Optimization for well-behaved problems

For statistical learning problems, "well-behaved" means:

- signal to noise ratio is decently high
- correlations between predictor variables are under control
- number of predictors p can be larger than number of observations n, but not absurdly so

For well-behaved learning problems, people have observed that gradient or generalized gradient descent can converge extremely quickly (much more so than predicted by $O(1/k)$ rate)

Largely unexplained by theory, topic of current research. E.g., very recent work[4] shows that for some well-behaved problems, w.h.p.:

$$\|x^{(k)} - x^\star\|^2 \leq c^k \|x^{(0)} - x^\star\|^2 + o(\|x^\star - x^{\mathsf{true}}\|^2)$$

---

[4] Agarwal et al. (2012), *Fast global convergence of gradient methods for high-dimensional statistical recovery*

Administrivia

- HW2 out as of this past Tuesday—due 10/9
- Scribing
  - Scribes 1–6 ready soon; handling errata
  - missing days: 11/6, 12/4, and 12/6
- Projects:
  - you should expect to be contacted by TA mentor in next weeks
  - project milestone: 10/30

Matrix differential calculus

10-725 Optimization

Geoff Gordon
Ryan Tibshirani

Matrix calculus pain

- Take derivatives of fns involving matrices:
  - write as huge multiple summations w/ lots of terms
  - take derivative as usual, introducing more terms
    - case statements for $i = j$ vs. $i \neq j$
  - try to recognize that output is equivalent to some human-readable form
  - hope for no indexing errors…
- Is there a better way?

Differentials

- Assume f sufficiently "nice"
- Taylor: f(y) = f(x) + f'(x) (y–x) + r(y–x)
  - with r(y–x) / |y–x| → 0 as y → x
- Notation:

Definition

- Write
  - dx = y – x
  - df = f(y) – f(x)
- Suppose
  - df =
  - with
  - and
- Then:

Matrix differentials

- For matrix X or matrix-valued function F(X):
  - dX =
  - dF =
  - where
  - and
- Examples:

Working with differentials

- Linearity:
  - d(f(x) + g(x)) =
  - d(k f(x)) =
- Common linear operators:
  - reshape(A, [m n k …])
  - vec(A) = A(:) = reshape(A, [], 1)
  - tr(A) =
  - A^T

Reshape

```
>> reshape(1:24, [2 3 4])
ans(:,:,1) =
     1     3     5
     2     4     6

ans(:,:,2) =
     7     9    11
     8    10    12

ans(:,:,3) =
    13    15    17
    14    16    18

ans(:,:,4) =
    19    21    23
    20    22    24
```

Working with differentials

- Chain rule: L(x) = f(g(x))
  - want:
  - have:

Working with differentials

- Product rule: L(x) = c(f(x), g(x))
  - where c is ***bilinear*** = linear in each argument (with other argument fixed)
  - e.g., L(x) = f(x)g(x): f, g scalars, vectors, or matrices

Lots of products

- Cross product: d(a × b) =
- Hadamard product A ○ B = A .* B
  - $(A \circ B)_{ij}$ =
  - d(A ○ B) =
- Kronecker product d(A ⊗ B) =
- Frobenius product A:B =
- Khatri-Rao product: d(A*B) =

Kronecker product

```
>> A = reshape(1:6, 2, 3)

A =

     1     3     5
     2     4     6

>> B = 2*ones(2)

B =

     2     2
     2     2
```

```
>> kron(A, B)

ans =

     2     2     6     6    10    10
     2     2     6     6    10    10
     4     4     8     8    12    12
     4     4     8     8    12    12

>> kron(B, A)

ans =

     2     6    10     2     6    10
     4     8    12     4     8    12
     2     6    10     2     6    10
     4     8    12     4     8    12
```

Hadamard product

- a, b vectors
  - a ○ b =
  - diag(a) diag(b) =
  - tr(diag(a) diag(b)) =
  - tr(diag(b)) =

Some examples

- $L = (Y - XW)^T(Y - XW)$: differential wrt W
  - $dL =$

Some examples

- L = dL =
- L =
  - dL =

Trace

- $\mathrm{tr}(A) = \sum A_{ii}$
  - $d\,\mathrm{tr}(f(x)) =$
  - $\mathrm{tr}(x) =$
  - $\mathrm{tr}(X^T) =$
- Frobenius product:
  - $A:B =$

Trace rotation

- tr(AB) =
- tr(ABC) =
  - size(A):
  - size(B):
  - size(C):

More

- Identities: for a matrix X,
  - $d(X^{-1}) =$
  - $d(\det X) =$
  - $d(\ln |\det X|) =$
  - …

Example: linear regression

- Training examples:
- Input feature vectors:
- Target vectors:
- Weight matrix:
- $\min_W L$ =
  - as matrix:

Linear regression

- $L = ||Y - WX||_F^2 =$

- $dL =$

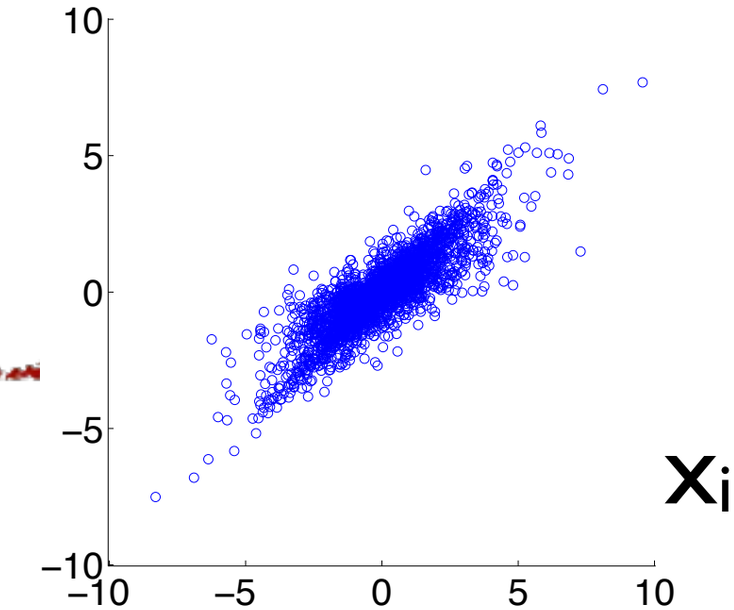
Identification theorems

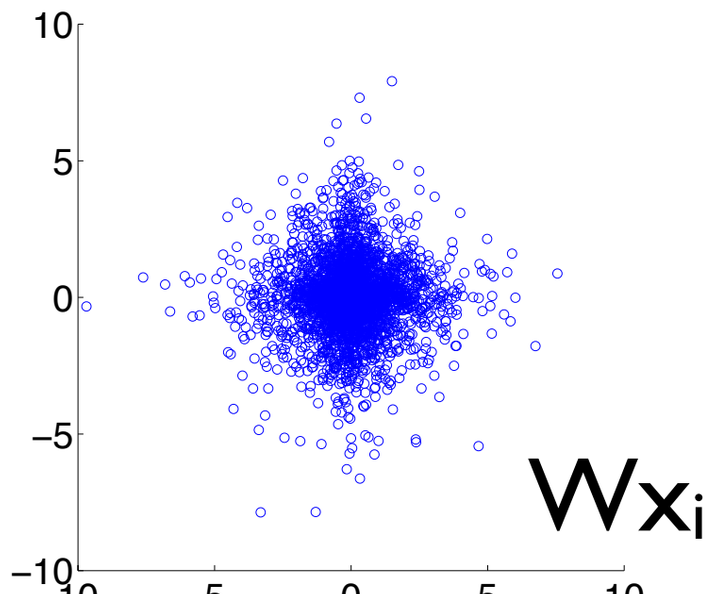
- Sometimes useful to go back and forth between differential & ordinary notations
  - not always possible: e.g., $d(X^TX) =$
- Six common cases (ID thms):

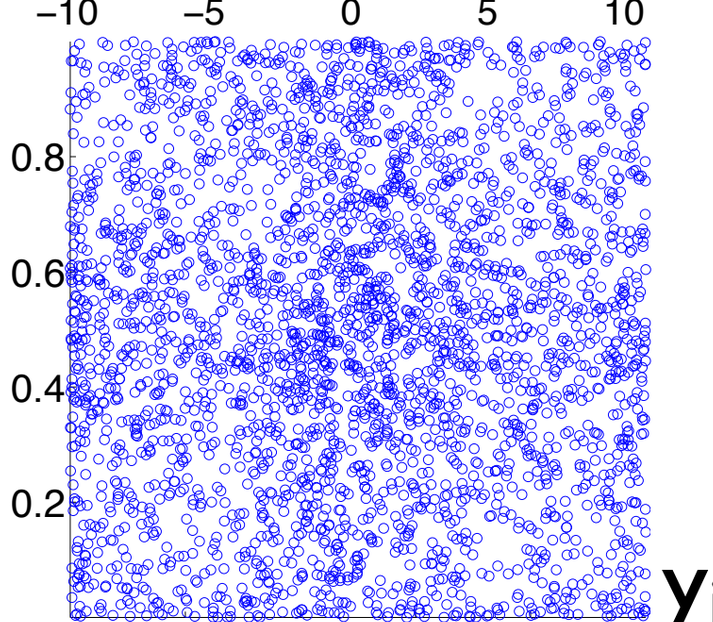
ID for df(x)	scalar x	vector $\boldsymbol{x}$	matrix X
scalar f	$df = a\,dx$	$df = \boldsymbol{a}^T d\boldsymbol{x}$	$df = \mathrm{tr}(A^T dX)$
vector $\boldsymbol{f}$	$d\boldsymbol{f} = \boldsymbol{a}\,dx$	$d\boldsymbol{f} = A\,d\boldsymbol{x}$	
matrix F	$dF = A\,dx$		

Ex: Infomax ICA

- Training examples $x_i \in \mathbb{R}^d$, i = 1:n
- Transformation $y_i = g(Wx_i)$
  - $W \in \mathbb{R}^{d \times d}$
  - g(z) =
- Want:



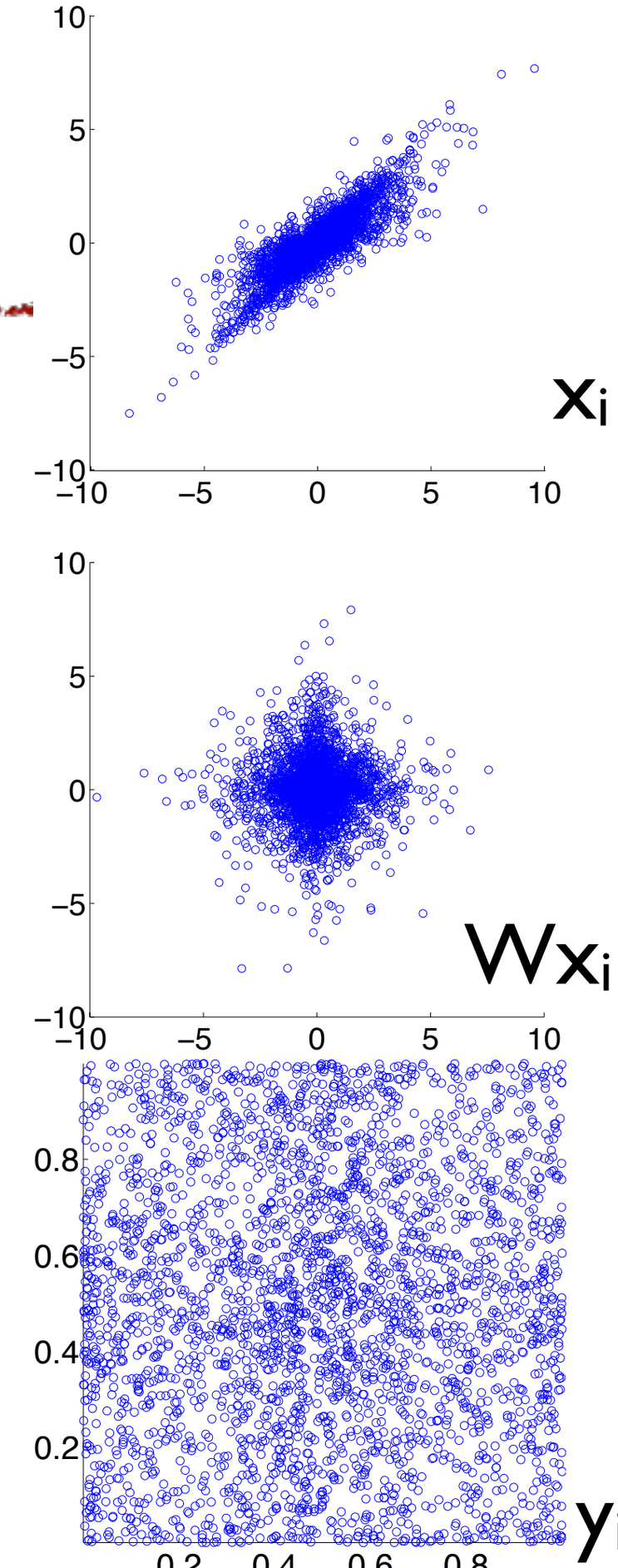




Ex: Infomax ICA

- $y_i = g(Wx_i)$
  - $dy_i =$

- Method: $\max_W \sum_i -\ln(P(y_i))$
  - where $P(y_i) =$



Gradient

- $L = \sum_i \ln |\det J_i| \qquad y_i = g(W x_i) \qquad dy_i = J_i \, dx_i$

Gradient

$J_i = \mathrm{diag}(u_i)\, W$ $\quad dJ_i = \mathrm{diag}(u_i)\, dW + \mathrm{diag}(v_i)\, \mathrm{diag}(dW\, x_i)\, W$

$dL =$

Natural gradient

- Matrix W, gradient dL = G:dW = tr($G^T dW$)
- step S = arg $\max_S$ L(S) = tr($G^T S$) – $||SW^{-1}||_F^2 / 2$
  - scalar case: L = gs – $s^2 / 2w^2$
    - L =
    - dL =

ICA natural gradient

- $[W^{-T} + C] W^T W =$

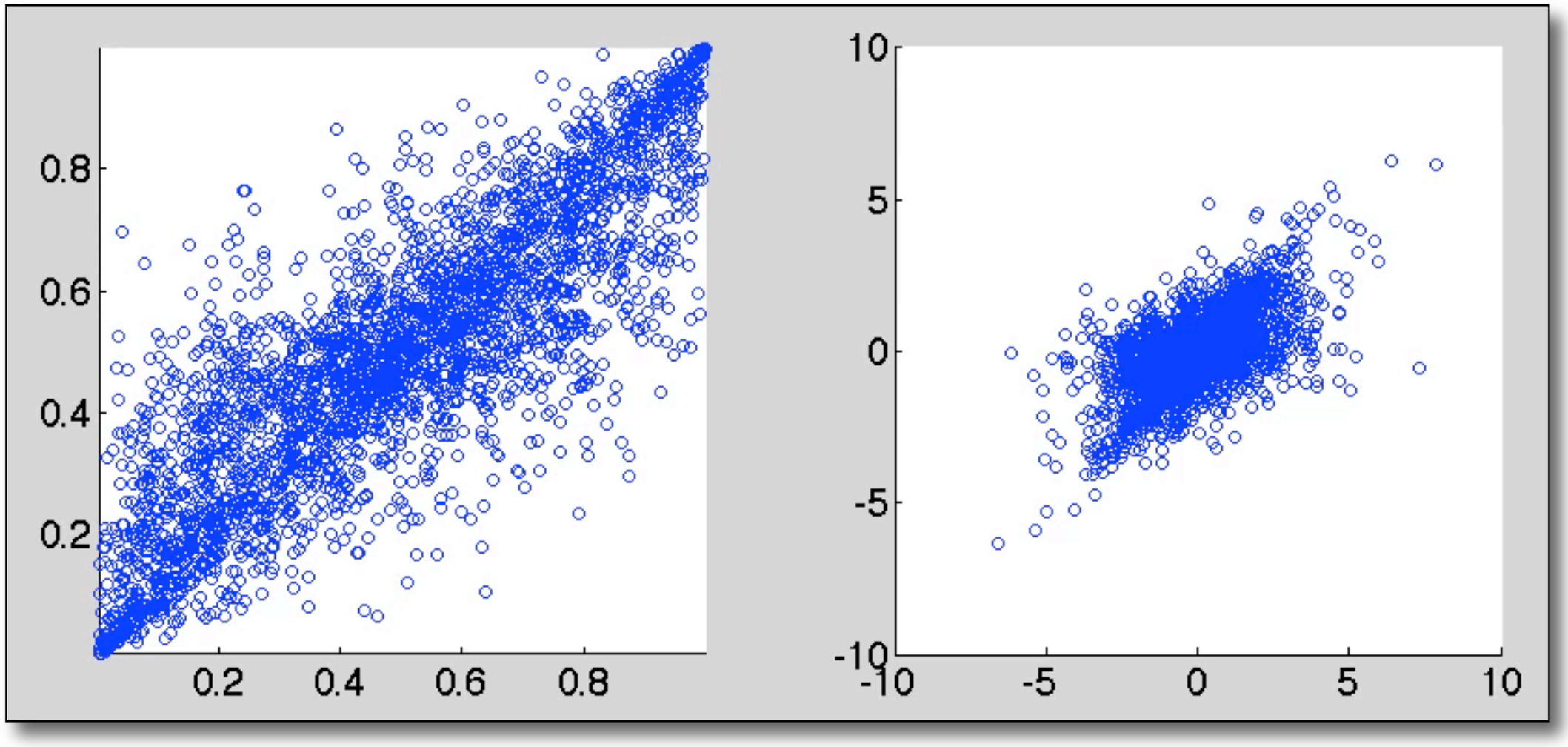
y_i

Wx_i

start with $W_0 = I$

ICA natural gradient

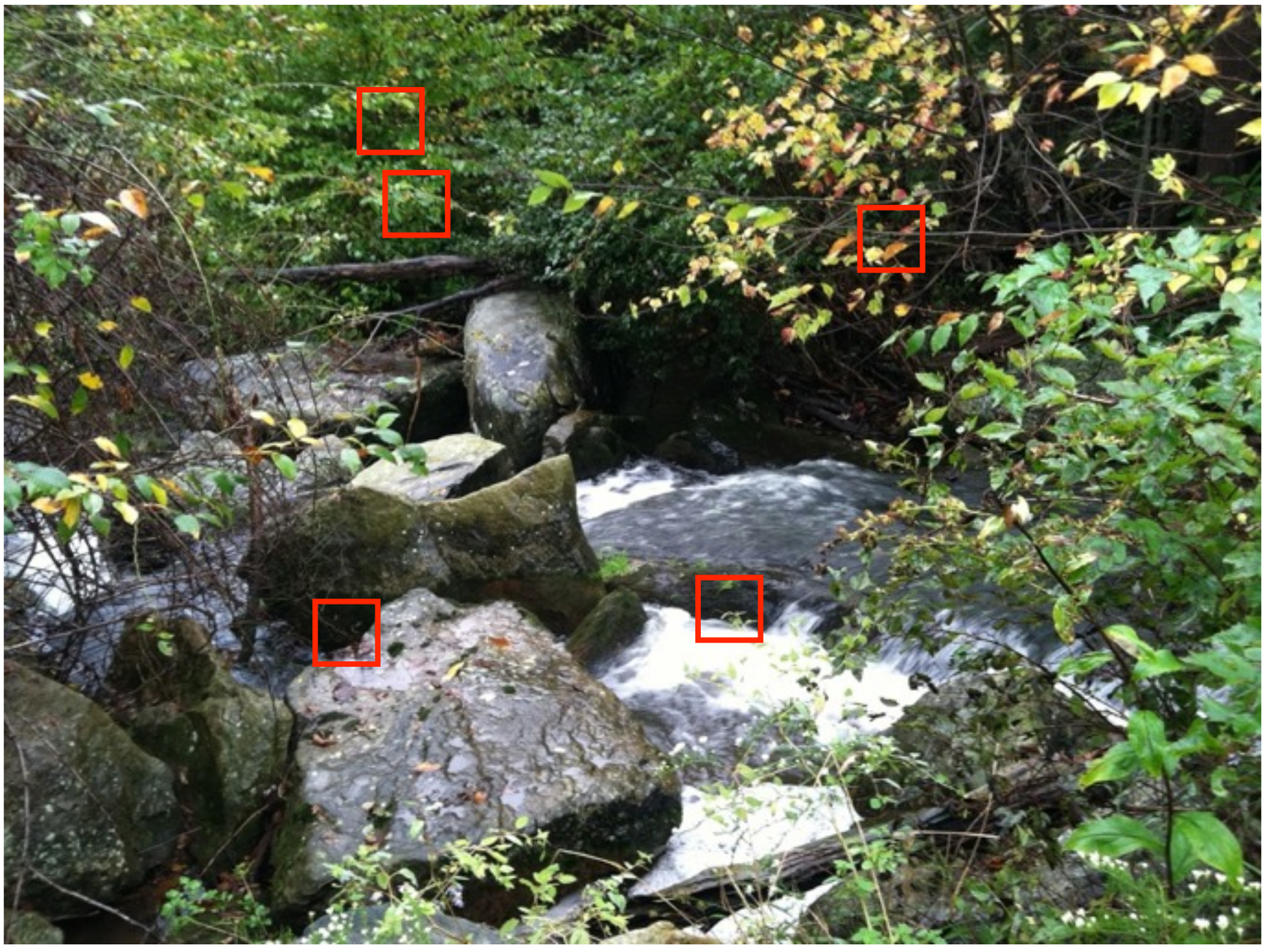
- $[W^{-T} + C] W^T W =$



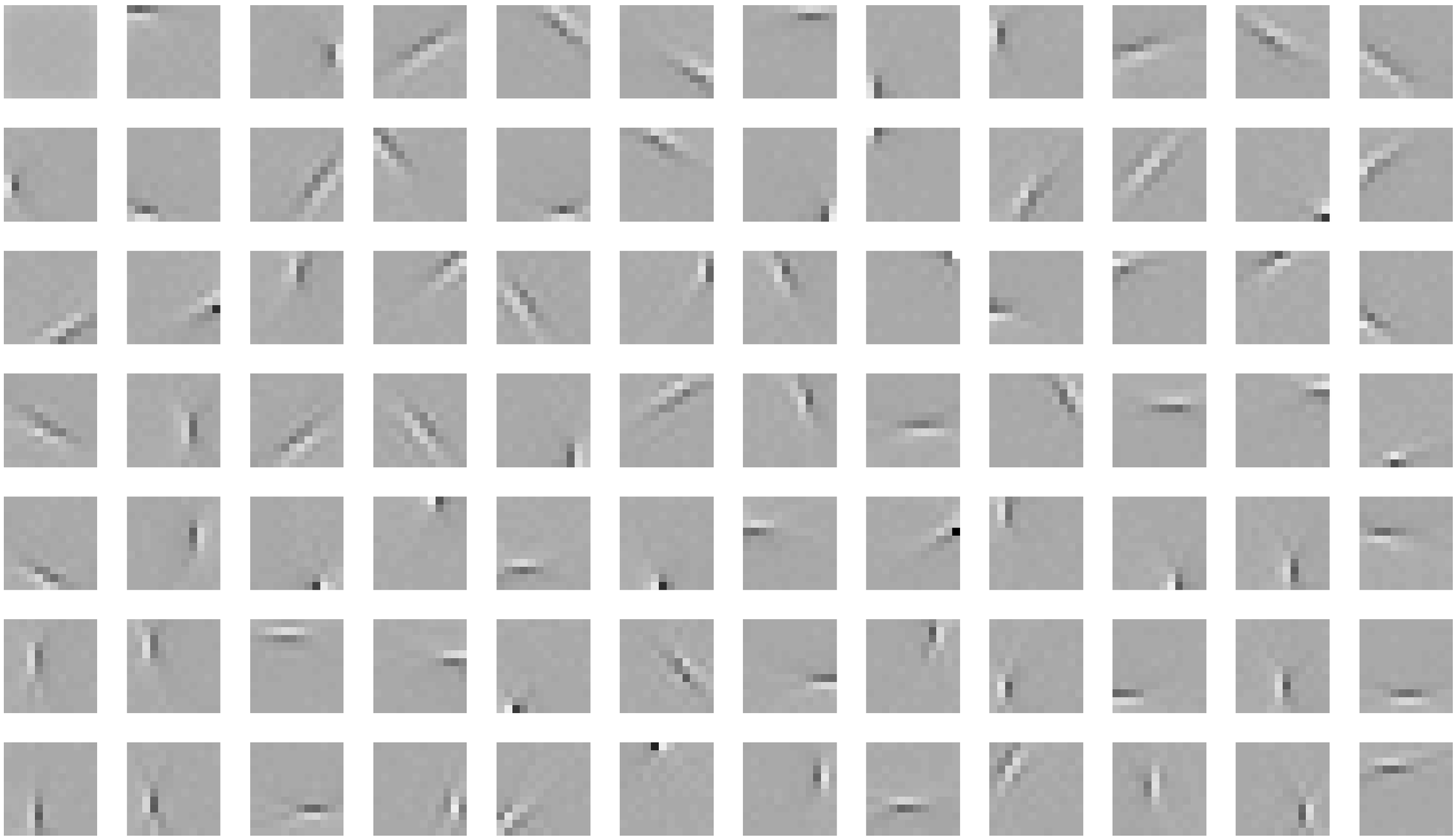
y_i Wx_i

start with $W_0 = I$

ICA on natural image patches



ICA on natural image patches



More info

- Minka's cheat sheet:
  - http://research.microsoft.com/en-us/um/people/minka/papers/matrix/
- Magnus & Neudecker. *Matrix Differential Calculus*. Wiley, 1999. 2nd ed.
  - http://www.amazon.com/Differential-Calculus-Applications-Statistics-Econometrics/dp/047198633X
- Bell & Sejnowski. An information-maximization approach to blind separation and blind deconvolution. Neural Computation, v7, 1995.