
[14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM. on Imaging Sciences 2(1), 2009
[15] Paul Rodríguez and Brendt Wohlberg, "An Efficient Algorithm for Sparse Representations with lpData Fidelity Term", in Proceedings of 4th IEEE Andean Technical Conference (ANDESCON), (Cusco, Perú), Oct 2008