Et voilà le bilan du point de synchro de ce matin :   
*(See attached file: 160111.zip)*  
  
J'en déduis à peu près le script suivant en relisant et en remettant "dans l'ordre" le contenu des photos - je vous laisserai relire, reformuler, triturer... jusqu'à obtenir la substantifique moëlle :)  
  
  
1ère photo (**2016-01-11 09.56.18\_formalisationBase\_1.jpg**)  
1- (en haut à gauche photo) on pose X = la matrice contenant les x\_ij' (les x\_ij sont en lignes comme des vecteurs de features). Dans X toutes les paires (i,j) sont représentées quelque soit la classe (taille de X : n² x p)  
\* note: en fait le seul endroit où l'on va se servir de cette notation est dans le QP pour aller dans le dual. On se servira en particulier d'une sous-matrice de X, que l'on pourrait appeler X\_pull, qui ne contient que les paires (i, Pull(i) pour tout i).  
2- (en haut à gauche photo) on rappelle que notre objectif est de trouver une combinaison linéaire des dh D(x\_i,x\_j) = sum(w\_h\*d\_h(x\_i,x\_j)).   
\* (dans une photo ultérieure) on rappelle aussi que l'on vise certaines propriétés pour D. Expliquer ce que cela implique comme conditions sur les w\_h et les d\_h, en particulier la contrainte w\_h>=0  
3- (en haut à droite photo) on décrit l'espace des paires. On en profite pour donner quelques interprétations (cf ce que Cao a déjà écrit) - voici quelques points importants qui me viennent à l'esprit:   
\* la norme dans cet espace représente une distante (c'est la distance D\_carrée =sum(d(x\_i,x\_j)²) , à laquelle on pourrait d'ailleurs donner une notation particulière peut-être pour l'utiliser facilement par la suite ?)  
\* une distance euclidienne faible dans cet espace ( sum( **(d(x\_i,x\_j)-d(x\_k,x\_l))²**  ) ) ne représente en rien une similarité entre les time series MAIS une similarité entre les différences (la différence entre x\_i et x\_j porte sur les mêmes modalités, dans les mêmes proportions et les mêmes quantités, que la différence entre x\_k et x\_l)  
\* (new - de mes réflexions dans le tram retour) si deux vecteurs x\_ij et x\_kl sont proportionnels, c'est à dire x\_ij = alpha \* x\_kl, alors la différence entre x\_i et x\_j porte sur les mêmes modalités et dans les mêmes proportions que la différence entre x\_k et x\_l.  
4- (à droite photo) on exprime notre objectif dans cet espace : D(x\_ij)=w'x\_ij qui peut se voir comme la norme 1 d'une transformation linéaire des vecteurs x\_ij : D(x\_ij)=||Wx\_ij||\_1. On peut illustrer cela avec le plot de la dernière photo (**2016-01-11 11.41.36\_distancePlots.jpg**, plot de droite qui montre les sphères unité de différentes fonctions D, y compris la notre)  
  
Note: pour poser le problème on utilisera plutôt la photo suivante.  
  
  
2e photo (**2016-01-11 10.17.39\_formalisationBase\_2.jpg**)  
5- on propose une formalisation du problème plus générale que celle de Weinberger. Pour cela   
\* on définit tout d'abord deux ensembles Pull(i) et Push(i).   
\* (milieu photo) on pose le problème que l'on propose (dans l'espace initial, mais sans les carrés de Weinberger et avec les ensembles Push(i) et Pull(i))  
\* on explique que l'on a enlevé les carrés pour que le problème soit plus général, mais que le problème de Weinberger s'y retrouve si on pose D(x\_i,x\_j)=D²(x\_i,x\_j)  
\* (en haut à droite photo) On explique ce que sont les push(i) et pull(i) chez Weinberger, avec les avantages et inconvénients  
\* (à droite photo)  puis on propose notre version avec les avantages et inconvénients  
  
6- maintenant on peut poser notre problème dans l'espace pairwise : photo milieu bas.   
\* on garde l'écriture avec des sommes sur Pull (pas besoin de faire apparaître la matrice X\_pull à ce stade je pense, on l'utilisera seulement pour le QP).   
\* Par contre par rapport à la photo on n'a peut être plus besoin de la notation ||Wx\_ij||\_1, on peut éventuellement revenir à la notation plus simple w'x\_ij (maintenant que l'on a vu qu'il n'y a pas nécessairement besoin de faire apparaitre cette norme 1 -ci).  
\* on ajoute bien en contrainte le w\_h >= 0 dans le problème en rappelant pourquoi (cf point 2 ci-dessus)  
  
  
3e photo (**2016-01-11 11.11.43\_L1\_L2\_1.jpg**)  
on a déjà parlé de la contrainte sur w plus tôt au point (2)  
dans cette photo il n'y a rien a reprendre pour le manuscrit mais on a clarifié les deux manières de voir apparaître une norme 1 dans la formalisation (en bas à droite de la photo).  
  
  
4e photo (**2016-01-11 11.31.08\_L1\_L2\_2.jpg**)  
  
7- (haut droite photo) problem interpretation : maintenant que notre problème est formalisé, on l'interprète. Pour cela on fait apparaître deux termes :   
\* un terme de régularisation R(D, X) = .... = (N.k) w'(x\_ij)barre    (pour le développement, cf article springer, page 8). C'est donc à une constante (Nk) près, une norme 1 du vecteur w, pondérée par le vecteur 'pullé' moyen  ((x\_ij)barre)  
\* un terme de loss L(Xsi) = la norme 1 du vecteur des Xsi\_ijl (triplets 'Push/Pull')  
\* la marge min(||x\_il-x\_ij||\_2) = 1/||w||\_2    Note: on n'en a pas parlé ce matin, donc à rediscuter la prochaine fois si pas clair  
  
8- On peut donc choisir un autre régulariseur ou une autre loss pour résoudre ce problème. Par la suite nous considérerons seulement un autre régulariseur (L2), afin de faire apparaître un problème de minimisation quadratique ne faisant appel qu'à des produits scalaires, pour pouvoir appliquer le kernel trick. C'est notre problème 2.  
  
9- intégralité du développement QP ici : écriture dual, kernelisation  
  
10- approximation SVM. Une fois que l'on a montré que le SVM est une approximation sur-contrainte du problème QP, expliquer qu'il peut cependant y avoir un intérêt pratique (réutiliser des librairies existantes).   
  
11- (à discuter et développer si nécéssaire). La preuve d'approximation SVM ne vaut-t'elle que pour la norme 2 ou pour les deux normes ? bien expliquer, je ne me souviens plus trop. Dans ce cas faut-t'il donner un numéro à ces problèmes approximés ? (problèmes 3 et 4 ?)  
  
12- (photo à droite) résolutions  
\* problème 1, on rappelle que l'on peut mettre ou pas la contrainte, et quel est l'intérêt. Puis présenter les solveurs à utiliser pour résoudre  
\* problème 2, idem discuter contraintes  
\* problèmes approximés : expliquer que l'on ne peut pas imposer la contrainte si on utilise des solveurs SVM externes.  
  
  
Je vous laisse tenir au courant Michèle ? merci d'avance  
  
Bon courage à tous les deux   
Prochains points : Mercredi 20 12h30-14h30 puis Vendredi 22 9h-13h